aws

Guía del usuario

Amazon CloudWatch

Amazon CloudWatch: Guía del usuario

Table of Contents

¿Qué es Amazon CloudWatch?

Amazon CloudWatch monitorea los recursos y las aplicaciones de Amazon Web Services (AWS) que ejecuta en AWS en tiempo real. Puede utilizar CloudWatch para recopilar y hacer un seguimiento de métricas, que son las variables que puede medir en los recursos y aplicaciones.

La página de inicio de CloudWatch muestra automáticamente las métricas sobre todos los servicios de AWS que utilice. También puede crear adicionalmente paneles personalizados para mostrar métricas sobre sus aplicaciones personalizadas, y mostrar colecciones personalizadas de métricas que elija.

Puede crear alarmas que vigilen métricas y enviar notificaciones o realizar cambios automáticamente en los recursos que está monitoreando cuando se infringe un umbral. Por ejemplo, puede monitorear el uso de la CPU y las lecturas y escrituras de disco de las instancias de Amazon EC2 y, a continuación, utilizar esos datos para determinar si se deben lanzar instancias adicionales para gestionar el aumento de la carga. También puede utilizar estos datos para parar las instancias infrautilizadas a fin de ahorrar dinero.

Con CloudWatch, se obtiene información sobre la utilización de recursos, el rendimiento de las aplicaciones y el estado operativo de todo el sistema.

Acceso a CloudWatch

Puede obtener acceso a CloudWatch con cualquiera de los siguientes métodos:

- Amazon CloudWatch console (Consola de Amazon CloudWatch) – https://console.aws.amazon.com/cloudwatch/
- CLI de AWS: para obtener más información, consulte Configuración inicial con AWS Command Line Interface en la Guía del usuario de AWS Command Line Interface.
- CloudWatch API: Para obtener más información, consulte la Amazon CloudWatch API Reference (Referencia de la API de Amazon CloudWatch).
- AWS SDK: Para obtener más información, consulte Tools for Amazon Web Services (Herramientas para Amazon Web Services)

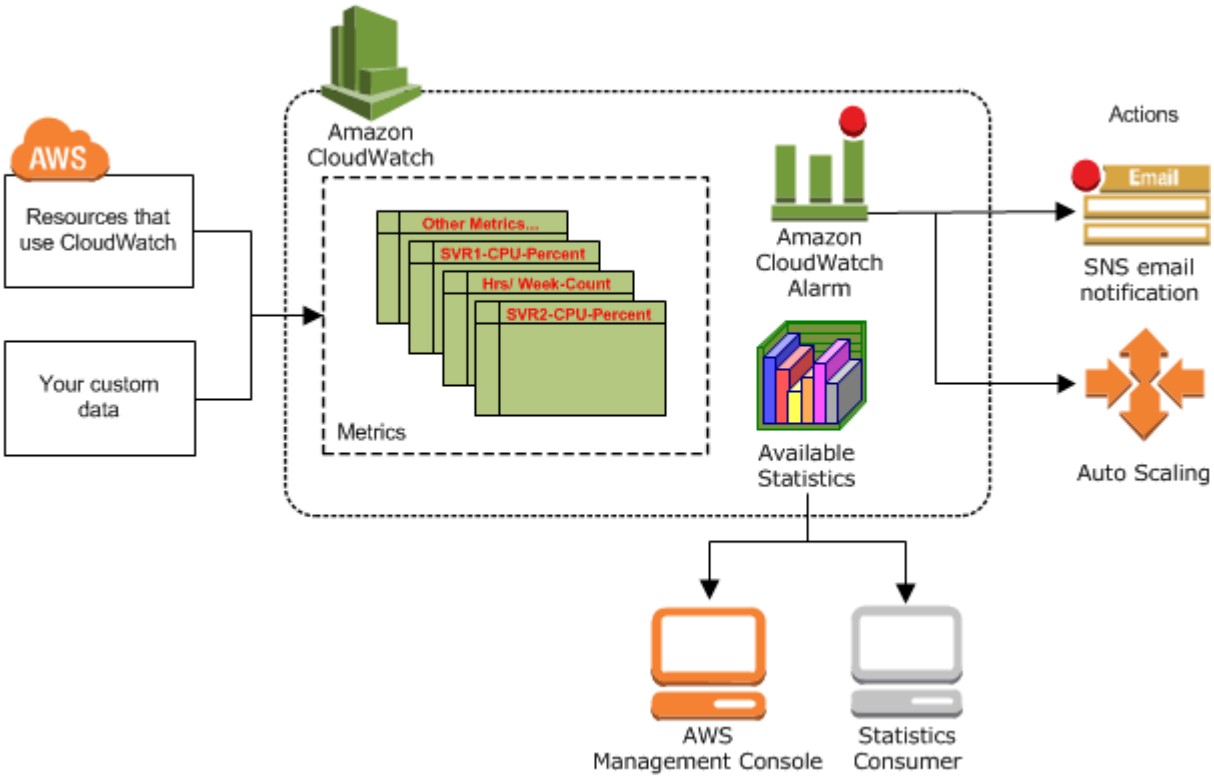
Servicios relacionados de AWS

Los siguientes servicios se utilizan junto con Amazon CloudWatch:

- Amazon Simple Notification Service (Amazon SNS) coordina y administra la entrega o el envío de mensajes a los puntos de enlace o clientes suscritos. Amazon SNS se utiliza con CloudWatch para enviar mensajes cuando se alcanza un umbral de alarma. Para obtener más información, consulte Configuración de notificaciones de Amazon SNS.
- Amazon EC2 Auto Scaling le permite lanzar o terminar instancias de Amazon EC2 automáticamente de acuerdo con las políticas que el usuario define, las verificaciones de estado y las programaciones. Puede utilizar una alarma de CloudWatch con Amazon EC2 Auto Scaling para regular las instancias EC2 en función de la demanda. Para obtener más información, consulte Dynamic Scaling (Escalado dinámico) en la Guía del usuario de Amazon EC2 Auto Scaling.
- AWS CloudTrail le permite monitorear las llamadas a la API de Amazon CloudWatch para su cuenta, incluidas las llamadas que realizan la AWS Management Console, AWS CLI y otros servicios. Cuando el registro de CloudTrail está activado, CloudWatch registra los archivos de registro en el bucket de Amazon S3 que especificó al configurar CloudTrail. Para obtener más información, consulte Registro de llamadas a la API de Amazon CloudWatch con AWS CloudTrail.
- AWS Identity and Access Management (IAM) es un servicio web que ayuda a controlar de forma segura el acceso de los usuarios a los recursos de AWS. Utilice IAM para controlar quién puede usar los recursos de AWS (autenticación), así como cuáles de ellos pueden usar y de qué manera pueden hacerlo (autorización). Para obtener más información, consulte Identity and Access Management para Amazon CloudWatch.

Funcionamiento de Amazon CloudWatch.

Amazon CloudWatch es básicamente un repositorio de métricas. Un servicio de AWS, como Amazon EC2, coloca las métricas en el repositorio, lo que logrará que recupere las estadísticas en función de dichas métricas. Si coloca sus propias métricas personalizadas en el repositorio, puede recuperar estadísticas sobre estas métricas también.



Puede utilizar las métricas para calcular estadísticas y, a continuación, presentar los datos gráficamente en la consola de CloudWatch. Para obtener más información sobre los demás recursos de AWS que generan y envían métricas a CloudWatch, consulte Servicios de AWS que publican métricas de CloudWatch.

Puede configurar acciones de alarma para detener, comenzar o terminar una instancia de Amazon EC2 cuando se cumplen determinados criterios. Por ejemplo, puede crear alarmas que inicien acciones de Amazon EC2 Auto Scaling y Amazon Simple Notification Service (Amazon SNS) en su nombre. Para obtener más información sobre cómo crear alarmas de CloudWatch, consulte Alarmas.

AWSLos recursos de informática en la nube de se alojan en centros de datos de alta disponibilidad. Para proporcionar más escalabilidad y fiabilidad, cada instalación de centro de datos se encuentra en una zona geográfica específica, conocida como región. Cada región está diseñada para estar totalmente aislada de las demás regiones, para lograr la máxima estabilidad y aislamiento en caso de error. Las métricas se almacenan por separado en las Regiones, pero puede utilizar la funcionalidad para diversas regiones de CloudWatch para agregar estadísticas de diferentes Regiones. Para obtener más información, consulte Consola de CloudWatch para cuentas y Regiones cruzadas y Regiones y puntos de enlace en la Referencia general de Amazon Web Services.

Conceptos de Amazon CloudWatch

Los siguientes conceptos y terminología son fundamentales para entender y utilizar Amazon CloudWatch:

- Espacios de nombres
- Métricas
- Dimensiones
- Resolución
- Statistics
- Percentiles
- Alarmas

Para obtener más información acerca de las cuotas de servicio para las métricas de CloudWatch, alarmas, solicitudes de la API y las notificaciones de alarma por email, consulte CloudWatch Service Quotas.

Espacios de nombres

Un espacio de nombres es un contenedor para métricas de CloudWatch. Las métricas en distintos espacios de nombres están aisladas entre sí, de forma que las métricas de distintas aplicaciones no estén acumuladas por error en las mismas estadísticas.

No hay ningún espacio de nombres predeterminado. Especifique un espacio de nombres para cada punto de datos que publique en CloudWatch. Puede especificar un nombre de espacio de nombres al crear una métrica. Estos nombres deben contener caracteres ASCII válidos y tener 255 caracteres o menos. Los caracteres posibles son: caracteres alfanuméricos (0-9A-Za-z), punto (.), guion (-), guion bajo (_), barra inclinada (/), almohadilla (#), dos puntos (:) y el espacio. Un espacio de nombres debe contener al menos un carácter que no sea un espacio en blanco.

Los espacios de nombres de AWS utilizan normalmente la siguiente convención de nomenclatura: `AWS/service`. Por ejemplo, Amazon EC2 utiliza el espacio de nombres de Amazon EC2. `AWS/EC2` Para obtener la lista de espacios de nombres de AWS, consulte Servicios de AWS que publican métricas de CloudWatch.

Métricas

Las Métricas son el concepto fundamental en CloudWatch. Un métrica representa una serie de puntos de datos ordenados por tiempo que se publican en CloudWatch. Una métrica es una variable que hay que monitorizar y los puntos de datos son los valores de esa variable a lo largo del tiempo. Por ejemplo, el uso de la CPU de una determinada instancia EC2 es una métrica que Amazon EC2 proporciona. Los propios puntos de datos pueden proceder de cualquier aplicación o actividad empresarial desde la que recopile datos.

De forma predeterminada, diversos servicios de AWS ofrecen métricas gratuitas para recursos (tales como instancias de Amazon EC2, volúmenes de Amazon EBS e instancias de base de datos de Amazon RDS). También puede habilitar el monitoreo detallado para algunos recursos, como las instancias Amazon EC2 o publicar sus propias métricas de aplicación. Para métricas personalizadas, añada los puntos de datos en cualquier orden y a la velocidad que elija. Puede recuperar estadísticas sobre dichos puntos de datos como un conjunto ordenado de datos de serie temporal.

Las métricas existen solo en la región en que se han creado. Las métricas no se pueden eliminar, pero vencen automáticamente a los 15 meses si no se publican datos nuevos. Los puntos de datos con más de 15 meses caducarán sucesivamente; a medida que se introducen nuevos puntos de datos, los datos con más de quince meses se eliminan.

Las métricas se definen de forma exclusiva mediante un nombre, un espacio de nombres y cero o varias dimensiones. Cada punto de datos de una métrica tiene una marca temporal y, opcionalmente, una unidad de medida. Recuperar estadísticas de CloudWatch para cualquier métrica.

Para obtener más información, consulte Ver métricas disponibles y Publicar métricas personalizadas de .

Marcas temporales

Cada punto de datos de métrica debe asociarse a una marca temporal. La marca temporal puede ser de hasta dos semanas en el pasado y de hasta dos horas en el futuro. Si no proporciona una marca temporal, CloudWatch crea una en función de la hora a la que se recibió el punto de datos.

Las marcas temporales son objetos `dateTime`, con la fecha completa más horas, minutos y segundos (por ejemplo, 2016-10-31T23:59:59 Z). Para obtener más información, consulte dateTime. Aunque no es necesario, le recomendamos que utilice la hora universal coordinada (UTC, por sus siglas en inglés). Al recuperar las estadísticas de CloudWatch, todas las horas se indican en UTC.

Las alarmas de CloudWatch verifican las métricas en función de la hora actual en UTC. Las métricas personalizadas enviadas a CloudWatch con marcas temporales que no sean la hora UTC actual pueden hacer que las alarmas muestren el estado Insufficient Data (Datos insuficientes) o dar lugar a retrasos en las alarmas.

Retención de métricas

CloudWatch retiene los datos de las métricas como se indica a continuación:

- Los puntos de datos con un período de menos de 60 segundos están disponibles durante 3 horas. Estos puntos de datos son métricas personalizadas de alta resolución.
- Los puntos de datos con un período de 60 segundos (1 minuto) están disponibles durante 15 días
- Los puntos de datos con un período de 300 segundos (5 minutos) están disponibles durante 63 días
- Los puntos de datos con un período de 3 600 segundos (1 hora) están disponibles para 455 días (15 meses)

Los puntos de datos que se publican inicialmente con un periodo más corto se acumulan para almacenarlos a largo plazo. Por ejemplo, si recopila datos con un periodo de 1 minuto, los datos están disponibles durante 15 días con una resolución de 1 minuto. Después de 15 días estos datos siguen estando disponibles, pero se acumulan y solo se pueden recuperar con una resolución de 5 minutos. Después de 63 días, los datos siguen acumulándose y están disponibles con una resolución de 1 hora.

> **Note**
>
> Las métricas que no han tenido nuevos puntos de datos en las últimas dos semanas no aparecen en la consola. Tampoco aparecen al escribir su nombre de métrica o nombres de dimensión en el cuadro de búsqueda de la pestaña Todas las métricas de la consola y no se devuelven en los resultados de un comando list-metrics. La mejor manera de recuperar estas métricas es con los comandos get-metric-data o get-metric-statistics de la AWS CLI.

Dimensiones

Una dimensión es un par de nombre-valor que forma parte de la identidad de una métrica. Puede asignar hasta 30 dimensiones a una métrica.

Cada métrica tiene características específicas que la describen y puede considerar las dimensiones como categorías para las características. Las dimensiones le ayudan a diseñar una estructura para su plan de estadísticas. Dado que las dimensiones forman parte del identificador único de una métrica, si añade un par único nombre/valor a una de las métricas, está creando una nueva variación de esa métrica.

Los servicios de AWS que envían datos a CloudWatch adjuntan dimensiones a cada métrica. Puede utilizar dimensiones para filtrar los resultados que muestra CloudWatch. Por ejemplo, puede obtener estadísticas para una instancia EC2 concreta especificando la dimensión `InstanceId` al buscar métricas.

Para las métricas que determinados servicios de AWS producen, como Amazon EC2, CloudWatch puede acumular datos a través de las dimensiones. Por ejemplo, si busca métricas en el espacio de nombres de Amazon EC2 pero sin especificar ninguna dimensión, CloudWatch acumula todos los datos de la métrica especificada para crear la estadística que ha solicitado. `AWS/EC2` CloudWatch no acumula entre dimensiones para las métricas personalizadas.

Combinación de dimensiones

CloudWatch trata cada combinación exclusiva de dimensiones como una métrica independiente, incluso si las métricas tienen el mismo nombre de métrica. Solo puede recuperar estadísticas utilizando combinaciones de dimensiones que haya publicado específicamente. Al recuperar estadísticas, especifique los mismos valores para el espacio de nombres, el nombre de la métrica y los parámetros de dimensión que se utilizaron cuando se crearon las métricas. También se pueden especificar las horas de inicio y finalización que utiliza CloudWatch para la agregación.

Por ejemplo, suponga que publica cuatro métricas distintas denominadas ServerStats en el espacio de nombres DataCenterMetric con las siguientes propiedades:

```
Dimensions: Server=Prod, Domain=Frankfurt, Unit: Count, Timestamp:
 2016-10-31T12:30:00Z, Value: 105
Dimensions: Server=Beta, Domain=Frankfurt, Unit: Count, Timestamp:
 2016-10-31T12:31:00Z, Value: 115
Dimensions: Server=Prod, Domain=Rio,       Unit: Count, Timestamp:
 2016-10-31T12:32:00Z, Value: 95
Dimensions: Server=Beta, Domain=Rio,       Unit: Count, Timestamp:
 2016-10-31T12:33:00Z, Value: 97
```

Si publica solo estas cuatro métricas, puede recuperar estadísticas para estas combinaciones de dimensiones:

- `Server=Prod,Domain=Frankfurt`
- `Server=Prod,Domain=Rio`
- `Server=Beta,Domain=Frankfurt`
- `Server=Beta,Domain=Rio`

No puede recuperar estadísticas para las siguientes dimensiones o si no especifica ninguna dimensión. (La excepción es el uso de la función SEARCH (BUSCAR) de los cálculos de las métricas, que puede recuperar estadísticas para varias métricas. Para obtener más información, consulte Usar expresiones de búsqueda en gráficos.)

- `Server=Prod`
- `Server=Beta`
- `Domain=Frankfurt`
- `Domain=Rio`

Resolución

Cada métrica es una de las siguientes:

- Resolución estándar, con datos cuya granularidad es de un minuto
- Alta resolución, con datos cuya granularidad es de un segundo

De forma predeterminada, las métricas producidas por los servicios de AWS son de resolución estándar. Al publicar una métrica personalizada, puede definirla como de resolución estándar o de alta resolución. Cuando publica una métrica de alta resolución, CloudWatch la almacena con una resolución de 1 segundo, y puede leerla y recuperarla con un periodo de 1 segundo, 5 segundos, 10 segundos, 30 segundos o cualquier múltiplo de 60 segundos.

Las métricas de alta resolución pueden ofrecerle más información inmediata acerca de las actividades de su aplicación, cuya duración sea inferior a un minuto. Tenga en cuenta que cada llamada a `PutMetricData` para una métrica personalizada se cobra; por tanto, realizar llamadas a `PutMetricData` con más frecuencia en una métrica de alta resolución podría derivar en cargos más elevados. Para obtener más información sobre los precios de CloudWatch, consulte Precios de Amazon CloudWatch.

Si configura una alarma en una métrica de alta resolución, puede especificar una alarma de alta resolución con un periodo de 10 o 30 segundos, o puede definir una alarma normal con un periodo de cualquier múltiplo de 60 segundos. El cargo es mayor para las alarmas de alta resolución con un periodo de 10 o 30 segundos.

Statistics

Las estadísticas son agregaciones de datos de métricas correspondientes a periodos especificados. CloudWatch proporciona estadísticas en función de los puntos de datos de las métricas que proporcionan los datos personalizados u otros servicios de AWS para CloudWatch. Las acumulaciones se realizan utilizando el espacio de nombres, el nombre de métrica, las dimensiones y la unidad de medida de punto de datos, dentro del período de tiempo que especifique.

Para obtener definiciones detalladas de las estadísticas que CloudWatch admite, consulte Definiciones de estadísticas de CloudWatch.

Unidades

Cada estadística tiene una unidad de medida. Entre las unidades de ejemplo se incluyen `Bytes`, `Seconds`, `Count` y `Percent`. Para ver la lista completa de las unidades que CloudWatch admite, consulte el tipo de datos MetricDatum en la Referencia de la API de Amazon CloudWatch.

Puede especificar una unidad al crear una métrica personalizada. Si no especifica una unidad, CloudWatch utiliza `None` como unidad. Las unidades ayudan a aportar significado conceptual a los datos. Aunque CloudWatch no concede ninguna importancia a una unidad internamente, otras aplicaciones obtienen información semántica en función de la unidad.

Los puntos de datos de métricas que especifican una unidad de medida se suman por separado. Cuando se obtienen estadísticas sin especificar una unidad, CloudWatch acumula todos los puntos de datos de la misma unidad en forma conjunta. Si tiene dos métricas idénticas con distintas unidades, se devuelven dos flujos de datos independientes, uno por cada unidad.

Periodos

Un período es el tiempo asociado a una estadística de Amazon CloudWatch específica. Cada estadística representa una suma de los datos de métricas recopilados durante un periodo de tiempo especificado. Los periodos se definen en números de segundos, y los valores válidos para el periodo son 1, 5, 10, 30 o cualquier múltiplo de 60. Por ejemplo, para especificar un periodo de seis minutos, utilice el valor de periodo 360. Puede ajustar la forma en que los datos se acumulan

variando la duración del periodo. El valor de predeterminado de un periodo es de 60 segundos. Un periodo puede ser tan breve como un segundo y debe ser un múltiplo de 60 si es mayor que el valor predeterminado de 60 segundos.

Solo las métricas personalizadas que defina con una resolución de almacenamiento de 1 segundo admiten periodos inferiores a un minuto. Aunque la opción de establecer un periodo inferior a 60 siempre está disponible en la consola, debe seleccionar un periodo acorde con el almacenamiento de la métrica. Para obtener más información sobre las métricas que admiten periodos inferiores a un minuto, consulte Métricas de alta resolución.

Al recuperar las estadísticas, puede especificar un periodo, hora de inicio y hora de finalización. Estos parámetros determinan la duración de tiempo asociada a las estadísticas. Los valores predeterminados de la hora de inicio y de finalización le proporcionan las estadísticas de la última hora. Los valores que especifique para la hora de inicio y la hora de finalización determinan cuántos periodos muestra CloudWatch. Por ejemplo, la recuperación de estadísticas utilizando los valores predeterminados para el período, hora de inicio y hora de finalización devuelven un conjunto acumulado de estadísticas para cada minuto de la hora anterior. Si prefiere estadísticas acumuladas en bloques de diez minutos, especifique un periodo de 600. Para estadísticas acumuladas en toda la hora, especifique un periodo de 3 600.

Cuando se acumulan estadísticas a lo largo de un periodo de tiempo, se marcan con la hora correspondiente al principio del periodo. Por ejemplo, los datos acumulados desde las 19:00 hasta las 20:00 horas se marcan como 19:00 h. Además, datos acumulados entre las 19:00 y las 20:00 h empiezan a ser visibles a las 19:00 horas y, a continuación, los valores de esos datos acumulados pueden cambiar a medida que CloudWatch recopila más muestras durante el periodo.

Los períodos también son importantes para las alarmas de CloudWatch. Cuando se crea una alarma para monitorear una métrica específica, le solicita a CloudWatch que compare dicha métrica con el valor del umbral que especificó. Tiene amplio control sobre la manera en que CloudWatch lleva a cabo la comparación. No solo puede especificar el periodo durante el que se realiza la comparación, sino que además puede especificar cuántos periodos de evaluación se utilizan para llegar a una conclusión. Por ejemplo, si especifica tres periodos de evaluación, CloudWatch compara una ventana de tres puntos de datos. CloudWatch solo le notifica si el punto de datos más antiguo falla y si los demás fallan o faltan.

Agregación

Amazon CloudWatch acumula estadísticas de acuerdo con la duración del periodo que especifique al recuperar las estadísticas. Publique tantos puntos de datos como desee con las mismas marcas

temporales o similares. CloudWatch las acumula de acuerdo con la longitud de período que se especifique. CloudWatch no acumula datos automáticamente entre Regiones, pero utilice matemáticas de métricas para agregar métricas de distintas Regiones.

Publique puntos de datos para una métrica que comparte no solo la misma marca temporal, sino también el mismo espacio de nombres y dimensiones. CloudWatch muestra las estadísticas acumuladas para dichos puntos de datos. También puede publicar varios puntos de datos para la misma métrica o distintas, con cualquier marca temporal.

Para conjuntos de datos de gran tamaño, puede insertar un conjunto de datos acumulados previamente denominado conjunto estadístico. Con los conjuntos estadísticos, usted le proporciona a CloudWatch los valores Min, Max, Sum y SampleCount para una serie de puntos de datos. Esto se utiliza generalmente cuando hay que recopilar datos muchas veces en un minuto. Por ejemplo, suponga que tiene una métrica para la latencia de solicitudes de una página web. No tiene sentido publicar datos con cada visita a la página web. Se recomienda que recopile la latencia de todas las visitas a dicha página web, las acumule una vez por minuto y que envíe dicho conjunto estadístico a CloudWatch.

Amazon CloudWatch no diferencia la fuente de una métrica. Si publica una métrica con el mismo espacio de nombres y dimensiones de distintas fuentes, CloudWatch las trata como una única métrica. Esto puede resultar útil para métricas de servicio en un sistema de escala distribuido. Por ejemplo, todos los anfitriones de una aplicación del servidor web podrían publicar métricas idénticas que representan la latencia de las solicitudes que están procesando. CloudWatch las trata como una única métrica, lo que le permite obtener estadísticas de mínimo, máximo, promedio y suma de todas las solicitudes en la aplicación.

Percentiles

Un percentil indica el peso relativo de un valor en un conjunto de datos. Por ejemplo, el percentil 95 significa que el 95 por ciento de los datos está por debajo de este valor y el 5 por ciento de los datos está por encima del mismo. Los percentiles le ayudan a entender mejor la distribución de los datos de métricas.

Los percentiles se suelen utilizar para aislar anomalías. En una distribución normal, el 95 % de los datos está dentro de dos desvíos estándar de la media y el 99,7 % de los datos está dentro de tres desvíos estándar de la media. Cualquier dato que quede fuera de las tres desvíos estándar se suele considerar una anomalía ya que se aleja mucho del valor medio. Por ejemplo, suponga que está monitorizando la utilización de la CPU de las instancias EC2 para asegurarse de que sus clientes disfruten de una buena experiencia. Si monitoriza la media, esto puede ocultar anomalías.

Si monitoriza el máximo, una única anomalía puede sesgar los resultados. Mediante los percentiles, puede monitorizar el percentil 95.º de la utilización de la CPU para comprobar si hay instancias con una carga excepcionalmente alta.

Algunas métricas de CloudWatch admiten percentiles como una estadística. Para estas métricas, puede monitorear el sistema y las aplicaciones con percentiles como haría al usar el resto de las estadísticas de CloudWatch (Promedio, Mínimo, Máximo y Suma). Por ejemplo, al crear una alarma, puede utilizar los percentiles como función estadística. Puede especificar el percentil con hasta dos decimales (por ejemplo, p 95,0123456789).

Las estadísticas de percentiles están disponibles para las métricas personalizadas, siempre y cuando publique puntos de datos sin resumir y sin formato para la métrica personalizada. Las estadísticas de percentiles no están disponibles para las métricas cuando alguno de los valores de métricas es un número negativo.

CloudWatch necesita puntos de datos sin procesar para calcular los percentiles. Si en cambio publica datos a través de un conjunto estadístico, solo puede recuperar estadísticas de percentiles para estos datos si es cierta una de las siguientes condiciones:

- El valor de SampleCount del conjunto estadístico es 1 y el mínimo, el máximo y la suma son todos iguales.
- El mínimo y el máximo son iguales y la suma es igual al mínimo multiplicado por SampleCount.

Los siguientes ejemplos de servicios de AWS incluyen métricas que admiten estadísticas de percentiles.

- API Gateway
- Application Load Balancer
- Amazon EC2
- Elastic Load Balancing
- Kinesis
- Amazon RDS

CloudWatch también admite las medias recortadas y otras estadísticas de rendimiento que pueden tener un uso similar como percentiles. Para obtener más información, consulte Definiciones de estadísticas de CloudWatch.

Alarmas

Puede utilizar una alarma para iniciar automáticamente acciones en su nombre. Una alarma vigila una única métrica durante el período especificado y realiza una o varias acciones especificadas según el valor de la métrica relativo a un determinado umbral durante un período de tiempo. La acción es una notificación que se envía a un tema de Amazon SNS o a una política de Auto Scaling. También puede agregar alarmas a paneles.

Las alarmas invocan acciones únicamente para los cambios de estado prolongados. Las alarmas de CloudWatch no invocan acciones simplemente porque se encuentren en un estado determinado. El estado debe haber cambiado y debe mantenerse durante el número de periodos especificado.

Al crear una alarma, seleccione un período de monitoreo de alarma mayor o igual que la resolución de la métrica. Por ejemplo, el monitoreo básico de Amazon EC2 proporciona métricas para las instancias cada 5 minutos. Al configurar una alarma en una métrica de monitorización básica, seleccione un periodo de al menos 300 segundos (5 minutos). El monitoreo detallado para Amazon EC2 proporciona métricas para las instancias con una resolución de 1 minuto. Al configurar una alarma en una métrica de monitorización detallada, seleccione un periodo de al menos 60 segundos (1 minuto).

Si configura una alarma en una métrica de alta resolución, puede especificar una alarma de alta resolución con un periodo de 10 o 30 segundos, o puede definir una alarma normal con un periodo de cualquier múltiplo de 60 segundos. El cargo es mayor en el caso de las alarmas de alta resolución. Para obtener más información acerca de las métricas de alta resolución, consulte Publicar métricas personalizadas de .

Para obtener más información, consulte Uso de las alarmas de Amazon CloudWatch y Crear una alarma desde una métrica en un gráfico.

Facturación y costos

Para obtener más información sobre los precios de CloudWatch, consulte Precios de Amazon CloudWatch.

Para obtener información que pueda ayudarle a analizar su factura y, posiblemente, a optimizar y reducir los costes, consulte. Facturación y costo de CloudWatch

Recursos de Amazon CloudWatch

Los recursos relacionados siguientes pueden serle de ayuda cuando trabaje con este servicio.

Recurso	Descripción
Preguntas frecuentes sobre Amazon CloudWatch	Las preguntas frecuentes tratan las principales preguntas planteadas por los desarrolladores sobre este producto.
Centro de desarrolladores de AWS	Punto de comienzo central para buscar documentación, ejemplos de código, notas de la versión y otra informaci ón que le ayudará a crear aplicaciones innovadoras con AWS.
AWS Management Console	La consola le permite llevar a cabo la mayoría de las funciones de Amazon CloudWatch y otras ofertas de AWS sin necesidad de realizar una programación.
Foros de discusión de Amazon CloudWatch	Foro de la comunidad para desarrolladores donde se tratan aspectos técnicos relacionados con Amazon CloudWatch.
AWS Support	El centro para crear y administrar los casos de AWS Support. También incluye enlaces a otros recursos útiles como foros, preguntas técnicas frecuentes, estado de los servicios y AWS Trusted Advisor.
Información del producto de Amazon CloudWatch	Página web principal con información acerca de Amazon CloudWatch.
Contacto	Un punto de contacto centralizado para las consultas relacionadas con la facturación, cuentas, eventos, abuso, etc. de AWS.

Configuración inicial

Para utilizar Amazon CloudWatch, se necesita una cuenta de AWS. Su cuenta de AWS le permite utilizar servicios (por ejemplo, Amazon EC2) para generar métricas que se pueden visualizar en la consola de CloudWatch, una interfaz de selección y activación basada en la web. Además, puede instalar y configurar la interfaz de línea de comandos (CLI) de AWS.

Registro en una Cuenta de AWS

Si no dispone de una Cuenta de AWS, siga estos pasos para crear una.

Procedimiento para registrarse en Cuenta de AWS

1. Abra https://portal.aws.amazon.com/billing/signup.
2. Siga las instrucciones que se le indiquen.

   Parte del procedimiento de registro consiste en recibir una llamada telefónica e indicar un código de verificación en el teclado del teléfono.

   Al registrarse en una Cuenta de AWS, se crea un Usuario raíz de la cuenta de AWS. El usuario raíz tendrá acceso a todos los Servicios de AWS y recursos de esa cuenta. Como práctica recomendada de seguridad, asigne acceso administrativo a un usuario y utilice únicamente el usuario raíz para realizar tareas que requieren acceso de usuario raíz.

AWS le enviará un email de confirmación cuando complete el proceso de registro. Puede ver la actividad de la cuenta y administrar la cuenta en cualquier momento entrando en https://aws.amazon.com/ y seleccionando Mi cuenta.

Creación de un usuario con acceso administrativo

Después de registrarse para obtener una Cuenta de AWS, proteja su Usuario raíz de la cuenta de AWS, habilite AWS IAM Identity Center y cree un usuario administrativo para no utilizar el usuario raíz en las tareas cotidianas.

Protección de Usuario raíz de la cuenta de AWS

1. Inicie sesión en AWS Management Console como propietario de la cuenta; para ello, elija Usuario raíz e introduzca el correo electrónico de su Cuenta de AWS. En la siguiente página, escriba su contraseña.

   Para obtener ayuda para iniciar sesión con el usuario raíz, consulte Signing in as the root user en la Guía del usuario de AWS Sign-In.
2. Active la autenticación multifactor (MFA) para el usuario raíz.

   Para obtener instrucciones, consulte Enabling a virtual multi-factor authentication (MFA) device (console) de Cuenta de AWS en la Guía del usuario de IAM.

Creación de un usuario con acceso administrativo

1. Activar IAM Identity Center

   Consulte las instrucciones en Enabling AWS IAM Identity Center en la Guía del usuario de AWS IAM Identity Center.
2. En IAM Identity Center, conceda acceso administrativo a un usuario.

   Para ver un tutorial sobre cómo utilizar Directorio de IAM Identity Center como origen de identidad, consulte Configuración del acceso de los usuarios con el Directorio de IAM Identity Center predeterminado en la Guía del usuario de AWS IAM Identity Center.

Inicio de sesión como usuario con acceso administrativo

- Para iniciar sesión con el usuario de IAM Identity Center, utilice la URL de inicio de sesión que se envió a la dirección de correo electrónico cuando creó el usuario de IAM Identity Center.

  Para obtener ayuda para iniciar sesión con un usuario del IAM Identity Center, consulte Inicio de sesión en el portal de acceso de AWS en la Guía del usuario de AWS Sign-In.

Concesión de acceso a usuarios adicionales

1. En IAM Identity Center, cree un conjunto de permisos que siga la práctica recomendada de aplicar permisos de privilegios mínimos.

Para conocer las instrucciones, consulte Create a permission set en la Guía del usuario de AWS IAM Identity Center.

2. Asigne usuarios a un grupo y, a continuación, asigne el acceso de inicio de sesión único al grupo.

   Para conocer las instrucciones, consulte Add groups en la Guía del usuario de AWS IAM Identity Center.

Inicie sesión en la consola de Amazon CloudWatch

Para iniciar sesión en la consola de Amazon CloudWatch

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. Si es necesario, utilice la barra de navegación para cambiar la Región por la Región en la que cuenta con los recursos de AWS.
3. Incluso si es la primera vez que utiliza la consola de CloudWatch, Your Metrics (Sus métricas) ya podría registrar métricas, dado que ha usado un producto de AWS que publica automáticamente las métricas en Amazon CloudWatch de forma gratuita. Otros servicios requieren que habilite las métricas.

   Si no tiene ninguna alarma, la sección Your Alarms incluirá un botón Create Alarm.

Configuración de la AWS CLI

Puede utilizar la AWS CLI o la CLI de Amazon CloudWatch para ejecutar comandos de CloudWatch. Tenga en cuenta que la AWS CLI reemplaza la CLI de CloudWatch y que se han incluido nuevas características de CloudWatch exclusivamente en la AWS CLI.

Para obtener más información sobre cómo instalar y configurar la AWS CLI, consulte Configuración inicial de la interfaz de línea de comandos de AWS en la Guía del usuario de la AWS Command Line Interface.

Para obtener más información sobre cómo se instalar y configura la CLI de Amazon CloudWatch, consulte Set Up the Command Line Interface (Configurar la interfaz de línea de comandos) en la Referencia de la CLI de Amazon CloudWatch.

Introducción a Amazon CloudWatch

Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.

Aparece la página de inicio de información general de CloudWatch.

La información general muestra los siguientes elementos, actualizados de forma automática.

- Alarmas por servicio de AWS muestra una lista de los servicios de AWS que utiliza en su cuenta, junto con el estado de las alarmas de esos servicios. Junto a eso, aparecen dos o cuatro alarmas en su cuenta. El número depende de la cantidad de servicios de AWS que utilice. Las alarmas que se muestran son las que tienen el estado ALARM o las que han cambiado de estado más recientemente.

  Estas áreas superiores le permiten evaluar de forma rápida el estado de sus servicios de AWS, al ver los estados de alarma en cada servicio y las alarmas que han cambiado de estado más recientemente. Esto le ayuda a monitorizar y diagnosticar problemas rápidamente.
- Debajo de estas áreas se encuentra el panel predeterminado, si existe. El panel predeterminado es un panel personalizado que ha creado y denominado CloudWatch-Default. Se trata de una forma cómoda para incluir métricas sobre sus propios servicios o aplicaciones personalizados en la página de información general o para adelantar métricas de claves adicionales desde servicios de AWS que más desea monitorizar.

> **Note**
>
> Los paneles automáticos de la página de inicio de CloudWatch muestran solo la información de la cuenta actual, incluso si la cuenta es una cuenta de supervisión configurada para la observabilidad entre cuentas de CloudWatch. Para obtener más información sobre la creación de paneles personalizados entre cuentas, consulte Panel para la observabilidad entre cuentas de CloudWatch.

A partir de este resumen, puede ver un panel de varios servicios con métricas de varios servicios de AWS, o centrar su vista en un grupo de recursos o un servicio de AWS específicos. De este modo, puede reducir la vista a un subconjunto de recursos en el que esté interesado. Para obtener más información, consulte las siguientes secciones.

Consulte el panel automático prediseñado para un solo servicio

Para ver el panel automático prediseñado para un solo servicio

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.

   Aparece la página de inicio.
2. En el panel de navegación izquierdo, elija Dashboard (Paneles).
3. Seleccione la pestaña Automatic dashboards (Paneles automáticos) y, a continuación, elija el servicio que desee ver.
4. Para pasar a ver las alarmas de este servicio, active la casilla de verificación En alarma, Datos insuficientes o Aceptar cerca de la parte superior de la pantalla, donde aparece el nombre del servicio en ese momento.
5. Al visualizar las métricas, puede centrarse en una métrica determinada de varias maneras:

   a. Para ver más información acerca de las métricas en cualquier gráfico, pase el puntero por el gráfico y haga clic en el icono de acciones, View in metrics (Ver en métricas).

      El gráfico aparece en una nueva pestaña, con las métricas pertinentes que se indican debajo del gráfico. Puede personalizar la vista de este gráfico, cambiando las métricas y los recursos mostrados, la estadística, el periodo y otros factores para obtener una mejor comprensión de la situación actual.

b. Puede ver los eventos de registros desde el intervalo de tiempo que se muestra en el gráfico. Esto puede ayudarle a descubrir los eventos que ocurrieron en su infraestructura que están provocando un cambio inesperado de las métricas.

   Para ver los eventos de registro, pase el puntero por el gráfico de alarma y elija el icono de acciones, View en logs (Ver en registros).

   La vista de CloudWatch Logs aparece en una pestaña nueva, en la que se muestra una lista de los grupos de registros. Para ver los eventos de registro en uno de estos grupos de registros que han ocurrido durante el intervalo de tiempo que se muestra en el gráfico original, elija ese grupo de registros.

6. Al visualizar las alarmas, puede centrarse en una alarma determinada de varias maneras:

   - Para ver más información acerca de una alarma, pase el puntero por la alarma, y elija el icono de acciones, View in alarms (Ver en alarmas).

     La vista de alarmas aparece en una nueva pestaña, en la que se muestra una lista de las alarmas, junto con información detallada acerca de la alarma elegida. Para ver el historial de esta alarma, elija la pestaña History (Historial).

7. Las alarmas siempre se actualizan una vez por minuto. Para actualizar la vista, elija el icono de actualización (dos flechas curvadas) en la parte superior derecha de la pantalla. Para cambiar la frecuencia de actualización automática de los elementos de la pantalla que no sean alarmas, elija la flecha hacia abajo junto al icono de actualización y elija la velocidad de actualización. También puede elegir desactivar la actualización automática.
8. Para cambiar el intervalo de tiempo mostrado en todos los gráficos y alarmas que se muestran actualmente, al lado de Time range (Intervalo de tiempo) en la parte superior de la pantalla, elija el intervalo. Para seleccionar entre más opciones de intervalo de tiempo que las que se muestran de forma predeterminada, elija custom (personalizado) .
9. Para volver al panel de varios servicios, elija Overview (Información general) en la lista de la parte superior de la pantalla que actualmente muestra el servicio en el que se centra.

   De forma alternativa, desde cualquier vista, puede elegir CloudWatch en la parte superior de la pantalla para quitar todos los filtros y volver a la página Overview (Información general).

Consulte el panel de varios servicios prediseñado

Puede pasar a la pantalla Cross-service dashboard (Panel de varios servicios) e interactuar con los paneles de todos los servicios de AWS que está utilizando. La consola de CloudWatch muestra los paneles en orden alfabético, además de una o dos métricas clave en cada panel.

> **Note**
>
> Si está utilizando cinco o más servicios de AWS, la consola de CloudWatch no mostrará el Cross-service dashboard (Panel de varios servicios) en la pantalla Overview (Información general).

Para abrir el Cross-service dashboard (Panel de varios servicios)

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.

   Esto lo llevará a la pantalla Overview (Información general).
2. En la pantalla Overview (Información general), seleccione el menú desplegable que dice Overview (Información general) y, luego, seleccione Cross service dashboard (Panel de varios servicios).

   Esto lo llevará a la pantalla Cross-service dashboard (Panel de varios servicios).
3. (Opcional) Si está utilizando la interfaz original, desplácese hasta la sección Cross-service dashboard (Panel de varios servicios) y, luego, elija View Cross-service dashboard (Ver el panel de varios servicios).

   Esto lo llevará a la pantalla Cross-service dashboard (Panel de varios servicios).
4. Puede centrarse en un servicio determinado de dos formas:

   a. Para ver más las métricas clave para un servicio, elija el nombre en la lista de la parte superior de la pantalla, donde el panel de varios servicios se muestra actualmente. O bien, puede elegir View Service dashboard (Ver panel de servicios) al lado del nombre de servicio.

      Aparece un panel automática para dicho servicio con más métricas para dicho servicio. Además, para algunos servicios, la parte inferior del panel de servicios muestra recursos relacionados con dicho servicio. Puede elegir uno de dichos recursos para dicha consola de servicio y centrarse más en dicho recurso.

b. Para ver todas las alarmas relacionadas con un servicio, elija el botón situado a la derecha de la pantalla junto a dicho nombre de servicio. El texto en este botón indica la cantidad de alarmas que ha creado en este servicio, y si se encuentran en el estado ALARM.

   Cuando se muestran las alarmas, puede mostrarse varias alarmas que tienen ajustes similares (como dimensiones, umbral o periodo) en un solo gráfico.

   A continuación, puede ver los detalles de una alarma y ver el historial de alarmas. Para hacerlo, pase el puntero por el gráfico de alarma y elija el icono de acciones, View en alarms (Ver en alarmas).

   La vista de alarmas aparece en una nueva pestaña del navegador, en la que se muestra una lista de las alarmas, junto con información detallada acerca de la alarma elegida. Para ver el historial de esta alarma, elija la pestaña History (Historial).

5. Puede centrarse en los recursos de un grupo de recursos concreto. Para ello, elija el grupo de recursos de la lista en la parte superior de la página donde se muestra All resources (Todos los recursos).

   Para obtener más información, consulte Consulte un panel prediseñado para un grupo de recursos.

6. Para cambiar el intervalo de tiempo mostrado en todos los gráficos y alarmas que se muestran actualmente, seleccione el intervalo que desea al lado de Time range (Intervalo de tiempo) en la parte superior de la pantalla. Elija custom (personalizado) para seleccionar entre más opciones de intervalo de tiempo que las que se muestran de forma predeterminada.

7. Las alarmas siempre se actualizan una vez por minuto. Para actualizar la vista, elija el icono de actualización (dos flechas curvadas) en la parte superior derecha de la pantalla. Para cambiar la frecuencia de actualización automática de los elementos de la pantalla que no sean alarmas, elija la flecha hacia abajo junto al icono de actualización y elija la velocidad de actualización que desea. También puede elegir desactivar la actualización automática.

Evite que un servicio aparezca en el panel de varios servicios

Puede evitar que las métricas de un servicio aparezcan en el panel de varios servicios. Esto le ayuda a centrar el panel de varios servicios en los servicios que más desee monitorear.

Si elimina un servicio del panel de varios servicios, las alarmas para dicho servicio seguirán apareciendo en las vistas de las alarmas.

Para eliminar las métricas de un servicio del panel de varios servicios

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.

   Aparece la página de inicio.
2. En la parte superior de la página, en Overview (Información general), elija el servicio que desea quitar.

   La vista cambia para mostrar métricas solo de dicho servicio.
3. Elija Actions (Acciones), a continuación desactive la casilla situada junto a Show on cross service dashboard (Mostrar en panel de varios servicios).

Consulte un panel prediseñado para un grupo de recursos

Puede centrar su vista para ver métricas y alarmas desde un único grupo de recursos. Usar grupos de recursos permite utilizar etiquetas para organizar proyectos, centrarse en un subconjunto de su arquitectura o distinguir entre los entornos de producción y desarrollo. También permiten centrarse en cada uno de estos grupos de recursos en la información general de CloudWatch. Para obtener más información, consulte ¿Qué es AWS Resource Groups?

Al centrarse en un grupo de recursos, la pantalla cambia para mostrar solo los servicios en los que tiene recursos etiquetados como parte de este grupo de recursos. El área de alarmas de uso reciente muestra únicamente las alarmas asociadas con los recursos que forman parte del grupo de recursos. Además, si ha creado un panel con el nombre CloudWatch-Default-ResourceGroupName, se muestra en el área Default dashboard (Panel predeterminado).

Puede bajar más centrándose tanto en un solo servicio de AWS como en un grupo de recursos al mismo tiempo. En el siguiente procedimiento solo se muestra cómo centrarse en un grupo de recursos.

Para centrarse en un grupo de recursos

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En la parte superior de la página donde se muestra All resources (Todos los recursos), elija un grupo de recursos.
3. Para ver varias métricas relacionados con este grupo de recursos, cerca de la parte inferior de la pantallpanel de varios serviciosja View cross service dashboard (Ver panel de varios servicios).

El panel de varios servicios aparece, mostrando solo los servicios relacionados con este grupo de recursos. Por cada servicio, se muestra uno o dos métricas clave.

4. Para cambiar el intervalo de tiempo mostrado en todos los gráficos y alarmas que se muestran actualmente, para Time range (Intervalo de tiempo) en la parte superior de la pantalla, seleccione un intervalo. Para seleccionar entre más opciones de intervalo de tiempo que las que se muestran de forma predeterminada, elija custom (personalizado) .
5. Las alarmas siempre se actualizan una vez por minuto. Para actualizar la vista, elija el icono de actualización (dos flechas curvadas) en la parte superior derecha de la pantalla. Para cambiar la frecuencia de actualización automática de los elementos de la pantalla que no sean alarmas, elija la flecha hacia abajo junto al icono de actualización y elija la velocidad de actualización. También puede elegir desactivar la actualización automática.
6. Para volver a mostrar información sobre todos los recursos de su cuenta, cerca de la parte superior de la pantalla donde el nombre del grupo de recursos se muestra actualmente, elija All resources (Todos los recursos).

Consulte el panel de varios servicios prediseñado

Puede pasar a la pantalla Cross-service dashboard (Panel de varios servicios) e interactuar con los paneles de todos los servicios de AWS que está utilizando. La consola de CloudWatch muestra los paneles en orden alfabético, además de una o dos métricas clave de cada panel.

> **Note**
>
> Si está utilizando cinco o más servicios de AWS, la consola de CloudWatch no mostrará el Cross-service dashboard (Panel de varios servicios) en la pantalla Overview (Información general).

Para abrir el Cross-service dashboard (Panel de varios servicios)

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.

   Esto lo llevará a la pantalla Overview (Información general).
2. En la pantalla Overview (Información general), seleccione el menú desplegable que dice Overview (Información general) y, luego, seleccione Cross service dashboard (Panel de varios servicios).

Esto lo llevará a la pantalla Cross-service dashboard (Panel de varios servicios).

3. (Opcional) Si está utilizando la interfaz original, desplácese hasta la sección Cross-service dashboard (Panel de varios servicios) y, luego, elija View Cross-service dashboard (Ver el panel de varios servicios).

   Esto lo llevará a la pantalla Cross-service dashboard (Panel de varios servicios).

4. Puede centrarse en un servicio determinado de dos formas:

   a. Para ver más las métricas clave para un servicio, elija el nombre en la lista de la parte superior de la pantalla, donde el panel de varios servicios se muestra actualmente. O bien, puede elegir View Service dashboard (Ver panel de servicios) al lado del nombre de servicio.

      Aparece un panel automática para dicho servicio con más métricas para dicho servicio. Además, para algunos servicios, la parte inferior del panel de servicios muestra recursos relacionados con dicho servicio. Puede elegir uno de dichos recursos para dicha consola de servicio y centrarse más en dicho recurso.

   b. Para ver todas las alarmas relacionadas con un servicio, elija el botón situado a la derecha de la pantalla junto a dicho nombre de servicio. El texto en este botón indica la cantidad de alarmas que ha creado en este servicio, y si se encuentran en el estado ALARM.

      Cuando se muestran las alarmas, puede mostrarse varias alarmas que tienen ajustes similares (como dimensiones, umbral o periodo) en un solo gráfico.

      A continuación, puede ver los detalles de una alarma y ver el historial de alarmas. Para hacerlo, pase el puntero por el gráfico de alarma y elija el icono de acciones, View en alarms (Ver en alarmas).

      La vista de alarmas aparece en una nueva pestaña del navegador, en la que se muestra una lista de las alarmas, junto con información detallada acerca de la alarma elegida. Para ver el historial de esta alarma, elija la pestaña History (Historial).

5. Puede centrarse en los recursos de un grupo de recursos concreto. Para ello, elija el grupo de recursos de la lista en la parte superior de la página donde se muestra All resources (Todos los recursos).

   Para obtener más información, consulte Consulte un panel prediseñado para un grupo de recursos.

6. Para cambiar el intervalo de tiempo mostrado en todos los gráficos y alarmas que se muestran actualmente, seleccione el intervalo que desea al lado de Time range (Intervalo de tiempo) en la parte superior de la pantalla. Elija custom (personalizado) para seleccionar entre más opciones de intervalo de tiempo que las que se muestran de forma predeterminada.
7. Las alarmas siempre se actualizan una vez por minuto. Para actualizar la vista, elija el icono de actualización (dos flechas curvadas) en la parte superior derecha de la pantalla. Para cambiar la frecuencia de actualización automática de los elementos de la pantalla que no sean alarmas, elija la flecha hacia abajo junto al icono de actualización y elija la velocidad de actualización que desea. También puede elegir desactivar la actualización automática.

Evite que un servicio aparezca en el panel de varios servicios

Puede evitar que las métricas de un servicio aparezcan en el panel de varios servicios. Esto le ayuda a centrar el panel de varios servicios en los servicios que más desee monitorear.

Si elimina un servicio del panel de varios servicios, las alarmas para dicho servicio seguirán apareciendo en las vistas de las alarmas.

Para eliminar las métricas de un servicio del panel de varios servicios

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.

   Aparece la página de inicio.
2. En la parte superior de la página, en Overview (Información general), elija el servicio que desea quitar.

   La vista cambia para mostrar métricas solo de dicho servicio.
3. Elija Actions (Acciones), a continuación desactive la casilla situada junto a Show on cross service dashboard (Mostrar en panel de varios servicios).

Consulte un panel prediseñado para un solo servicio de AWS

En la página de inicio de CloudWatch, puede centrar la vista en un solo servicio de AWS. Puede bajar más centrándose tanto en un solo servicio de AWS como en un grupo de recursos al mismo tiempo. En el siguiente procedimiento solo se muestra cómo centrarse en un servicio de AWS.

Para centrarse en un solo servicio

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.

   Aparece la página de inicio.
2. En Descripción general, donde actualmente se muestra la descripción general en el menú desplegable, elija Paneles de servicio.
3. Elija el servicio en el que desee centrarse.

   La vista cambia para mostrar gráficos de las métricas clave del servicio seleccionado.
4. Para pasar a ver las alarmas de este servicio, active la casilla de verificación En alarma, Datos insuficientes o Aceptar cerca de la parte superior de la pantalla, donde aparece el nombre del servicio en ese momento.
5. Al visualizar las métricas, puede centrarse en una métrica determinada de varias maneras:

   a. Para ver más información acerca de las métricas en cualquier gráfico, pase el puntero por el gráfico y haga clic en el icono de acciones, View in metrics (Ver en métricas).

      El gráfico aparece en una nueva pestaña, con las métricas pertinentes que se indican debajo del gráfico. Puede personalizar la vista de este gráfico, cambiando las métricas y los recursos mostrados, la estadística, el periodo y otros factores para obtener una mejor comprensión de la situación actual.

   b. Puede ver los eventos de registros desde el intervalo de tiempo que se muestra en el gráfico. Esto puede ayudarle a descubrir los eventos que ocurrieron en su infraestructura que están provocando un cambio inesperado de las métricas.

      Para ver los eventos de registro, pase el puntero por el gráfico de alarma y elija el icono de acciones, View en logs (Ver en registros).

      La vista de CloudWatch Logs aparece en una pestaña nueva, en la que se muestra una lista de los grupos de registros. Para ver los eventos de registro en uno de estos grupos de registros que han ocurrido durante el intervalo de tiempo que se muestra en el gráfico original, elija ese grupo de registros.
6. Al visualizar las alarmas, puede centrarse en una alarma determinada de varias maneras:

   - Para ver más información acerca de una alarma, pase el puntero por la alarma, y elija el icono de acciones, View in alarms (Ver en alarmas).

La vista de alarmas aparece en una nueva pestaña, en la que se muestra una lista de las alarmas, junto con información detallada acerca de la alarma elegida. Para ver el historial de esta alarma, elija la pestaña History (Historial).

7. Las alarmas siempre se actualizan una vez por minuto. Para actualizar la vista, elija el icono de actualización (dos flechas curvadas) en la parte superior derecha de la pantalla. Para cambiar la frecuencia de actualización automática de los elementos de la pantalla que no sean alarmas, elija la flecha hacia abajo junto al icono de actualización y elija la velocidad de actualización. También puede elegir desactivar la actualización automática.
8. Para cambiar el intervalo de tiempo mostrado en todos los gráficos y alarmas que se muestran actualmente, al lado de Time range (Intervalo de tiempo) en la parte superior de la pantalla, elija el intervalo. Para seleccionar entre más opciones de intervalo de tiempo que las que se muestran de forma predeterminada, elija custom (personalizado) .
9. Para volver al panel de varios servicios, elija Overview (Información general) en la lista de la parte superior de la pantalla que actualmente muestra el servicio en el que se centra.

   De forma alternativa, desde cualquier vista, puede elegir CloudWatch en la parte superior de la pantalla para quitar todos los filtros y volver a la página Overview (Información general).

Consulte un panel prediseñado para un grupo de recursos

Puede centrar su vista para ver métricas y alarmas desde un único grupo de recursos. Usar grupos de recursos permite utilizar etiquetas para organizar proyectos, centrarse en un subconjunto de su arquitectura o distinguir entre los entornos de producción y desarrollo. También permiten centrarse en cada uno de estos grupos de recursos en la información general de CloudWatch. Para obtener más información, consulte ¿Qué es AWS Resource Groups?

Al centrarse en un grupo de recursos, la pantalla cambia para mostrar solo los servicios en los que tiene recursos etiquetados como parte de este grupo de recursos. El área de alarmas de uso reciente muestra únicamente las alarmas asociadas con los recursos que forman parte del grupo de recursos. Además, si ha creado un panel con el nombre CloudWatch-Default-ResourceGroupName, se muestra en el área Default dashboard (Panel predeterminado).

Puede bajar más centrándose tanto en un solo servicio de AWS como en un grupo de recursos al mismo tiempo. En el siguiente procedimiento solo se muestra cómo centrarse en un grupo de recursos.

Para centrarse en un grupo de recursos

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En la parte superior de la página donde se muestra All resources (Todos los recursos), elija un grupo de recursos.
3. Para ver varias métricas relacionados con este grupo de recursos, cerca de la parte inferior de la pantallpanel de varios serviciosja View cross service dashboard (Ver panel de varios servicios).

   El panel de varios servicios aparece, mostrando solo los servicios relacionados con este grupo de recursos. Por cada servicio, se muestra uno o dos métricas clave.
4. Para cambiar el intervalo de tiempo mostrado en todos los gráficos y alarmas que se muestran actualmente, para Time range (Intervalo de tiempo) en la parte superior de la pantalla, seleccione un intervalo. Para seleccionar entre más opciones de intervalo de tiempo que las que se muestran de forma predeterminada, elija custom (personalizado) .
5. Las alarmas siempre se actualizan una vez por minuto. Para actualizar la vista, elija el icono de actualización (dos flechas curvadas) en la parte superior derecha de la pantalla. Para cambiar la frecuencia de actualización automática de los elementos de la pantalla que no sean alarmas, elija la flecha hacia abajo junto al icono de actualización y elija la velocidad de actualización. También puede elegir desactivar la actualización automática.
6. Para volver a mostrar información sobre todos los recursos de su cuenta, cerca de la parte superior de la pantalla donde el nombre del grupo de recursos se muestra actualmente, elija All resources (Todos los recursos).

Facturación y costo de CloudWatch

En esta sección se describe cómo las funciones de Amazon CloudWatch generan costos. También proporciona métodos que pueden ayudar a analizar, optimizar y reducir los costos de CloudWatch. En esta sección, a veces nos referimos a los precios al describir las características de CloudWatch. Para obtener información acerca de los precios, consulte Precios de Amazon CloudWatch.

Temas

Analice los datos de uso y costo de CloudWatch con Cost Explorer

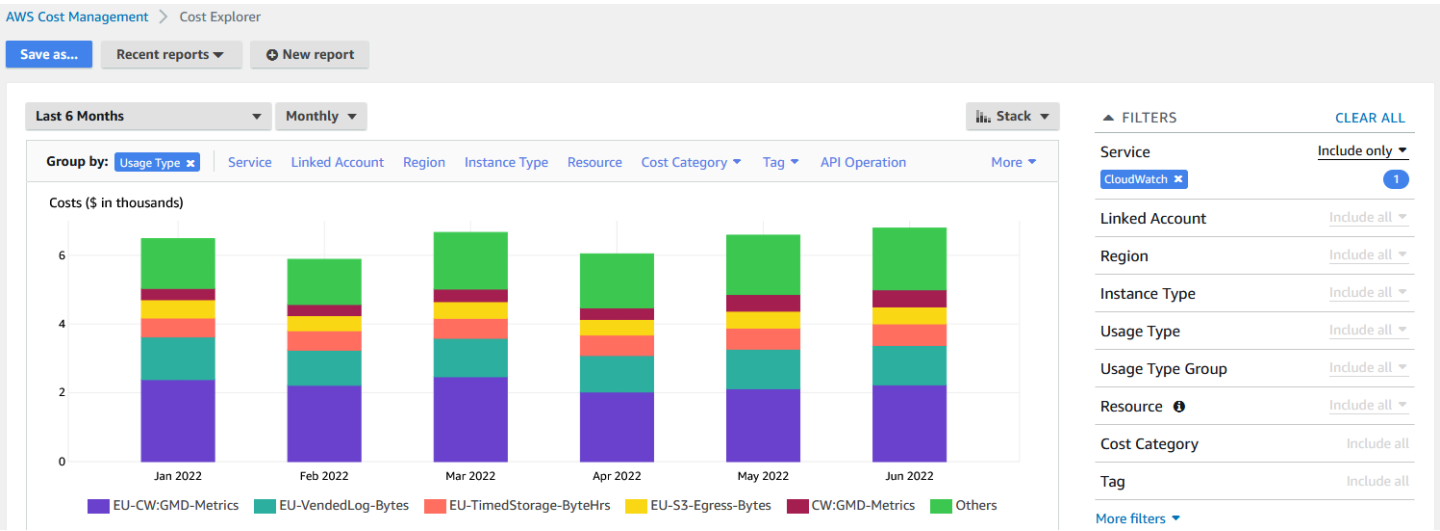
Con AWS Cost Explorer, puede visualizar y analizar los datos de costo y uso de Servicios de AWS a lo largo del tiempo, incluido CloudWatch. Para obtener más información, consulte Introducción a AWS Cost Explorer.

El siguiente procedimiento describe cómo usar Cost Explorer para visualizar y analizar los datos de uso y costo de CloudWatch.

Visualizar y analizar los datos de uso y costo de CloudWatch

1. Inicie sesión en la consola de Cost Explorer en https://console.aws.amazon.com/cost-management/home#/custom.
2. En FILTERS (FILTROS), para Service (Servicio), seleccione CloudWatch.
3. Para Group by (Agrupar por), seleccione Usage Type (Tipo de uso). También puede agrupar los resultados por otras categorías, como las siguientes:
   - API Operation (Operación de la API): vea qué operaciones de la API generaron más costos.
   - Region (Región): vea qué regiones generaron más costos.

En la siguiente imagen se muestra un ejemplo de los costos que las funciones de CloudWatch generaron a lo largo de seis meses.



Para ver qué características de CloudWatch generaron más costos, consulte los valores de `UsageType`. Por ejemplo, `EU-CW:GMD-Metrics` representa los costos que generaron las solicitudes de API masivas de CloudWatch.

> **Note**
>
> Las cadenas de `UsageType` coinciden con características y regiones específicas. Por ejemplo, la primera parte de `EU-CW:GMD-Metrics` (`EU`) coincide con la región Europa (Irlanda), y la segunda parte de `EU-CW:GMD-Metrics` (`GMD-Metrics`) coincide con las solicitudes API masivas de CloudWatch.
>
> La cadena completa de `UsageType` puede formatearse de la siguiente manera: `<Region>-CW:<Feature>` o `<Region>-<Feature>`.
>
> Para mejorar la legibilidad, las cadenas de `UsageType` en las tablas de este documento se han reducido a sus sufijos de cadena. Por ejemplo, `EU-CW:GMD-Metrics` se acorta a `GMD-Metrics`.

En la siguiente tabla, se incluyen los nombres de cada característica CloudWatch, se enumeran los nombres de cada subfunción y se enumeran las cadenas de `UsageType`.

Característica CloudWatch	Característica secundaria de CloudWatch	**`UsageType`**
Métricas de CloudWatch	Métricas personalizadas	`MetricMonitorUsage`

Característica CloudWatch	Característica secundaria de CloudWatch	**`UsageType`**
	Monitoreo detallado	`MetricMonitorUsage`
	Métricas integradas	`MetricMonitorUsage`
Solicitudes de API de CloudWatch	Solicitudes API	`Requests`
	En bloque (Obtener)	`GMD-Metrics`
	Contributor Insights	`GIRR-Metrics`
	Instantánea de imagen de mapa de bits	`GMWI-Metrics`
Flujos métricos de CloudWatch	Flujos métricos	`MetricStreamUsage`
Paneles de CloudWatch	Panel de control con 50 métricas o menos	`DashboardsUsageHour-Basic`
	Panel de control con más de 50 métricas	`DashboardsUsageHour`
Alarmas de CloudWatch	Estándar (alarma métrica)	`AlarmMonitorUsage`
	Alta resolución (alarma métrica)	`HighResAlarmMonitorUsage`
	Alarma de consulta de Información de métricas	`MetricInsightAlarmUsage`

Característica CloudWatch	Característica secundaria de CloudWatch	**UsageType**
	Compuesto (alarma agregada)	`CompositeAlarmMonitorUsage`
Señales de aplicación de CloudWatch	Señales de aplicación	`Application-Signals`
Registros personalizados de CloudWatch	Recopilar (ingerir)	`DataProcessing-Bytes`
	Almacenar (archivar)	`TimedStorage-ByteHrs`
	Analizar (consultar)	`DataScanned-Bytes`
Registros de acceso poco frecuente de CloudWatch	Recopilar (ingerir)	`DataProcessingIA-Bytes`
Registros ofrecidos por CloudWatch	Entrega (Registros de Amazon CloudWatch)	`VendedLog-Bytes`
	Entrega (Registros de acceso poco frecuente de CloudWatch)	`VendedLogIA-Bytes`
	Entrega (Amazon Simple Storage Service)	`S3-Egress-ComprBytes` `S3-Egress-Bytes`
	Entrega (Amazon Data Firehose)	`FH-Egress-Bytes`
Contributor Insights	Registros de CloudWatch (reglas)	`ContributorInsightRules`

Característica CloudWatch	Característica secundaria de CloudWatch	**`UsageType`**
	Registros de CloudWatch (Eventos)	`ContributorInsightEvents`
	Amazon DynamoDB (reglas)	`ContributorRulesManaged`
	DynamoDB (eventos)	`ContributorEventsManaged`
Valores controlados (Synthetics)	Ejecute	`Canary-runs`
Evidently	Eventos	`Evidently-event`
	Unidades de análisis	`Evidently-eau`
RUM	Eventos	`RUM-event`

Analice los datos de uso y costo de CloudWatch con AWS Cost and Usage Report y Athena

También es posible analizar los datos de uso y costo de CloudWatch mediante AWS Cost and Usage Report y con Amazon Athena. AWS Cost and Usage Report contiene un conjunto completo de datos de costo y uso. Puede crear informes que realicen un seguimiento de sus costos y uso, y puede publicar estos informes en un bucket de S3 de su elección. También puede descargar y eliminar sus informes de su bucket de S3. Para obtener más información, consulte ¿Qué es AWS Cost and Usage Report? en la Guía del usuario de AWS Cost and Usage Report.

> **Note**
>
> El uso de AWS Cost and Usage Report no implica cargos adicionales. Solo paga por el almacenamiento cuando publica sus informes en Amazon Simple Storage Service (Amazon S3). Para obtener más información, consulte Quotas and restrictions (Cuotas y limitaciones) en la Guía del usuario de AWS Cost and Usage Report.

Athena es un servicio de consultas que puede usar con AWS Cost and Usage Report para analizar los datos de costo y uso. Puede consultar sus informes en su bucket de S3 sin necesidad de descargarlos primero. Para obtener más información, consulte ¿Qué es Amazon Athena? en la Guía del usuario de Amazon Athena. Para obtener más información, consulte ¿Qué es Amazon Athena? en la Guía del usuario de Amazon Athena. Para obtener información, consulte Precios de Amazon Athena.

El siguiente procedimiento describe el proceso para habilitar AWS Cost and Usage Report e integrar el servicio con Athena. El procedimiento contiene dos consultas de ejemplo que puede utilizar para analizar los datos de uso y costo de CloudWatch.

> **Note**
>
> Puede utilizar cualquiera de las consultas de ejemplo de este documento. Todas las consultas de ejemplo de este documento corresponden a una base de datos denominada costandusagereport y muestran resultados del mes de abril y del año 2022. Puede cambiar esta información. Sin embargo, antes de ejecutar una consulta, asegúrese de que el nombre de la base de datos coincida con el nombre de la base de datos de la consulta.

Analizar los datos de costos y uso con AWS Cost and Usage Report y Athena

1. Habilite AWS Cost and Usage Report. Para obtener más información, consulte Creacion de informes de costo y uso en la Guía de informes de uso y costo de AWS Cost and Usage Report.

> **Tip**
>
> Cuando cree sus informes, asegúrese de seleccionar Incluir ID de recurso. De lo contrario, los informes no incluirán la columna `line_item_resource_id`. Esta línea le ayuda a identificar mejor los costos al analizar los datos de costos y uso.

2. Integre AWS Cost and Usage Report con Athena Para obtener más información, consulte Configurar Athena mediante AWS CloudFormation plantillas en la Guía del usuario de AWS Cost and Usage Report.
3. Consulte sus informes de costos y uso.

Ejemplo: consulta de Athena

Puede utilizar la siguiente consulta para mostrar qué características de CloudWatch generaron la mayor cantidad de costos en un mes determinado.

```
SELECT
CASE
-- Metrics
WHEN line_item_usage_type LIKE '%%MetricMonitorUsage%%' THEN 'Metrics (Custom, Detailed
 monitoring management portal EMF)'
WHEN line_item_usage_type LIKE '%%Requests%%' THEN 'Metrics (API Requests)'
WHEN line_item_usage_type LIKE '%%GMD-Metrics%%' THEN 'Metrics (Bulk API Requests)'
WHEN line_item_usage_type LIKE '%%MetricStreamUsage%%' THEN 'Metric Streams'
-- Dashboard
WHEN line_item_usage_type LIKE '%%DashboardsUsageHour%%' THEN 'Dashboards'
-- Alarms
WHEN line_item_usage_type LIKE '%%AlarmMonitorUsage%%' THEN 'Alarms (Standard)'
WHEN line_item_usage_type LIKE '%%HighResAlarmMonitorUsage%%' THEN 'Alarms (High
 Resolution)'
WHEN line_item_usage_type LIKE '%%MetricInsightAlarmUsage%%' THEN 'Alarms (Metrics
 Insights)'
WHEN line_item_usage_type LIKE '%%CompositeAlarmMonitorUsage%%' THEN 'Alarms
 (Composite)'
-- Logs
WHEN line_item_usage_type LIKE '%%DataProcessing-Bytes%%' THEN 'Logs (Collect - Data
 Ingestion)'
-- Logs
WHEN line_item_usage_type LIKE '%%DataProcessingIA-Bytes%%' THEN 'Infrequent Access
 Logs (Collect - Data Ingestion)'
```

```
WHEN line_item_usage_type LIKE '%%TimedStorage-ByteHrs%%' THEN 'Logs (Storage -
 Archival)'
WHEN line_item_usage_type LIKE '%%DataScanned-Bytes%%' THEN 'Logs (Analyze - Logs
 Insights queries)'
-- Vended Logs
WHEN line_item_usage_type LIKE '%%VendedLog-Bytes%%' THEN 'Vended Logs (Delivered to
 CW)'
WHEN line_item_usage_type LIKE '%%VendedLogIA-Bytes%%' THEN 'Vended Infrequent Access
 Logs (Delivered to CW)'
WHEN line_item_usage_type LIKE '%%FH-Egress-Bytes%%' THEN 'Vended Logs (Delivered to
 Kinesis FH)'
WHEN (line_item_usage_type LIKE '%%S3-Egress-Bytes%%') OR (line_item_usage_type LIKE '%
%S3-Egress-
ComprBytes%%') THEN 'Vended Logs (Delivered to S3)'
-- Other
WHEN line_item_usage_type LIKE '%%Application-Signals%%' THEN 'Application Signals'
WHEN line_item_usage_type LIKE '%%Canary-runs%%' THEN 'Synthetics'
WHEN line_item_usage_type LIKE '%%Evidently%%' THEN 'Evidently'
WHEN line_item_usage_type LIKE '%%RUM-event%%' THEN 'RUM'
ELSE 'Others'
END AS UsageType,
-- REGEXP_EXTRACT(line_item_resource_id,'^(?:.+?:){5}(.+)$',1) as ResourceID,
-- SUM(CAST(line_item_usage_amount AS double)) AS UsageQuantity,
SUM(CAST(line_item_unblended_cost AS decimal(16,8))) AS TotalSpend
FROM
costandusagereport
WHERE
product_product_name = 'AmazonCloudWatch'
AND year='2022'
AND month='4'
AND line_item_line_item_type NOT IN
 ('Tax','Credit','Refund','EdpDiscount','Fee','RIFee')
-- AND line_item_usage_account_id = '123456789012' - If you want to filter on a
 specific account, you can
remove this comment at the beginning of the line and specify an AWS account.
GROUP BY
1
ORDER BY
TotalSpend DESC,
UsageType;
```

Ejemplo: consulta de Athena

Puede utilizar la siguiente consulta para mostrar los resultados de`UsageType` y `Operation`. Esto le muestra cómo las características de CloudWatch generaron costos. Los resultados también muestran los valores de `UsageQuantity` y `TotalSpend`, para que puedas ver sus costos de uso totales.

> **Tip**
>
> Para obtener más información acerca de `UsageType`, agregue la línea siguiente a esta consulta:
> `line_item_line_item_description`
> Esta línea crea una columna llamada Description (Descripción).

```
SELECT
bill_payer_account_id as Payer,
line_item_usage_account_id as LinkedAccount,
line_item_usage_type AS UsageType,
line_item_operation AS Operation,
line_item_resource_id AS ResourceID,
SUM(CAST(line_item_usage_amount AS double)) AS UsageQuantity,
SUM(CAST(line_item_unblended_cost AS decimal(16,8))) AS TotalSpend
FROM
costandusagereport
WHERE
product_product_name = 'AmazonCloudWatch'
AND year='2022'
AND month='4'
AND line_item_line_item_type NOT IN
 ('Tax','Credit','Refund','EdpDiscount','Fee','RIFee')
GROUP BY
bill_payer_account_id,
line_item_usage_account_id,
line_item_usage_type,
line_item_resource_id,
line_item_operation
```

Prácticas recomendadas para optimizar y reducir costos

Métricas de CloudWatch

Muchos Servicios de AWS, como Amazon Elastic Compute Cloud (Amazon EC2), Amazon S3 y Amazon Data Firehose, envían métricas automáticamente a CloudWatch sin costo alguno. Sin embargo, las métricas que se agrupan en las siguientes categorías pueden generar costos adicionales:

- Métricas personalizadas, supervisión detallada y métricas integradas
- Solicitudes API
- Flujos métricos

Para obtener más información, consulte Uso de métricas de Amazon CloudWatch.

Métricas personalizadas, supervisión detallada y métricas integradas

Métricas personalizadas

Puede crear métricas personalizadas para organizar los puntos de datos en cualquier orden y velocidad.

Todas las métricas personalizadas se prorratean por hora. Solo se miden cuando se envían a CloudWatch. Para obtener información sobre los precios de las métricas, consulte Precios de Amazon CloudWatch.

En la siguiente tabla, se enumeran los nombres de las subcaracterísticas relevantes para métricas de CloudWatch. La tabla incluye las cadenas de `UsageType` y `Operation`, que puede ayudarlo a analizar e identificar los costos relacionados con las métricas.

> **Note**
>
> Para obtener más información sobre las métricas que se enumeran en la siguiente tabla mientras consulta los datos de costo y uso con Athena, haga coincidir las cadenas de `Operation` con los resultados que se muestran para `line_item_operation`.

Característica secundaria de CloudWatch	**`UsageType`**	**`Operation`**	Finalidad
Métricas personalizadas	`MetricMonitorUsage`	`MetricStorage`	Métricas personalizadas
Monitoreo detallado	`MetricMonitorUsage`	`MetricStorage:AWS/` *`{Service}`*	Monitoreo detallado
Métricas integradas	`MetricMonitorUsage`	`MetricStorage:AWS/Logs-EMF`	Métricas incrustadas de registros
Filtros de registro	`MetricMonitorUsage`	`MetricStorage:AWS/CloudWatchLogs`	Filtros de métricas de grupos de registros

Monitoreo detallado

CloudWatch tiene dos tipos de supervisión:

- Monitoreo básico

  La supervisión básica es gratuita y se habilita automáticamente para todos los Servicios de AWS que admiten la característica.
- Monitoreo detallado

  La supervisión detallada implica costos y agrega diferentes mejoras en función de la Servicio de AWS. Para cada Servicio de AWS que admita una supervisión detallada, puede elegir si desea habilitarla para ese servicio. Para obtener más información, consulte Supervisión básica y detallada.

> **Note**
>
> Otros Servicios de AWS admiten una supervisión detallada y pueden hacer referencia a esta función con un nombre diferente. Por ejemplo, la supervisión detallada de Amazon S3 se denominamétricas de solicitudes.

Al igual que en las métricas personalizadas, la supervisión detallada se prorratea por hora y se mide solo cuando los datos se envían a CloudWatch. La supervisión detallada genera costos por la cantidad de métricas que se envían a CloudWatch. Para reducir los costos, solo habilite la supervisión detallada cuando sea necesario. Para obtener información sobre el precio de la supervisión detallada, consulte Precios de Amazon CloudWatch.

Ejemplo: consulta de Athena

Puede utilizar la siguiente consulta para mostrar qué instancias EC2 tienen la supervisión detallada habilitada.

```
SELECT
bill_payer_account_id as Payer,
line_item_usage_account_id as LinkedAccount,
line_item_usage_type AS UsageType,
line_item_operation AS Operation,
line_item_resource_id AS ResourceID,
SUM(CAST(line_item_usage_amount AS double)) AS UsageQuantity,
SUM(CAST(line_item_unblended_cost AS decimal(16,8))) AS TotalSpend
FROM
costandusagereport
WHERE
product_product_name = 'AmazonCloudWatch'
AND year='2022'
AND month='4'
AND line_item_operation='MetricStorage:AWS/EC2'
AND line_item_line_item_type NOT IN
 ('Tax','Credit','Refund','EdpDiscount','Fee','RIFee')
GROUP BY
bill_payer_account_id,
line_item_usage_account_id,
line_item_usage_type,
line_item_resource_id,
line_item_operation,
line_item_line_item_description
```

```
ORDER BY line_item_operation
```

Métricas integradas

Con el formato métrico integrado de CloudWatch, puede incorporar datos de aplicaciones como datos de registro, de modo que pueda generar métricas procesables. Para obtener más información, consulte Incorporación de registros de alta cardinalidad y generación de métricas con formato de métrica integrada de CloudWatch

Las métricas incorporadas generan costos según el número de registros ingeridos, el número de registros archivados y el número de métricas personalizadas generadas.

En la siguiente tabla se enumeran los nombres de las subcaracterísticas relevantes para el formato de métrica incrustado de CloudWatch. La tabla incluye las cadenas de `UsageType` y `Operation`, que puede ayudar a analizar e identificar los costos.

Característica secundaria de CloudWatch	**`UsageType`**	**`Operation`**	Finalidad
Métricas personalizadas	`MetricMonitorUsage`	`MetricStorage:AWS/Logs-EMF`	Métricas incrustadas de registros
Incorporación de registros	`DataProcessing-Bytes`	`PutLogEvents`	Carga un lote de eventos de registro en el grupo de registro o flujo de registro especificado.
Archivo de registros	`TimedStorage-ByteHrs`	`HourlyStorageMetering`	Almacena registros por hora y registros por byte en Registros de CloudWatch

Para analizar los costos, utilice AWS Cost and Usage Report con Athena para que pueda identificar qué métricas generan costos y determinar cómo se generan los costos.

Para aprovechar al máximo los costos generados por el formato de métricas incrustadas de CloudWatch, evite crear métricas basadas en dimensiones de alta cardinalidad. De este modo, CloudWatch no crea una métrica personalizada para cada combinación de dimensiones única. Para obtener más información, consulte Dimensiones.

Si usa Información de contenedores de CloudWatch para aprovechar el formato métrico incorporado, puede usar AWS Distro para Open Telemetry como alternativa para aprovechar al máximo los costos relacionados con las métricas. Con Información de contenedores puede recopilar, agregar y resumir métricas y registros de sus aplicaciones en contenedores y microservicios. Cuando habilita Información de contenedores, el agente de CloudWatch envía los registros a CloudWatch para que pueda usarlos para generar métricas incorporadas. Sin embargo, el agente de CloudWatch solo envía una cantidad fija de métricas a CloudWatch y se le cobran todas las métricas disponibles, incluidas las que no esté utilizando. Con AWS Distro for Open Telemetry, puede configurar y personalizar qué métricas y dimensiones se envían a CloudWatch. Esto le ayuda a reducir el volumen de datos y el costo que genera Información de contenedores. Para obtener más información, consulte los siguientes recursos:

- Uso de Información de contenedores
- AWS Distro para OpenTelemetry

Solicitudes API

CloudWatch incluye los siguientes tipos de solicitudes de API:

- Solicitudes de API
- En bloque (Obtener)
- Contributor Insights
- Instantánea de imagen de mapa de bits

Las solicitudes de API generan costos según el tipo de solicitud y el número de métricas solicitadas.

En la siguiente tabla se enumeran los tipos de solicitudes de API e incluye las cadenas de `UsageType` y `Operation`, que puede ayudarlo a analizar e identificar los costos relacionados con la API.

Tipo de solicitud de API	**UsageType**	**Operation**	Finalidad
Solicitudes API	Requests	GetMetric Statistics	Obtiene estadísticas para las métricas especificadas
	Requests	ListMetrics	Enumera las métricas especificadas
	Requests	PutMetricData	Publica puntos de datos de métricas en CloudWatch
	Requests	GetDashboard	Muestra los detalles de los paneles especificados
	Requests	ListDashboards	Muestra los paneles de una cuenta
	Requests	PutDashboard	Crea o actualiza un panel
	Requests	DeleteDas hboards	Elimina todos los paneles especificados
En bloque (Obtener)	GMD-Metrics	GetMetricData	Recupera los valores métricos de CloudWatch
Contributor Insights	GIRR-Metrics	GetInsigh tRuleReport	Devuelve datos de serie temporal recopilados por una regla de Contributor Insights

Tipo de solicitud de API	**UsageType**	**Operation**	Finalidad
Instantánea de imagen de mapa de bits	GMWI-Metrics	GetMetric WidgetImage	Recupera una instantánea de una o varias métricas de CloudWatch como imagen de mapa de bits

Para analizar los costos, utilice el Cost Explorer y agrupe los resultados por Operación de la API.

Los costos de las solicitudes de API varían, y usted incurre en costos cuando excede el número de llamadas a la API que se le proporcionan en virtud del límite de capa gratuita de AWS.

> **Note**
>
> GetMetricData y GetMetricWidgetImage no se incluyen en el límite de capa gratuita de AWS. Para obtener más información, consulte Uso del nivel gratuito de AWS en la Guía del usuario de AWS Billing.

Las solicitudes de API que normalmente impulsan los costos sonsolicitudes Put y Get.

PutMetricData

PutMetricData genera costos cada vez que se llama y puede incurrir en costos significativos según el caso de uso. Para obtener más información, consulte PutMetricData en la Referencia de la API de Amazon CloudWatch.

Para aprovechar al máximo los costos generados por PutMetricData, agrupe más datos en sus llamadas a la API. Según el caso de uso, considere la posibilidad de utilizar los Registros de CloudWatch o el formato de métrica incrustado de CloudWatch para inyectar datos de métricas. Para obtener más información, consulte los siguientes recursos:

- ¿Qué son los Registros de Amazon CloudWatch? en la Guía del usuario de los Registros de Amazon CloudWatch

- Incorporación de registros de alta cardinalidad y generación de métricas con formato de métrica integrada de CloudWatch
- Reducción de costos y concentración en nuestros clientes con métricas personalizadas incorporadas de Amazon CloudWatch

GetMetricData

`GetMetricData` también puede generar costos significativos. Los casos de uso comunes que impulsan los costos incluyen herramientas de supervisión de terceros que extraen datos para generar información. Para obtener más información, consulte GetMetricData en la Referencia de la API de los Registros de Amazon CloudWatch.

Para reducir los costos generados por `GetMetricData`, considere extraer solo los datos que se supervisan y usan, o considere extraer datos con menos frecuencia. Dependiendo de su caso de uso, es posible que considere utilizar flujos métricos en lugar de `GetMetricData`, para que pueda enviar datos casi en tiempo real a terceros a un costo menor. Para obtener más información, consulte los siguientes recursos:

- Uso de flujos métricos
- Flujos métricos de CloudWatch: enviar métricas de AWS para socios y aplicaciones en tiempo real

GetMetricStatistics

Dependiendo de su caso de uso, es posible que considere utilizar `GetMetricStatistics` en lugar de `GetMetricData`. Con `GetMetricData`, puede recuperar datos rápidamente y a escala. Sin embargo, `GetMetricStatistics` se incluye en el límite de capa gratuita de AWS para hasta un millón de solicitudes de API, que puede ayudar a reducir los costos si no necesita recuperar tantas métricas y puntos de datos por llamada. Para obtener más información, consulte los siguientes recursos:

- GetMetricStatistics en la Referencia de la API de Amazon CloudWatch
- ¿Debo usar GetMetricData o GetMetricStatistics?

> **Note**
>
> Los generadores de llamadas externos realizan llamadas a la API. Actualmente, la única forma de identificar a estas personas que llaman es abrir una solicitud de soporte técnico al

equipo de CloudWatch y solicitar información sobre ellas. Para obtener información acerca de la creación de una solicitud de asistencia técnica, consulte ¿Cómo puedo conseguir ayuda técnica de AWS?.

Flujos métricos de CloudWatch

Con los flujos métricos de CloudWatch, puede enviar métricas de forma continua a destinos de AWS y destinos de proveedores de servicios de terceros.

Los flujos métricos generan costos en función de la cantidad de actualizaciones métricas. Las actualizaciones de métrica siempre incluyen valores para las siguientes estadísticas:

- `Minimum`
- `Maximum`
- `Sample Count`
- `Sum`

Para obtener más información, consulte Estadísticas que se pueden transmitir en streaming.

Para analizar los costos que generan las flujos métricos de CloudWatch, utilice AWS Cost and Usage Report con Athena. De esta manera, puede identificar qué flujos métricos generan costos y determinar cómo se generan los costos.

Ejemplo: consulta de Athena

Puede utilizar la siguiente consulta para realizar un seguimiento de qué flujos métricos generan costos en función del nombre de recurso de Amazon (ARN).

```
SELECT
SPLIT_PART(line_item_resource_id,'/',2) AS "Stream Name",
line_item_resource_id as ARN,
SUM(CAST(line_item_unblended_cost AS decimal(16,2))) AS TotalSpend
FROM
costandusagereport
WHERE
product_product_name = 'AmazonCloudWatch'
AND year='2022'
AND month='4'
```

```
AND line_item_line_item_type NOT IN
 ('Tax','Credit','Refund','EdpDiscount','Fee','RIFee')
-- AND line_item_usage_account_id = '123456789012' - If you want to filter on a
 specific account, you can
remove this comment at the beginning of the line and specify an AWS account.
AND line_item_usage_type LIKE '%%MetricStreamUsage%%'
GROUP BY line_item_resource_id
ORDER BY TotalSpend DESC
```

Para reducir los costos generados por los flujos métricos de CloudWatch, transmita solo las métricas que aporten valor a su negocio. También puede detener o pausar cualquier transmisión de métricas que no esté utilizando.

Alarmas de CloudWatch

Con las alarmas de CloudWatch, puede crear alarmas basadas en una sola métrica, alarmas basadas en una consulta de Información de métricas y alarmas compuestas que observan otras alarmas.

> **Note**
>
> Los costos de las alarmas métricas y compuestas se prorratean por horas. Usted incurre en costos por las alarmas solo mientras estas existan. Para optimizar los costos, asegúrese de no dejar alarmas mal configuradas o de bajo valor. Para ello, puede automatizar la limpieza de las alarmas de CloudWatch que ya no necesite. Para obtener más información, consulte Automating Amazon CloudWatch Alarm Cleanup at Scale.

Alarmas de métricas

Las alarmas de métricas tienen los siguientes ajustes de resolución:

- Estándar (se evalúan cada 60 segundos)
- Alta resolución (se evalúan cada 10 segundos)

Al crear una alarma de métrica, los costos se basan en la configuración de resolución de la alarma y en el número de métricas a las que esta hace referencia. Por ejemplo, una alarma de métrica que hace referencia a una métrica incurre en un costo métrico de alarma por hora. Para obtener más información, consulte Uso de las alarmas de Amazon CloudWatch.

Si crea una alarma de métrica que contiene una expresión matemática de métrica, que hace referencia a varias métricas, incurrirá en un costo por cada métrica de alarma a la que se haga referencia en la expresión matemática de métrica. Para obtener más información acerca de cómo crear una alarma de métrica que contenga una expresión matemática de métrica de, consulte Creación de una alarma de CloudWatch basada en una expresión matemática de métrica.

Si crea una alarma de detección de anomalías, en la que la esta analiza datos de métricas anteriores para crear un modelo de valores esperados, incurrirá en un costo por cada métrica de alarma a la que se hace referencia en la alarma más dos métricas adicionales, una para las métricas de banda superior e inferior que crea el modelo de detección de anomalías. Para obtener más información acerca de cómo crear una alarma de detección de anomalías, consulte Creación de una alarma de CloudWatch basada en la detección de anomalías.

Alarma de consulta de Información de métricas

Las alarmas de consulta de Información de métricas son un tipo específico de alarma de métrica, solo disponible con la resolución estándar (evaluada cada 60 segundos).

Cuando crea una alarma de consulta de Información de métricas, sus costos se basan en la cantidad de métricas que analice la consulta a la que hace referencia su alarma. Por ejemplo, una alarma de consulta de Información de métrica que haga referencia a una consulta cuyo filtro coincida con diez métricas, tiene un costo de diez métricas analizadas por hora. Para más información, consulte el ejemplo acerca de los precios en Amazon CloudWatch Pricing (Precios de Amazon CloudWatch).

Si crea una alarma que contenga una consulta de Información de métricas y una expresión matemática métrica, se registra como una alarma de consulta de Información de métricas. Si su alarma contiene una expresión matemática de métrica que haga referencia a otras métricas además de las métricas analizadas por la consulta de Información de métricas, tendrá un costo adicional por cada métrica de alarma a la que se haga referencia en la expresión matemática de métrica. Para obtener más información acerca de cómo crear una alarma de métrica que contenga una expresión matemática de métrica de, consulte Creación de una alarma de CloudWatch basada en una expresión matemática de métrica.

Alarmas compuestas

Las alarmas compuestas contienen expresiones de reglas que especifican cómo deben evaluar los estados de otras alarmas para determinar sus propios estados. Las alarmas compuestas incurren en un costo estándar por hora, independientemente de cuántas otras alarmas evalúen. Las alarmas a las que hacen referencia las alarmas compuestas en expresiones de reglas incurren en costos separados. Para obtener más información, consulte Creación de una alarma compuesta.

Tipos de uso de alarmas

En la siguiente tabla, se enumeran los nombres de las subfunciones relevantes para alarmas de CloudWatch. La tabla incluye las cadenas de `UsageType`, que puede ayudarlo a analizar e identificar los costos relacionados con las alarmas.

Función secundaria de CloudWatch	**`UsageType`**
Alarma métrica estándar	`AlarmMonitorUsage`
Alarma métrica de alta resolución	`HighResAlarmMonitorUsage`
Alarma de consulta de Información de métricas	`MetricInsightAlarmUsage`
Alarma compuesta	`CompositeAlarmMonitorUsage`

Reducción de costos de alarmas

Para optimizar los costos generados por las alarmas matemáticas de métricas que agregan cuatro o más métricas, puede agregar los datos antes de enviarlos a CloudWatch. De esta forma, puede crear una alarma para una sola métrica en lugar de una alarma que agrupe los datos de varias métricas. Para obtener más información, consulte Publicación de métricas personalizadas.

Para optimizar los costos generados por las alarmas de consulta de Información de métricas, puede asegurarse de que el filtro que use en la consulta coincida solo con las métricas que quiera supervisar.

La mejor manera de reducir los costos es eliminar todas las alarmas innecesarias o no utilizadas. Por ejemplo, puede eliminar alarmas que evalúan métricas emitidas por recursos de AWS que ya no existen.

Ejemplo: compruebe si hay alarmas en estado ***`INSUFFICIENT_DATA`*** con ***`DescribeAlarms`***

Si elimina un recurso, pero no las alarmas métricas que emite el recurso, estas seguirán existiendo y, por lo general, pasarán al state `INSUFFICIENT_DATA`. Para comprobar si las alarmas están en estado `INSUFFICIENT_DATA`, utilice el siguiente comando AWS Command Line Interface (AWS CLI).

```
$ aws cloudwatch describe-alarms –state-value INSUFFICIENT_DATA
```

Otras formas de reducir costos incluyen:

- Asegúrese de crear alarmas para las métricas correctas.
- Asegúrese de no tener ninguna alarma habilitada en las regiones en las que no esté trabajando.
- Recuerde que, aunque las alarmas compuestas reducen el ruido, también generan costos adicionales.
- Al decidir si crear una alarma estándar o una alarma de alta resolución, tenga en cuenta su caso de uso y el valor que aporta cada tipo de alarma.

Registros de CloudWatch

Los registros de Amazon CloudWatch tiene los siguientes tipos de registro:

- Registros de personalizados (registros que crea para sus aplicaciones)
- Registros proporcionados (registros que otros Servicios de AWS, como Amazon Virtual Private Cloud (Amazon VPC) y Amazon Route 53, crean en su nombre)

Para obtener más información acerca de los registros proporcionados, consulte Habilitación registros desde determinados servicios de AWS en la Guía del usuario de Registros de Amazon CloudWatch.

Los registros personalizados y proporcionados generan costos en función del número de registros que se recopiló, almacenó y analizó. Por separado, los registros proporcionados generan costos de entrega a Amazon S3 y Firehose.

En la siguiente tabla se enumeran los nombres de las características de Registro de CloudWatch y los nombres de las subfunciones relevantes. La tabla incluye las cadenas de `UsageType` y `Operation`, que puede ayudarlo a analizar e identificar los costos relacionados con los registros.

Característica de Registros de CloudWatch	Característica secundaria de los Registros de CloudWatch	**`UsageType`**	**`Operation`**	Finalidad
Registros personalizados	Recopilar (ingerir)	`DataProcessing-Bytes`	`PutLogEvents`	Carga un lote de registros en un flujo de registro específico

Característica de Registros de CloudWatch	Característica secundaria de los Registros de CloudWatch	**`UsageType`**	**`Operation`**	Finalidad
	Almacenar (archivar)	`TimedStorage-ByteHrs`	`HourlyStorageMetering`	Almacena registros por hora y registros por byte en Registros de CloudWatch
	Analizar (consultas de Logs Insights)	`DataScanned-Bytes`	`StartQuery`	Registra datos analizados por consultas de CloudWatch Logs Insights
Registros proporcionados	Entrega (Registros de CloudWatch)	`VendedLog-Bytes`	`PutLogEvents`	Carga un lote de registros en un flujo de registro específico
	Entrega (Amazon S3)	`S3-Egress-ComprBytes` `S3-Egress-Bytes`	`LogDelivery`	Envía registros proporcionados (CloudWatch, Amazon S3 o Firehose)
	Entrega (Firehose)	`FH-Egress-Bytes`	`LogDelivery`	Envía registros proporcionados (CloudWatch, Amazon S3 o Firehose)

Para analizar los costos, utilice AWS Cost and Usage Report con Athena, para que pueda identificar qué registros están generando costos y determinar cómo se generan los costos.

Ejemplo: consulta de Athena

Puede usar la siguiente consulta para realizar un seguimiento de los registros que generan costos por ID de recurso.

```
SELECT
bill_payer_account_id as Payer,
line_item_usage_account_id as LinkedAccount,
line_item_resource_id AS ResourceID,
line_item_usage_type AS UsageType,
SUM(CAST(line_item_unblended_cost AS decimal(16,8))) AS TotalSpend,
SUM(CAST(line_item_usage_amount AS double)) AS UsageQuantity
FROM
costandusagereport
WHERE
product_product_name = 'AmazonCloudWatch'
AND year='2022'
AND month='4'
AND line_item_operation IN
 ('PutLogEvents','HourlyStorageMetering','StartQuery','LogDelivery')
AND line_item_line_item_type NOT IN
 ('Tax','Credit','Refund','EdpDiscount','Fee','RIFee')
GROUP BY
bill_payer_account_id,
line_item_usage_account_id,
line_item_usage_type,
line_item_resource_id,
line_item_operation
ORDER BY
TotalSpend DESC
```

Para aprovechar al máximo los costos que generan los Registros de CloudWatch, tenga en cuenta lo siguiente:

- Registre solo los eventos que aporten valor a su negocio. Esto le ayuda a generar menos costos de incorporación.
- Cambie la configuración de retención de registros para generar menos costos de almacenamiento. Para obtener más información, consulte Cambiar la retención de datos de registro en Registros de CloudWatch en la Guía de usuario de Registros de Amazon CloudWatch.

- Ejecute consultas que CloudWatch Logs Insights guarde automáticamente en su historial. De esta forma, se generan menos costos de análisis. Para obtener más información, consulte Ver consultas en marcha o historial de consultas en la Guía del usuario de Registros de Amazon CloudWatch.
- Utilice el agente de CloudWatch para recopilar registros del sistema y de la aplicación y enviarlos a CloudWatch. De esta forma, solo puede recopilar los eventos de registro que cumplan con sus criterios. Para obtener más información, consulte El agente de Amazon CloudWatch agrega ayuda para expresiones de filtro de registro.

Para reducir los costos de los registros proporcionados, tenga en cuenta su caso de uso y, luego, determine si los registros deben enviarse a CloudWatch o Amazon S3. Para obtener más información, consulte Registros enviados a Amazon S3 en la Guía del usuario de Registros de Amazon CloudWatch.

> **Tip**
>
> Si quiere usar filtros de métricas, filtros de suscripción, Información de registros de CloudWatch e Información de colaboradores, envíe los registros proporcionados a CloudWatch.
> Como alternativa, si trabaja con registros de flujo de VPC y los usa con fines de auditoría y cumplimiento, envíe los registros entregados a Amazon S3.
> Para obtener información sobre cómo realizar un seguimiento de los cargos que se generan al publicar registros de flujo de VPC en buckets de S3, consulte Uso de AWS Cost and Usage Report y etiquetas de asignación de costos para comprender la ingesta de datos de los registros de flujo de VPC en Amazon S3.

Para obtener información adicional sobre cómo aprovechar al máximo los costos que generan los CloudWatch Logs, consulte ¿Qué grupo de registros está provocando un aumento repentino en mi factura de Registros de CloudWatch?.

Uso de paneles de Amazon CloudWatch

Los paneles de Amazon CloudWatch son páginas de inicio personalizables en la consola de CloudWatch que puede utilizar para monitorear los recursos en una única vista, incluso aquellos que se esparcen entre diferentes regiones. Puede utilizar los paneles de CloudWatch para crear vistas personalizadas de las métricas y las alarmas para los recursos de AWS.

Con los paneles, puede crear lo siguiente:

- Una única vista para las métricas y las alarmas que seleccione a fin de ayudarle a evaluar el estado de los recursos y las aplicaciones de una o más Regiones. Puede seleccionar el color utilizado para cada métrica en cada gráfico, de modo que pueda realizar un seguimiento de la misma métrica fácilmente en varios gráficos.
- Un manual de estrategia operativo que ofrece asesoramiento a los miembros del equipo durante eventos operativos sobre cómo responder a determinados incidentes.
- Una vista común de las medidas de los recursos y las aplicaciones críticos que pueden compartir los miembros del equipo para un flujo de comunicación más rápido durante los eventos operativos.

Si tiene varias cuentas de AWS, puede configurar la observabilidad entre cuentas de CloudWatch y, a continuación, crear detallados paneles entre cuentas en las cuentas de monitoreo. Estos paneles pueden incluir gráficos de las métricas de las cuentas de origen y widgets de Información de registros de CloudWatch que tengan consultas a los grupos de registro de las cuentas de origen. Además, las alarmas que cree en la cuenta de monitoreo pueden ver las métricas en las cuentas de origen. Para obtener más información, consulte Observabilidad entre cuentas de CloudWatch.

Puede crear paneles desde la consola o mediante la AWS CLI o la operación de la API `PutDashboard`. Puede agregar paneles a una lista de favoritos, desde la que puede acceder no solo a los paneles favoritos, sino también a los paneles que se visitaron recientemente. Para obtener más información, consulte Agregar un panel de la lista de favoritos.

Para acceder a los paneles de CloudWatch, necesita uno de los siguientes:

- La política `AdministratorAccess`
- La política `CloudWatchFullAccess`
- Una política personalizada que incluya uno o varios de estos permisos específicos:
  - `cloudwatch:GetDashboard` y `cloudwatch:ListDashboards` para poder ver los paneles

- `cloudwatch:PutDashboard` para poder crear o modificar paneles
- `cloudwatch:DeleteDashboards` para poder eliminar paneles

Contenido

Creación de un panel de CloudWatch

Para comenzar, cree un panel de CloudWatch. Puede crear varios paneles y agregar tableros a una lista de favoritos. No se verá limitado al número de paneles que puede tener en la Cuenta de AWS. Todos los paneles son globales. No son específicos de cada región.

En el siguiente procedimiento, se muestra cómo crear un panel desde la consola de CloudWatch. Puede utilizar la operación de la API `PutDashboard` para crear un panel desde la interfaz de la línea de comandos. La operación de la API contiene una cadena JSON que define el contenido del panel. Para obtener más información acerca de cómo se crea un panel con la operación de la API `PutDashboard`, consulte PutDashboard en la Referencia de la API de Amazon CloudWatch.

> **Tip**
>
> Si va a crear un nuevo panel con la operación de la API `PutDashboard`, puede utilizar la cadena JSON desde un panel que ya existe.

Para crear un panel desde la consola

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Dashboards (Paneles) y después Create dashboard (Crear panel).
3. En el cuadro de diálogo Create new dashboard (Crear nuevo panel), escriba un nombre para el panel y luego elija Create dashboard (Crear panel).

   Si utiliza el nombre CloudWatch-Default o CloudWatch-Default-***ResourceGroupName***, el panel aparece en la información general de la página de inicio de CloudWatch en Default Dashboard (Panel predeterminado). Para obtener más información, consulte Introducción a Amazon CloudWatch.
4. En el cuadro de diálogo Add to this dashboard (Agregar a este panel), realice una de las acciones siguientes:
   - Para agregar un gráfico al panel, elija Line (Línea) o Stacked area (Área apilada) y, a continuación, elija Configure (Configurar). En el cuadro de diálogo Add metric graph (Añadir gráfico de métrica), seleccione las métricas que desea representar gráficamente y, a continuación, elija Create widget (Crear un widget). Si una métrica no aparece en el cuadro de diálogo porque no ha publicado datos durante más de 14 días, puede agregarla manualmente. Para obtener más información, consulte Grafique las métricas manualmente en un panel de CloudWatch.
   - Para agregar al panel un número que muestre la métrica, elija Number (Número) y después elija Configure (Configurar). En el cuadro de diálogo Add metric graph (Añadir gráfico de métrica), seleccione las métricas que desea representar gráficamente y, a continuación, elija Create widget (Crear un widget).
   - Para agregar un bloque de texto al panel, elija Text (Texto) y después elija Configure (Configurar). En el cuadro de diálogo New text widget (Nuevo widget de texto), para Markdown (Marcado), dele formato utilizando Markdown (Marcado) y, a continuación, elija Create widget (Crear un widget).
5. (Opcional) Elija Add widget (Agregar un widget) y luego repita el paso 4 para agregar otro widget al panel. Puede repetir este paso varias veces.

   Para cada gráfico del panel, hay un ícono de información en la parte superior derecha. Seleccione este ícono para ver las descripciones de las métricas del gráfico.
6. Elija Save dashboard (Guardar panel).

Panel para la observabilidad entre cuentas de CloudWatch

Si tiene varias cuentas de AWS, puede configurar la observabilidad entre cuentas de CloudWatch y, a continuación, crear detallados paneles entre cuentas en las cuentas de monitoreo. Puede buscar, visualizar y analizar sin problemas sus métricas, registros y seguimientos sin límites en la cuenta.

Para más información sobre cómo configurar la observabilidad entre cuentas de CloudWatch, consulte Observabilidad entre cuentas de CloudWatch.

Con la observabilidad entre cuentas de CloudWatch, puede hacer lo siguiente en el panel de una cuenta de monitoreo:

- Puede buscar, ver y crear gráficos de métricas que estén guardados en las cuentas de origen. Un solo gráfico puede incluir métricas de varias cuentas.
- Puede crear alarmas en la cuenta de monitoreo que puedan controlar las métricas de las cuentas de origen.
- Puede ver los eventos de registro de los grupos de registro ubicados en las cuentas de origen y hacer consultas de Información de registros de CloudWatch de los grupos de registro que estén en las cuentas de origen. Una sola consulta de Información de registros de CloudWatch en una cuenta de monitoreo puede consultar varios grupos de registro en varias cuentas de origen a la vez.
- Ver los nodos de las cuentas de origen en un mapa de seguimiento de X-Ray. A continuación, puede filtrar el mapa para ver cuentas de origen específicas.

Al iniciar sesión en una cuenta de monitoreo, aparece una insignia azul de la Monitoring account (Cuenta de monitoreo) en la parte superior derecha de cada página que admita la función de observabilidad entre cuentas de CloudWatch.

Paneles para cuentas y Regiones cruzadas

Cree cross-account cross-Region dashboards (paneles para cuentas y Regiones cruzadas), que resumen los datos de CloudWatch de varias cuentas y Regiones de AWS en un solo panel. Desde este panel de alto nivel, puede obtener una vista de toda la aplicación y también profundizar en paneles más específicos sin tener que iniciar y cerrar la sesión de cada cuenta ni cambiar de Región.

Puede crear paneles para diversas cuentas y regiones en la AWS Management Console y mediante programación.

Requisito previo

Para poder crear un panel para diversas cuentas y regiones, debe habilitar al menos una cuenta de uso compartido y una cuenta de monitorización. Además, para poder usar la consola de CloudWatch a fin de crear un panel de cuentas cruzadas, habilite la consola para la funcionalidad de cuentas cruzadas. Para obtener más información, consulte Consola de CloudWatch para cuentas y Regiones cruzadas.

Creación y uso de un panel para diversas cuentas y regiones con la AWS Management Console

Puede usar la AWS Management Console con el fin de crear un panel para diversas cuentas y regiones.

Para crear un panel para cuentas y Regiones cruzadas

1. Inicie sesión en la cuenta de monitoreo.
2. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
3. En el panel de navegación, elija Dashboards (Paneles).
4. Elija un panel o cree uno nuevo.
5. En la parte superior de la pantalla, puede cambiar entre cuentas y Regiones. Al crear el panel, puede incluir widgets de diversas cuentas y regiones. Los widgets incluyen gráficos, alarmas y widgets de CloudWatch Logs Insights.

Creación de un gráfico con métricas de diferentes cuentas y regiones

1. Inicie sesión en la cuenta de monitoreo.
2. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
3. En el panel de navegación, elija Metrics (Métricas) y, a continuación, All metrics (Todas las métricas).
4. Seleccione la cuenta y la región desde la que desea agregar métricas. Puede seleccionar su cuenta y región en los menús desplegables de cuenta y región que se encuentran cerca de la parte superior derecha de la pantalla.
5. Agregue las métricas que desee al gráfico. Para obtener más información, consulte Representación gráfica de las métricas.
6. Repita los pasos 4-5 para agregar métricas de otras cuentas y regiones.

7. (Opcional) Seleccione la pestaña Graphed metrics (Métricas gráficas) y agregue una función matemática de métrica que utilice las métricas que haya elegido. Para obtener más información, consulte Uso de la calculadora de métricas.

   También puede configurar un gráfico único para incluir varias funciones `SEARCH`. Cada búsqueda puede hacer referencia a una cuenta o región diferente.

8. Cuando haya terminado con el gráfico, elija Actions (Acciones) y Add to dashboard (Agregar al panel).

   Seleccione el panel para diversas cuentas y elija Add to dashboard (Agregar al panel).

Adición de una alarma de una cuenta diferente al panel para diversas cuentas

1. Inicie sesión en la cuenta de monitoreo.
2. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
3. En la parte superior de la página, elija la cuenta en la que se encuentra la alarma.
4. En el panel de navegación, elija Alarms.
5. Seleccione la casilla situada junto a la alarma que desea agregar y elija Add to dashboard (Agregar al panel).
6. Seleccione el panel para diversas cuentas al que desea agregarla y elija Add to dashboard (Agregar al panel).

Creación de un panel para cuentas y Regiones cruzadas mediante programación

Puede utilizar las API y los SDK de AWS para crear paneles mediante programación. Para obtener más información, consulte PutDashboard.

Con el fin de habilitar los paneles para diversas cuentas y regiones, hemos añadido nuevos parámetros a la estructura del cuerpo del panel, como se muestra en la tabla y en los ejemplos siguientes. Para obtener más información acerca de la estructura general del cuerpo del panel, consulte Sintaxis y estructura del cuerpo de los paneles.

Parámetro	Uso	Ámbito	Predeterminado
`accountId`	Especifica el ID de la cuenta en la que se encuentra el widget o la métrica.	Widget o métrica	Cuenta que ha iniciado sesión actualmente
`region`	Especifica la región de la métrica.	Widget o métrica	Región actual seleccionada en la consola

Los siguientes ejemplos ilustran el código fuente JSON de los widgets en un panel para diversas cuentas y regiones.

En este ejemplo se establece el campo `accountId` en el ID de la cuenta de uso compartido en el nivel de widget. Esto especifica que todas las métricas de este widget provendrán de esa cuenta de uso compartido y región.

```
{
    "widgets": [
        {
            ...
            "properties": {
              "metrics": [
                     …..
                     ],
                    "accountId": "111122223333",
                    "region": "us-east-1"
           }
       }
  ]
}
```

En este ejemplo se establece el campo `accountId` de forma diferente para cada métrica. En este ejemplo, los diferentes valores de la expresión matemática de esta métrica provienen de diferentes cuentas de uso compartido y regiones.

```
{
"widgets": [
    {
```

```
        ...
        "properties": {
            "metrics": [
                [ { "expression": "SUM(METRICS())", "label": "[avg: ${AVG}]
 Expression1", "id": "e1", "stat": "Sum" } ],
                [ "AWS/EC2", "CPUUtilization", { "id": "m2", "accountId":
 "5555666677778888", "region": "us-east-1", "label": "[avg: ${AVG}] ApplicationALabel
 " } ],
                [ ".", ".", { "id": "m1", "accountId": "9999000011112222", "region":
 "eu-west-1", "label": "[avg: ${AVG}] ApplicationBLabel" } ]
            ],
            "view": "timeSeries",
            "region":"us-east-1", ---> home region of the metric. Not present in above
 example
            "stacked": false,
            "stat": "Sum",
            "period": 300,
            "title": "Cross account example"
        }
    }
]
}
```

Este ejemplo muestra un widget de alarma.

```
{
    "type": "metric",
    "x": 6,
    "y": 0,
    "width": 6,
    "height": 6,
    "properties": {
        "accountID": "111122223333",
        "title": "over50",
        "annotations": {
            "alarms": [
                "arn:aws:cloudwatch:us-east-1:379642911888:alarm:over50"
            ]
        },
        "view": "timeSeries",
        "stacked": false
    }
}
```

Este ejemplo es para un widget de CloudWatch Logs Insights.

```
{
    "type": "log",
    "x": 0,
    "y": 6,
    "width": 24,
    "height": 6,
    "properties": {
        "query": "SOURCE 'route53test' | fields @timestamp, @message\n| sort @timestamp
 desc\n| limit 20",
        "accountId": "111122223333",
        "region": "us-east-1",
        "stacked": false,
        "view": "table"
    }
}
```

Otra forma de crear paneles mediante programación consiste en crear primero uno en la AWS Management Console y, luego, copiar el código fuente JSON de este panel. Para ello, cargue el panel y elija Actions (Acciones) y View/edit source (Ver/editar código fuente). A continuación, puede copiar el código JSON del panel y utilizarlo como plantilla para crear paneles similares.

Cree paneles flexibles con variables de panel

Utilice las variables de panel para crear paneles flexibles que puedan mostrar rápidamente diferentes contenidos en varios widgets, según el valor de un campo de entrada del panel. Por ejemplo, puede crear un panel que pueda cambiar rápidamente entre distintas funciones de Lambda o ID de instancia de Amazon EC2, o uno que pueda cambiar a distintas regiones de AWS.

Después de crear un panel que utilice una variable, puede copiar el mismo patrón de variables en otros paneles existentes.

El uso de variables de panel mejora el flujo de trabajo operativo de las personas que los utilizan. También puede reducir sus costes, porque usará las variables de panel en uno solo, en lugar de crear varios paneles similares.

> **Note**
>
> Si comparte un panel que contiene variables de panel, las personas con las que comparte el panel no podrán cambiar los valores de las variables.

Tipos de variables de panel

La variable de panel puede ser una variable de propiedad o una variable de patrón.

- Las variables de propiedad cambian todas las instancias de una propiedad en todos los widgets del panel. Esta propiedad puede ser cualquier propiedad de JSON de la fuente JSON de un panel, por ejemplo `region`. También puede ser el nombre de una dimensión para una métrica, como `InstanceID` o`FunctionName`.

  Para ver un tutorial que utiliza una variable de propiedad, consulte Tutorial: Crear un panel de Lambda con el nombre de la función como variable.

  Para obtener más información sobre el código fuente JSON de los paneles, consulte Estructura y sintaxis del cuerpo del panel. En la consola de CloudWatch, puede ver el código fuente JSON de cualquier panel personalizado seleccionando Acciones y Ver/editar fuente.
- Las variables de patrón utilizan un patrón de expresión regular para cambiar toda una propiedad de JSON o solo una parte determinada de ella.

  Para ver un tutorial que utiliza una variable de patrón, consulte Tutorial: Cree un panel que utilice un patrón de expresiones regulares para cambiar de una región a otra.

Las variables de propiedad se aplican a la mayoría de los casos de uso y su configuración es menos compleja.

Temas

- Tutorial: Crear un panel de Lambda con el nombre de la función como variable
- Tutorial: Cree un panel que utilice un patrón de expresiones regulares para cambiar de una región a otra
- Copiar una variable a otro panel

Tutorial: Crear un panel de Lambda con el nombre de la función como variable

Los pasos de este procedimiento ilustran cómo crear un panel flexible que muestre una variedad de gráficos métricos mediante una variable de propiedad. Esto incluye un cuadro de selección desplegable en el panel de control que puede utilizar para cambiar las métricas de todos los gráficos entre distintas funciones de Lambda.

Otros ejemplos de casos de uso para este tipo de panel incluyen usar `InstanceId` como variable para crear un panel de métricas con un menú desplegable con los ID de las instancias. Como alternativa, puede crear un panel que utilice `region` como variable el mismo conjunto de métricas de distintas regiones.

Usar una variable de propiedad de panel para crear un panel de Lambda flexible

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, seleccione Dashboards, Create dashboard.
3. Introduzca un nombre para el panel y seleccione Crear panel.
4. Añadir widgets al panel de control que muestren las métricas de una función de Lambda. Al crear estos widgets, especifique Lambda y Por nombre de función para las métricas del widget. Para la función, especifique una de las funciones de Lambda que desee incluir en este panel.

   Para obtener más información sobre cómo añadir widgets a un panel, consulte Creación y operación de widgets en paneles de CloudWatch.
5. Después de añadir los widgets, mientras ve el panel, elija Acciones, Variables y Crear una variable.
6. Seleccione Variable de propiedad.
7. En Propiedad que cambia la variable, elija FunctionName.
8. Para el tipo de entrada, en este caso de uso, recomendamos elegir Seleccionar menú (desplegable). Esto crea un menú desplegable en el panel de control en el que puede seleccionar el nombre de la función de Lambda para mostrar las métricas.

   Si se tratara de un panel en el que se alternara entre solo dos o tres valores diferentes para una variable, el botón de opción sería una buena elección.

   Si prefiere introducir o pegar los valores de la variable, puede elegir Entrada de texto. Esta opción no incluye una lista desplegable ni botones de opción.

9. Cuando elija Seleccionar menú (desplegable), deberá elegir si desea rellenar el menú introduciendo valores o utilizando una búsqueda métrica. Para este caso, vamos a suponer que tiene un gran número de funciones de Lambda y no desea introducirlas todas manualmente. Elija Usar los resultados de una búsqueda métrica y, a continuación, haga lo siguiente:

    a. Elija Consultas prediseñadas, Lambda, Errores.

       (Al elegir Errores no se añade la métrica de Errores al panel de control; pero rellena rápidamente el cuadro de selección de variables FunctionName.)

    b. Elija Por nombre de función y, a continuación, elija Buscar.

       En el botón Búsqueda, verá la opción FunctionName seleccionada. También verá un mensaje sobre cuántos valores de dimensión FunctionName se han encontrado para rellenar el cuadro de entrada.

10. (Opcional) Para obtener más ajustes, seleccione Ajustes secundarios y realice una o varias de las siguientes acciones:

    - Para personalizar el nombre de la variable, introdúzcalo en Nombre de variable personalizada.
    - Para personalizar la etiqueta del campo de entrada de la variable, introdúzcala en Etiqueta de entrada.
    - Para establecer el valor predeterminado de esta variable cuando se abra el panel por primera vez, introduzca el valor predeterminado en Valor predeterminado.

11. Seleccione Añadir variable.

    Aparece un cuadro de selección desplegable FunctionName cerca de la parte superior del panel de control. Puede seleccionar una función de Lambda en este cuadro y todos los widgets que utilicen la variable mostrarán información sobre la función seleccionada.

    Más adelante, si agrega más widgets al panel que observen las métricas de Lambda con la dimensión FunctionName, utilizarán la variable automáticamente.

Tutorial: Cree un panel que utilice un patrón de expresiones regulares para cambiar de una región a otra

Los pasos de este procedimiento ilustran cómo crear un panel flexible que pueda cambiar de una región a otra. Este tutorial utiliza una variable de patrón de expresión regular en lugar de una variable

de propiedad. Para ver un tutorial que utiliza una variable de propiedad, consulte Tutorial: Crear un panel de Lambda con el nombre de la función como variable.

En muchos casos de uso, puede crear un panel que cambie de una región a otra utilizando una variable de propiedad. Sin embargo, si los widgets se basan en los nombres de recursos de Amazon (ARN) que incluyen nombres de regiones, debe usar una variable de patrón para cambiar los nombres de las regiones dentro de los ARN.

Usar una variable de patrón de panel para crear un panel flexible multirregional

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, seleccione Dashboards, Create dashboard.
3. Introduzca un nombre para el panel y seleccione Crear panel.
4. Agregar widgets al panel. Cuando añada los widgets en los que desee mostrar datos específicos de una región, evite especificar dimensiones con valores que aparezcan solo en una región. Por ejemplo, para las métricas de Amazon EC2, especifique las métricas que se agreguen en lugar de las que usen InstanceID como dimensión.

   Para obtener más información sobre cómo añadir widgets a un panel, consulte Creación y operación de widgets en paneles de CloudWatch.
5. Después de añadir los widgets, mientras ve el panel, elija Acciones, Variables y Crear una variable.
6. Seleccione Variable de patrón.
7. En Propiedad que cambia la variable, introduzca el nombre de la región del panel de control actual, por ejemplo**`us-east-2`**.

   Habrá introducido la región correcta si la etiqueta situada debajo de ese cuadro muestra los widgets que se verán afectados por la variable.
8. En Tipo de entrada, para este caso de uso, seleccione Botón de opción.
9. En Definir cómo se rellenan las entradas, elija Crear una lista de valores personalizados.
10. En Cree sus valores personalizados, introduzca las regiones entre las que desee cambiar, con una región en cada línea. Después de cada región, introduzca una coma y, a continuación, la etiqueta que desee mostrar en ese botón de opción. Por ejemplo:

    `us-east-1, N. Virginia`

    `us-east-2, Ohio`

eu-west-3, Paris

A medida que rellena los valores personalizados, el panel de vista previa se actualiza para mostrar el aspecto que tendrán los botones de opción.

11. (Opcional) Para obtener más ajustes, seleccione Ajustes secundarios y realice una o varias de las siguientes acciones:

    - Para personalizar el nombre de la variable, introdúzcalo en Nombre de variable personalizada.
    - Para personalizar la etiqueta del campo de entrada de la variable, introdúzcala en Etiqueta de entrada. En este tutorial, escriba **Region:**.

      Si introduce un valor aquí, el panel de vista previa se actualiza para mostrar el aspecto que tendrán los botones de opción.
    - Para establecer el valor predeterminado de esta variable cuando se abra el panel por primera vez, introduzca el valor predeterminado en Valor predeterminado.

12. Seleccione Añadir variable.

    Aparece el panel con una etiqueta de región: junto a los botones de opción de las regiones, cerca de la parte superior. Al cambiar de una región a otra, todos los widgets que utilizan la variable mostrarán información sobre la región seleccionada.

Copiar una variable a otro panel

Después de crear un panel con variables útiles, puede copiar estas variables en otros paneles de control existentes. Para obtener más información acerca de las variables , consulte Cree paneles flexibles con variables de panel.

Copiar una variable del panel a otro panel

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Paneles y, a continuación, elija el nombre del panel que contiene la variable que desea copiar. Introduzca una cadena para buscar los paneles con nombres coincidentes, si es necesario.
3. Elija Acciones, Variables y Administrar variables.
4. Seleccione el botón de opción situado junto a la variable que desea copiar y elija Copiar a otro panel.

5. Seleccione el cuadro de selección y comience a escribir el nombre del panel en el que desea copiar la variable.
6. Seleccione el nombre del panel y elija Copiar variable.

Creación y operación de widgets en paneles de CloudWatch

Utilice los temas de esta sección para crear y operar con gráficos, alarmas y widgets de texto de los paneles.

Contenido

Agregue o elimine un gráfico desde un panel de CloudWatch

Puede agregar gráficos que contengan una o varias métricas al panel de CloudWatch. Entre los tipos de gráficos que se pueden agregar al panel, se incluyen Line (Línea), Stacked area (Área apilada), Number (Número), Gauge (Calibre), Bar (Barra) y Pie (Circular). Puede eliminar gráficos del panel cuando ya no los necesita. En los procedimientos de esta sección, se describe cómo agregar y quitar gráficos del panel. Para obtener más información acerca de cómo editar un gráfico en el panel, consulte Editar un gráfico en un panel de CloudWatch.

Para añadir un gráfico a un panel

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Dashboards (Paneles) y, a continuación, elija un panel.
3. Elija el símbolo +y, a continuación, seleccione el gráfico que desea agregar al panel y elija Siguiente.

   - Si selecciona Line (Línea), Stacked area (Área apilada), Bar (Barra) o Pie (Circular), elija Metrics (Métricas).
4. En la pestaña Examinar, busque las métricas que desee representar gráficamente y seleccione las que desee.
5. (Opcional) Para cambiar el intervalo de tiempo del gráfico, seleccione uno de los intervalos de tiempo predefinidos en la parte superior de la pantalla. Los intervalos de tiempo abarcan de 1 hora a 1 semana (1 h, 3 h, 12 h, 1 día, 3 días o 1 sem.).

   Para configurar su propio intervalo de tiempo, elija Custom (Personalizado).

   - (Opcional) Para que este widget siga utilizando el intervalo de tiempo que ha seleccionado, incluso si el intervalo de tiempo del resto del panel se modifica posteriormente, seleccione Persistir el intervalo de tiempo.
6. (Opcional) Puede cambiar el tipo de widget del gráfico por medio del menú desplegable que se encuentra junto a los intervalos de tiempo predefinidos.
7. (Opcional) En Graphed metrics (Métricas diagramadas), puede agregar una etiqueta dinámica a la métrica y cambiar la etiqueta de la métrica, el color de la etiqueta, la estadística y el periodo de la métrica. También puede determinar la posición de las etiquetas en el eje Y de izquierda a derecha.

   a. Para añadir una etiqueta dinámica, elija Graphed metrics (Métricas diagramadas) y, a continuación, elija Dynamic labels (Etiquetas dinámicas). Las etiquetas dinámicas muestran una estadística sobre la métrica en la leyenda del gráfico. Las etiquetas dinámicas se actualizan de manera automática cada vez que se actualiza el panel o el gráfico. De forma predeterminada, los valores dinámicos que añade a las etiquetas aparecen al principio de las etiquetas. Para obtener más información, consulte Uso de etiquetas dinámicas.

   b. Para cambiar el color de una métrica, elija el cuadrado de color que se encuentra junto a ella.

c. Para cambiar la estadística, seleccione el menú desplegable de Statistic (Estadística) y, a continuación, elija un nuevo valor. Para obtener más información, consulte Statistics (Estadísticas).

d. Para cambiar el periodo, seleccione el menú desplegable ubicado debajo de la columna Period (Periodo) y, a continuación, elija un nuevo valor.

8. Si va a crear un widget de calibre, debe elegir la pestaña Opciones y especificar los valores mínimo y máximo que se utilizarán en los dos extremos del calibre.

9. (Opcional) Para personalizar el eje Y, elija Options (Opciones). Puede añadir una etiqueta personalizada en el Left Y-axis (Eje Y izquierdo) en el campo Label (Etiqueta). Si el gráfico muestra valores en el lado derecho del eje Y, también puede personalizar esa etiqueta. Además, puede establecer límites mínimos y máximos en los valores del eje Y, de manera que el gráfico muestre solo los intervalos de valores que usted especifique.

10. (Opcional) Para añadir o editar anotaciones horizontales en gráficos de líneas o áreas apiladas, o para añadir umbrales a los widgets de medición, seleccione Opciones:

    a. Para añadir una anotación horizontal o un umbral, elija Añadir anotación horizontal o Añadir umbral.

    b. En Etiqueta, introduzca una etiqueta para la anotación y, a continuación, seleccione el icono de marca de verificación.

    c. En Value (Valor), elija el icono de lápiz y papel ubicado junto al valor actual e ingrese un valor nuevo. Después de ingresar su valor, seleccione el icono de marca de verificación.

    d. En Fill (Completar), seleccione el menú desplegable y especifique cómo utilizará el sombreado. Puede elegir None (Ninguno), Above (Arriba), Between (Entre) o Below (Abajo). Para cambiar el color de relleno, elija el cuadrado de color ubicado junto a la anotación.

    e. En Axis (Eje), especifique si su anotación debe aparecer en el lado izquierdo o derecho del eje Y.

    f. Para ocultar una anotación, quite la marca de la casilla de verificación situada junto a la anotación que desea ocultar.

    g. Para eliminar una anotación, elija la X en Actions (Acciones).

> **Note**
>
> Puede repetir estos pasos para agregar varias anotaciones horizontales al mismo gráfico o calibre.

11. (Opcional) Para agregar o editar anotaciones verticales, elija Options (Opciones):

    a. Para agregar una anotación vertical, elija Add vertical annotation (Añadir anotación vertical).
    b. En Label (Etiqueta), elija el icono de lápiz y papel ubicado junto a la anotación actual e ingrese una anotación nueva. Si desea mostrar solo la fecha y la hora, deje vacío el campo Label (Etiqueta).
    c. En Date (Fecha), elija la fecha y la hora actuales e ingrese la fecha y la hora nuevas.
    d. En Fill (Completar), seleccione el menú desplegable y especifique cómo la anotación utilizará el sombreado. Puede elegir None (Ninguno), Above (Arriba), Between (Entre) o Below (Abajo). Para cambiar el color de relleno, seleccione el cuadrado de color ubicado junto a la anotación.
    e. Para ocultar una anotación, quite la marca de la casilla de verificación situada junto a la anotación que desea ocultar.
    f. Para eliminar una anotación, elija la X en Actions (Acciones).

    > **Note**
    >
    > Puede repetir estos pasos para agregar varias anotaciones verticales al mismo gráfico.

12. Elija Create widget (Crear widget).
13. Elija Save dashboard (Guardar panel).

Para eliminar un gráfico de un panel

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Dashboards (Paneles) y, a continuación, elija un panel.
3. En la esquina superior derecha del gráfico que desea eliminar, elija Widget actions (Acciones del widget) y, luego, Delete (Eliminar).
4. Elija Save dashboard (Guardar panel).

Grafique las métricas manualmente en un panel de CloudWatch

Si una métrica no ha publicado datos en los últimos 14 días, no podrá encontrarla cuando busque métricas para agregar a un gráfico en un panel de CloudWatch. Utilice los pasos siguientes para agregar cualquier métrica manualmente a un gráfico existente.

Para agregar una métrica que no puede encontrar en la búsqueda a un gráfico

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Dashboards (Paneles) y, a continuación, elija un panel.
3. El panel ya debe contener un gráfico al que desee agregar la métrica. Si no es así, cree el gráfico y añádale cualquier métrica. Para obtener más información, consulte Agregue o elimine un gráfico desde un panel de CloudWatch .
4. Elija Actions (Acciones), View/edit source (Ver/editar código fuente).

   Aparece un bloque JSON. El bloque especifica los widgets del panel y su contenido. A continuación se muestra un ejemplo de una parte de este bloque, que define un gráfico.

```
{
            "type": "metric",
            "x": 0,
            "y": 0,
            "width": 6,
            "height": 3,
            "properties": {
                "view": "singleValue",
                "metrics": [
                    [ "AWS/EBS", "VolumeReadOps", "VolumeId",
 "vol-1234567890abcdef0" ]
                ],
                "region": "us-west-1"
            }
        },
```

   En este ejemplo, la siguiente sección define la métrica mostrada en este gráfico.

```
[ "AWS/EBS", "VolumeReadOps", "VolumeId", "vol-1234567890abcdef0" ]
```

5. Agregue una coma después del corchete final si no hay una y, a continuación, agregue una sección similar entre corchetes después de la coma. En esta nueva sección, especifique el espacio de nombres, el nombre de la métrica y las dimensiones necesarias de la métrica que va a agregar al gráfico. A continuación, se muestra un ejemplo.

```
[ "AWS/EBS", "VolumeReadOps", "VolumeId", "vol-1234567890abcdef0" ],
[ "MyNamespace", "MyMetricName", "DimensionName", "DimensionValue" ]
```

Para obtener más información sobre el formato de las métricas en JSON, consulte Propiedades de un objeto de widget de métrica.

6. Seleccione Actualizar.

Uso de gráficos existentes

Siga los procedimientos descritos en estas secciones para editar y modificar los widgets existentes de los gráficos del panel .

Temas

- Edite un gráfico en un panel de CloudWatch
- Mueva o cambie el tamaño de un gráfico en un panel de CloudWatch.
- Cambie el nombre de un gráfico en un panel de CloudWatch

Edite un gráfico en un panel de CloudWatch

Es posible editar los gráficos que se agregan al panel de CloudWatch. Puede cambiar el título, la estadística o el periodo de un gráfico. También se pueden agregar métricas a los gráficos, actualizarlas y eliminarlas de ellos. Si el gráfico contiene más de una métrica, puede ocultar las métricas que no está utilizando para reducir el desorden. En los procedimientos de esta sección, se describe cómo editar un gráfico en el panel. Para obtener más información acerca de la creación de un gráfico, consulte Agregar o eliminar un gráfico desde un panel de CloudWatch.

New interface

Para editar un gráfico en un panel

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Dashboards (Paneles) y, a continuación, elija un panel.
3. En la esquina superior derecha del gráfico que desea editar, elija Widget actions (Acciones del widget) y, luego, elija Edit (Editar).
4. Para cambiar el título del gráfico, elija el icono de lápiz y papel ubicado junto al título actual. Ingrese el nuevo título y, a continuación, elija Apply (Aplicar).

5. (Opcional) Para cambiar el intervalo de tiempo del gráfico, seleccione uno de los intervalos de tiempo predefinidos en la parte superior del gráfico. Los intervalos de tiempo abarcan de 1 hora a 1 semana (1 h, 3 h, 12 h, 1 día, 3 días o 1 sem.).

   Para configurar su propio intervalo de tiempo, elija Custom (Personalizado).

   - (Opcional) Para que este widget siga utilizando el intervalo de tiempo que ha seleccionado, incluso si el intervalo de tiempo del resto del panel se modifica posteriormente, seleccione Persistir el intervalo de tiempo.

6. Puede cambiar el tipo de widget del gráfico por medio del menú desplegable que se encuentra junto a los intervalos de tiempo predefinidos.

7. En Graphed metrics (Métricas diagramadas), puede agregar una etiqueta dinámica a la métrica y cambiar la etiqueta de la métrica, el color de la etiqueta, la estadística y el periodo de la métrica. También puede determinar la posición de las etiquetas en el eje Y de izquierda a derecha.

   a. Para agregar una etiqueta dinámica a la métrica, elija Dynamic labels (Etiquetas dinámicas). Las etiquetas dinámicas muestran una estadística sobre la métrica en la leyenda del gráfico. Las etiquetas dinámicas se actualizan de manera automática cada vez que se actualiza el panel o el gráfico. De forma predeterminada, los valores dinámicos que agrega a las etiquetas aparecen al principio de las etiquetas. Para obtener más información, consulte Uso de etiquetas dinámicas.

   b. Para cambiar el color de una métrica, elija el cuadrado de color que se encuentra junto a ella.

   c. Para cambiar la estadística, elija el valor estadístico en la columna Statistic (Estadística) y, a continuación, elija un nuevo valor. Para obtener más información, consulte Statistics.

   d. Para cambiar el periodo, elija el valor del periodo ubicado debajo de la columna Period (Periodo) y, a continuación, elija un nuevo valor.

8. Para agregar o editar anotaciones horizontales, elija Options (Opciones):

   a. Para agregar una anotación horizontal, elija Add horizontal annotation (Añadir anotación horizontal).

   b. En Label (Etiqueta), elija el icono de lápiz y papel ubicado junto a la anotación actual. A continuación, ingrese la nueva anotación. Después de ingresar la anotación, elija el icono de marca de verificación.

c. En Value (Valor), elija el icono de lápiz y papel ubicado junto al valor actual de la métrica. A continuación, ingrese el nuevo valor de la métrica. Después de ingresar su valor, seleccione el icono de marca de verificación.

d. En Fill (Completar), elija el menú desplegable que se encuentra debajo de la columna y, a continuación, especifique cómo utilizará el sombreado la anotación. Puede elegir None (Ninguno), Above (Arriba), Between (Entre) o Below (Abajo). Si elige Between (Entre), aparecerá otro nuevo campo de etiqueta y de valor.

> **Tip**
>
> Puede cambiar el color de relleno eligiendo el cuadrado de color ubicado junto a la anotación.

e. En Axis (Eje), especifique si su anotación debe aparecer en el lado izquierdo o derecho del eje Y.

f. Para ocultar una anotación, anule la selección de la casilla de verificación situada junto a la anotación que desea ocultar en el gráfico.

g. Para eliminar una anotación, elija X en la columna Actions (Acciones).

> **Note**
>
> Puede repetir estos pasos para agregar varias anotaciones horizontales al mismo gráfico.

9. Para agregar o editar anotaciones verticales, elija Options (Opciones):

a. Para agregar una anotación vertical, elija Add vertical annotation (Añadir anotación vertical).

b. En Label (Etiqueta), elija el icono de lápiz y papel ubicado junto a la anotación actual. A continuación, ingrese la nueva anotación. Después de ingresar la anotación, elija el icono de marca de verificación.

> **Tip**
>
> Para mostrar solo la fecha y la hora, deje vacío el campo Label (Etiqueta).

c. En Date (Fecha), elija la fecha y la hora actuales. A continuación, ingrese las nuevas fecha y hora.

d. En Fill (Completar), elija el menú desplegable que se encuentra debajo de la columna y, a continuación, especifique cómo utilizará el sombreado la anotación. Puede elegir None (Ninguno), Above (Arriba), Between (Entre) o Below (Abajo). Si elige Between (Entre), aparecerá un nuevo campo de etiqueta y de valor.

> **Tip**
>
> Puede cambiar el color de relleno eligiendo el cuadrado de color ubicado junto a la anotación.

> **Note**
>
> Puede repetir estos pasos para agregar varias anotaciones verticales al mismo gráfico.

e. Para ocultar una anotación, anule la selección de la casilla de verificación situada junto a la anotación que desea ocultar en el gráfico.

f. Para eliminar una anotación, elija X en la columna Actions (Acciones).

10. Para personalizar el eje Y, elija Options (Opciones). Puede ingresar una etiqueta personalizada en Label (Etiqueta) debajo del Left Y-axis (Eje Y izquierdo). Si el gráfico muestra valores en el eje Y derecho, también puede personalizar esa etiqueta. Además, puede establecer mínimos y máximos en los valores del eje Y, de manera que el gráfico muestre solo el intervalo de valores que usted especifique.
11. Cuando termine de realizar los cambios, seleccione Update widget (Actualizar widget).

Para ocultar o cambiar la posición de una leyenda del gráfico

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Dashboards (Paneles) y, a continuación, elija un panel.
3. En la esquina superior derecha del gráfico que desea editar, elija Widget actions (Acciones del widget). Elija Legend (Leyenda) y, luego, seleccione Hidden (Oculta), Bottom (Inferior) o Right (Derecha).

Para ocultar las métricas de forma temporal en un gráfico de un panel

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Dashboards (Paneles) y, a continuación, elija un panel.
3. Seleccione el cuadrado de color de la métrica que desea ocultar en el pie de página del gráfico. Aparecerá una X en el cuadrado de color cuando pase el cursor sobre él, y el cuadrado se volverá gris cuando lo elija.
4. Para restaurar la métrica oculta, borre la X del cuadrado gris.

Original interface

Para editar un gráfico en un panel

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Dashboards (Paneles) y, a continuación, elija un panel.
3. Coloque el cursor sobre la esquina superior derecha del gráfico que desea editar. Seleccione Widget actions (Acciones del widget) y, luego, Edit (Editar).
4. Para cambiar el título del gráfico, elija el icono del lápiz ubicado junto al título actual y, a continuación, ingrese el título nuevo.
5. Para cambiar el intervalo de tiempo del gráfico, elija uno de los intervalos de tiempo predefinidos en la esquina superior del gráfico. Estos abarcan de 1 hora a 1 semana (1 h, 3 h, 12 h, 1 día, 3 días o 1 sem.).
   - Para configurar su propio intervalo de tiempo, elija Custom (Personalizado).
6. Para cambiar el tipo de widget del gráfico, seleccione la pestaña Graph options (Opciones de gráfico). Puede elegir Line (Línea), Stacked area (Área apilada), Number (Número), Bar (Barra) o Pie (Circular).

   > **Tip**
   >
   > Puede cambiar el tipo de widget del gráfico por medio del menú desplegable que se encuentra junto a los intervalos de tiempo predefinidos.

7. En Graphed metrics (Métricas diagramadas), puede agregar una etiqueta dinámica a la métrica y cambiar la etiqueta de la métrica, el color de la etiqueta, la estadística y el periodo

de la métrica. También puede determinar la posición de las etiquetas en el eje Y de izquierda a derecha.

a. Para agregar una etiqueta dinámica a la métrica, elija Dynamic labels (Etiquetas dinámicas). Las etiquetas dinámicas muestran una estadística sobre la métrica en la leyenda del gráfico. Las etiquetas dinámicas se actualizan de manera automática cada vez que se actualiza el panel o el gráfico. De forma predeterminada, los valores dinámicos que agrega a las etiquetas aparecen al principio de las etiquetas. Para obtener más información, consulte Uso de etiquetas dinámicas.

b. Para cambiar el color de una métrica, elija el cuadrado de color que se encuentra junto a ella.

c. Para cambiar la estadística, elija el valor estadístico en la columna Statistic (Estadística) y, a continuación, elija un nuevo valor. Para obtener más información, consulte Statistics.

d. Para cambiar el periodo, elija el valor del periodo ubicado debajo de la columna Period (Periodo) y, a continuación, elija un nuevo valor.

8. Para agregar o editar anotaciones horizontales, elija Graph options:

a. Para agregar una anotación horizontal, elija Add horizontal annotation (Añadir anotación horizontal).

b. En Label (Etiqueta), elija el icono de lápiz ubicado junto a la anotación actual. A continuación, ingrese la nueva anotación. Después de ingresar la anotación, elija el icono de marca de verificación.

c. En Value (Valor), elija el icono de lápiz ubicado junto al valor actual de la métrica. A continuación, ingrese el nuevo valor de la métrica. Después de ingresar su valor, seleccione el icono de marca de verificación.

d. En Fill (Completar), elija el menú desplegable que se encuentra debajo de la columna y, a continuación, especifique cómo utilizará el sombreado la anotación. Puede elegir None (Ninguno), Above (Arriba), Between (Entre) o Below (Abajo). Si elige Between (Entre), aparecerá un nuevo campo de etiqueta y de valor.

> **Tip**
>
> Puede cambiar el color de relleno eligiendo el cuadrado de color ubicado junto a la anotación.

e. En Axis (Eje), especifique si su anotación debe aparecer en el lado izquierdo o derecho del eje Y.

f. Para ocultar una anotación, anule la selección de la casilla de verificación situada junto a la anotación que desea ocultar en el gráfico.

g. Para eliminar una anotación, elija X en la columna Actions (Acciones).

> **Note**
>
> Puede repetir estos pasos para agregar varias anotaciones horizontales al mismo gráfico.

9. Para agregar o editar anotaciones verticales, elija Graph options (Opciones de gráfico):

a. Para agregar una anotación vertical, elija Add vertical annotation (Añadir anotación vertical).

b. En Label (Etiqueta), elija el icono de lápiz ubicado junto a la anotación actual. A continuación, ingrese la nueva anotación. Después de ingresar la anotación, elija el icono de marca de verificación.

> **Tip**
>
> Para mostrar solo la fecha y la hora, deje vacío el campo Label (Etiqueta).

c. En Date (Fecha), elija el icono de lápiz ubicado junto a la fecha y la hora actuales. A continuación, ingrese las nuevas fecha y hora.

d. En Fill (Completar), elija el menú desplegable que se encuentra debajo de la columna y, a continuación, especifique cómo utilizará el sombreado la anotación. Puede elegir None (Ninguno), Above (Arriba), Between (Entre) o Below (Abajo). Si elige Between (Entre), aparecerá un nuevo campo de etiqueta y de valor.

> **Tip**
>
> Puede cambiar el color de relleno eligiendo el cuadrado de color ubicado junto a la anotación.

> **Note**
>
> Puede repetir estos pasos para agregar varias anotaciones verticales al mismo gráfico.

e. Para ocultar una anotación, anule la selección de la casilla de verificación situada junto a la anotación que desea ocultar en el gráfico.

f. Para eliminar una anotación, elija X en la columna Actions (Acciones).

10. Para personalizar el eje Y, elija Graph options (Opciones del gráfico). Puede ingresar una etiqueta personalizada en Label (Etiqueta) debajo del Left Y-axis (Eje Y izquierdo). Si el gráfico muestra valores en el eje Y derecho, también puede personalizar esa etiqueta. Además, puede establecer mínimos y máximos en los valores del eje Y, de manera que el gráfico muestre solo el intervalo de valores que usted especifique.
11. Cuando termine de realizar los cambios, seleccione Update widget (Actualizar widget).

Para ocultar o cambiar la posición de una leyenda del gráfico

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Dashboards (Paneles) y, a continuación, elija un panel.
3. Coloque el cursor sobre la esquina superior derecha del gráfico que desea editar y, a continuación, elija Widget actions (Acciones del widget). Elija Legend (Leyenda) y, luego, seleccione Hidden (Oculta), Bottom (Inferior) o Right (Derecha).

Para ocultar las métricas de forma temporal en un gráfico de un panel

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Dashboards (Paneles) y, a continuación, elija un panel.
3. Seleccione el cuadrado de color de la métrica que desea ocultar en el pie de página del gráfico. Aparecerá una X en el cuadrado de color cuando pase el cursor sobre él, y el cuadrado se volverá gris cuando lo elija.
4. Para restaurar la métrica oculta, borre la X del cuadrado gris.

Mueva o cambie el tamaño de un gráfico en un panel de CloudWatch.

Organice y cambie el tamaño de gráficos en su panel de CloudWatch.

Para mover un gráfico en un panel

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Dashboards (Paneles) y, a continuación, elija un panel.
3. Realice una de las acciones siguientes:
   - Coloque el cursor sobre el título del gráfico hasta que aparezca el icono de selección. Seleccione y arrastre el gráfico a una nueva ubicación en el panel.
   - Para mover el widget a la parte superior izquierda o inferior izquierda del panel, seleccione los puntos suspensivos verticales en la parte superior derecha del widget para abrir el menú de acciones del widget. A continuación, seleccione Mover y elija a dónde quiere mover el widget.
4. Elija Save dashboard (Guardar panel).

Para cambiar el tamaño de un gráfico

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Dashboards (Paneles) y, a continuación, elija un panel.
3. Para aumentar o reducir el tamaño, coloque el cursor sobre el gráfico y, a continuación, arrastre la esquina inferior derecha del gráfico. Deje de arrastrar la esquina cuando haya alcanzado el tamaño deseado.
4. Elija Save dashboard (Guardar panel).

Para ampliar un gráfico de forma temporal

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Dashboards (Paneles) y, a continuación, elija un panel.
3. Seleccione el gráfico. O bien coloque el cursor sobre el título del gráfico y elija Widget actions, Enlarge.

Cambie el nombre de un gráfico en un panel de CloudWatch

Cambie el nombre predeterminado que CloudWatch le asigna a un gráfico en el panel.

Para cambiar el nombre de un gráfico en un panel

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Dashboards (Paneles) y, a continuación, elija un panel.
3. Coloque el cursor sobre el título del gráfico y elija Widget actions (Acciones del widget), Edit (Editar).
4. En la pantalla Edit graph (Editar gráfico), cerca de la parte superior, elija el título del gráfico.
5. Para Title (Título), escriba un nuevo nombre y seleccione Ok (Aceptar) (marca de verificación). En la parte inferior derecha de la pantalla Edit graph (Editar gráfico), elija Update widget (Actualizar widget).

Agregue un widget explorador de métricas a un panel de CloudWatch

Los widgets exploradores de métricas incluyen gráficos de varios recursos que tienen la misma etiqueta o comparten la misma propiedad de recurso, como un tipo de instancia. Estos widgets se mantienen actualizados, ya que los recursos que concuerden se crean o eliminan. La adición de widgets exploradores de métricas al panel le ayuda a solucionar problemas de su entorno de manera más eficiente.

Por ejemplo, monitoree la flota de instancias EC2 mediante etiquetas que representen los entornos, como producción o prueba. A continuación, utilice estas etiquetas para filtrar y agregar las métricas operativas, como `CPUUtilization`, para comprender el estado y el rendimiento de las instancias EC2 asociadas a cada etiqueta.

En los siguientes pasos se explica cómo se agrega un widget explorador de métricas a un panel mediante la consola. También puede agregarla mediante programación o mediante AWS CloudFormation. Para obtener más información, consulte Metrics Explorer Widget Object Definition (Definición de objeto del widget explorador de métricas) y AWS::CloudWatch::Dashboard.

Para agregar un widget explorador de métricas a un panel

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Dashboards (Paneles).
3. Elija el nombre del panel al que desea agregar el widget explorador de métricas.
4. Elija el símbolo +.
5. Seleccione Explorer (Explorador) y luego elija Next (Siguiente).

> **Note**
>
> Para agregar un widget explorador de métricas, es necesario que el usuario esté conectado a la nueva vista del panel. Para ello, elija Dashboards (Paneles) en el panel de navegación y, después, try out the new interface (probar la nueva interfaz) en el banner de la parte superior de la página.

6. Realice una de las acciones siguientes:
   - Para usar una plantilla, elija Pre-filled Explorer widget (Widget Explorador precargado) y, a continuación, seleccione una plantilla para utilizarla.
   - Para crear una visualización personalizada, elija Empty Explorer widget (Vaciar el widget explorador).
7. Seleccione Crear.

   Si ha utilizado una plantilla, el widget aparecerá en el panel con las métricas que seleccionó. Si está satisfecho con el widget explorador y el panel, elija Save dashboard (Guardar panel).

   Si no ha utilizado una plantilla, siga los pasos que se describen a continuación.
8. En el nuevo widget en Explorer (Explorador), en el cuadro Metrics (Métricas), elija una sola métrica o todas las métricas disponibles de un servicio.

   Después de elegir una métrica, repita este paso de forma opcional para agregar más métricas.
9. Para cada métrica que seleccione, CloudWatch muestra la estadística que utilizará inmediatamente después del nombre de métrica. Para cambiarlo, elija el nombre de la estadística y, luego, elija la estadística que desee.
10. En From (Desde), elija una etiqueta o una propiedad de recurso para filtrar los resultados.

    Después de hacer esto, puede optar por repetir este paso para elegir más etiquetas o propiedades de recurso.

    Si elige varios valores de la misma propiedad, como dos tipos de instancias de EC2, el explorador mostrará todos los recursos que concuerden con cualquiera de las propiedades elegidas. Se trata como a una operación lógica OR.

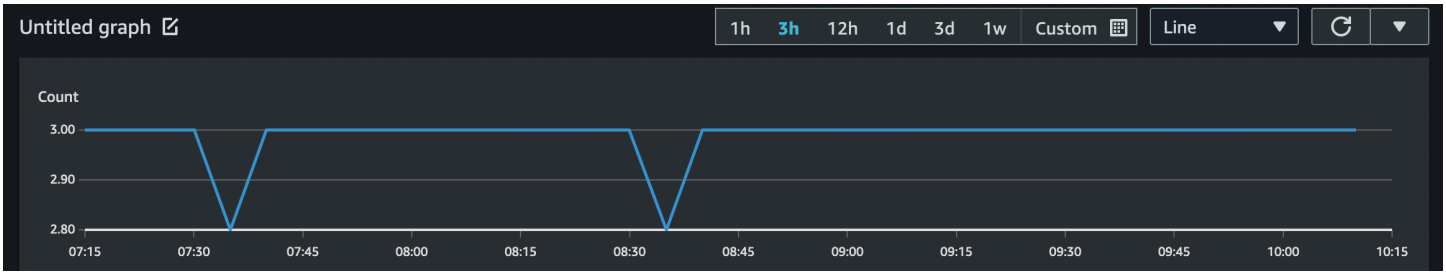
    Si elige diferentes propiedades o etiquetas, como la etiqueta de **`Production`** y el tipo de instancia M5, sólo se mostrarán los recursos que concuerden con todas estas selecciones. Esto se trata como una operación Y.

11. (Opcional) Para Aggregate by (Agregar por), elija una estadística que se utilizará para agregar las métricas. Luego, junto a for (Para), elija cómo agregar la métrica de la lista. Puede agregar todos los recursos que se muestran actualmente o agregarlos mediante una sola etiqueta o propiedad de recurso.

    De acuerdo al modo que elija para agregar, el resultado puede ser una sola serie temporal o varias series temporales.
12. Bajo el título Split by (Dividir por), puede elegir dividir un único gráfico con varias series temporales en diferentes gráficos. La división se puede hacer por una variedad de criterios que usted elige enSplit by (Dividir por).
13. En Graph options (Opciones de gráficos), refine el gráfico mediante un cambio de período, el tipo de gráfico, la ubicación de la leyenda y el diseño.
14. Si está satisfecho con el widget explorador y el panel, elija Save dashboard (Guardar panel).

Agregue o elimine un widget de línea desde un panel de CloudWatch

Con el widget de línea, puede comparar las métricas durante periodos de tiempo. También puede utilizar la característica de zoom de minimapa del widget para inspeccionar secciones de gráficos de líneas sin cambiar entre vistas ampliadas y alejadas. En los procedimientos de esta sección, se describe cómo agregar y quitar un widget de línea en un panel de CloudWatch. Para obtener información acerca del uso de la característica de zoom de minimapa del widget con gráficos de líneas, consulte Ampliar un gráfico de línea o área apilada.



Para agregar un widget de línea a un panel

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Dashboards (Paneles) y, a continuación, elija un panel.
3. Elija el símbolo + y seleccione Line (Línea).
4. Elija Metrics (Métricas).
5. Elija Browse (Navegar) y seleccione la métrica que desea graficar.

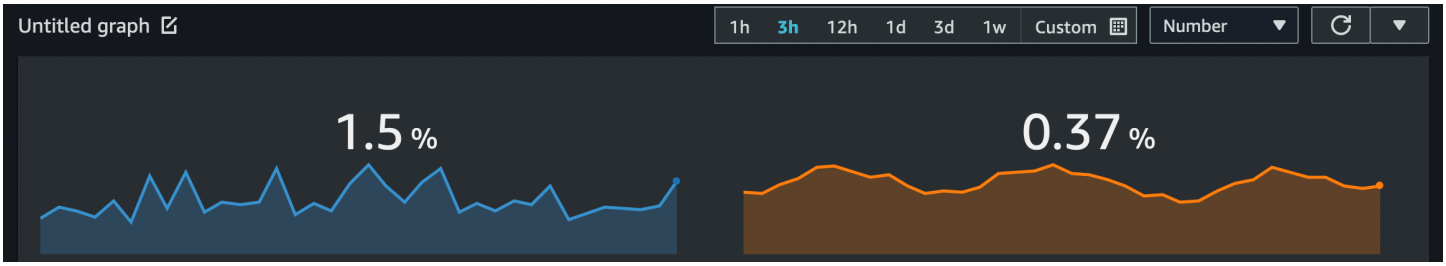
6. Elija Create widget (Crear un widget) y, a continuación, elija Save dashboard (Guardar panel).

Para eliminar un widget de línea desde un panel

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Dashboards (Paneles) y, a continuación, elija un panel.
3. En la esquina superior derecha del widget de línea que desea eliminar, elija Widget actions (Acciones del widget) y, luego, elija Delete (Eliminar).
4. Elija Save dashboard (Guardar panel).

Añada o elimine un widget numérico desde un panel de CloudWatch

Con el widget numérico, puede ver los valores y tendencias de las métricas más recientes tan pronto como aparecen. Dado que el widget numérico incluye la característica de minigráfico, puede visualizar las mitades superior e inferior de las tendencias métricas en un solo gráfico. En los procedimientos de esta sección, se describe cómo agregar y quitar un widget numérico de un panel de CloudWatch.



> **Note**
>
> Solo la nueva interfaz admite la característica de minigráfico. Al crear un widget numérico, la característica de minigráfico se incluye automáticamente.

Para añadir un widget numérico a un panel

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Dashboards (Paneles) y, a continuación, elija un panel.
3. Elija el símbolo + y seleccione Number (Número).
4. En la pestaña Examinar, busque o examine la métrica que desea mostrar.

5. (Opcional) Para cambiar el color de la característica de minigráfico, elija Graphed metrics (Métricas diagramadas) y seleccione el cuadro de color situado junto a la etiqueta métrica. Aparece un menú en el que se puede elegir un color diferente o ingresar un código de color hexadecimal de seis dígitos para especificar un color.
6. (Opcional) Para desactivar la característica de minigráfico, elija Options (Opciones). En Sparkline (Minigráfico), quite la marca de la casilla de verificación.
7. (Opcional) Para cambiar el intervalo de tiempo del gráfico, seleccione uno de los intervalos de tiempo predefinidos en la parte superior del widget. Los intervalos de tiempo abarcan de 1 hora a 1 semana (1 h, 3 h, 12 h, 1 día, 3 días o 1 sem.).

   Para configurar su propio intervalo de tiempo, elija Custom (Personalizado).

   - (Opcional) Para que este widget siga utilizando el intervalo de tiempo que ha seleccionado, incluso si el intervalo de tiempo del resto del panel se modifica posteriormente, seleccione Persistir el intervalo de tiempo.
8. (Opcional) Para que el widget numérico muestre un agregado (1h, 3h, 12h, 1d, 3d o 1s).

   Para configurar su propio intervalo de tiempo, elija Custom (Personalizado).

   - (Opcional) Para que este widget muestre un promedio del valor de la métrica en todo el intervalo de tiempo, en lugar del valor más reciente, elija Opciones, El valor del rango de tiempo muestra el valor de todo el intervalo de tiempo.
9. Elija Create widget (Crear un widget), y elija Save dashborad (Guardar panel).

> **Tip**
>
> Puede desactivar la característica de minigráfico desde el widget numérico de la pantalla del panel. En la esquina superior derecha del widget numérico que desea modificar, elija Widget actions (Acciones del widget). Seleccione Sparkline (Minigráfico) y, a continuación, elija Hide Sparkline (Ocultar minigráfico).

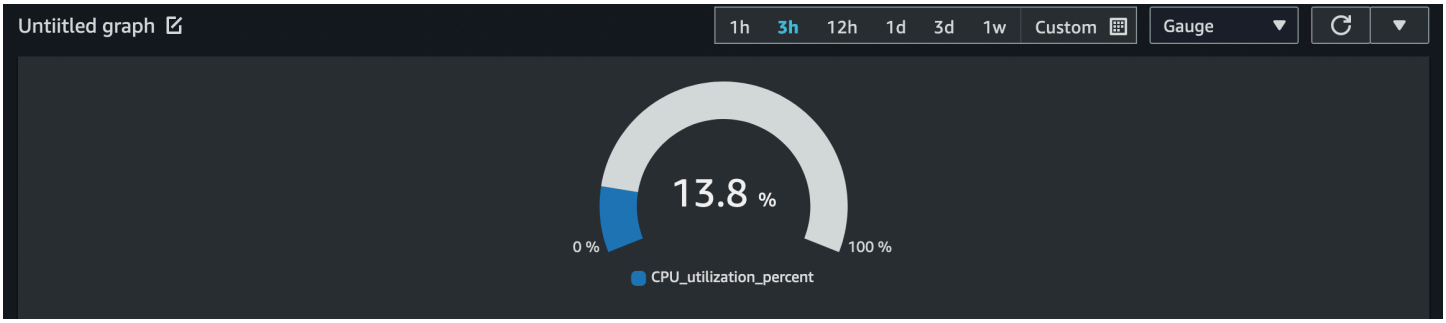
Para eliminar un widget numérico desde un panel

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Dashboards (Paneles) y, a continuación, elija el panel que contiene el widget numérico que desea eliminar.

3. En la esquina superior derecha del widget numérico que desea eliminar, elija Widget actions (Acciones del widget) y, luego, elija Delete (Eliminar).
4. Elija Save dashboard (Guardar panel).

Agregue o elimine un widget de calibre desde un panel de CloudWatch

Con el widget de calibre, puede visualizar los valores de métrica que van entre los rangos. Por ejemplo, puede utilizar el widget de calibre para graficar porcentajes y utilización de la CPU, de modo que pueda observar y diagnosticar cualquier problema de rendimiento que se presente. En los procedimientos de esta sección se describe cómo agregar y quitar un widget de calibre desde un panel de CloudWatch.



> **Note**
>
> Solo la nueva interfaz de la consola de CloudWatch admite la creación del widget de calibre. Debe establecer un rango de calibre al crear este widget.

Para agregar un widget de calibre a un panel

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Dashboards (Paneles) y, a continuación, elija un panel.
3. En la pantalla del panel, elija el símbolo + y, a continuación, seleccione Gauge (Calibre).
4. Elija Browse (Navegar) y luego seleccione la métrica que desea graficar.
5. Elija Options. En Gauge range (Rango de calibre), establezca valores para Min (Mínimo) y Max (Máximo). En los porcentajes, como la utilización de la CPU, le recomendamos que establezca los valores de `Min` en `0` y `Max` en `100`.
6. (Opcional) Para cambiar el color del widget de calibre, elija Graphed metrics (Métricas diagramadas) y seleccione el cuadro de color situado junto a la etiqueta métrica. Aparece un

menú en el que se puede elegir un color diferente o ingresar un código de color hexadecimal de seis dígitos para especificar un color.

7. Elija Create widget (Crear un widget), y elija Save dashborad (Guardar panel).

Para eliminar un widget de calibre desde un panel

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Dashboards (Paneles) y, a continuación, elija el panel que contiene el widget de calibre que desea eliminar.
3. En la esquina superior derecha del widget de calibre que desea eliminar, elija Widget actions (Acciones del widget) y elija Delete (Eliminar).
4. Elija Save dashboard (Guardar panel).

Agregue un widget personalizado a un panel de CloudWatch

Un widget personalizado es un widget del panel de CloudWatch que puede llamar a cualquier función de AWS Lambda con parámetros personalizados. A continuación, muestra el HTML o JSON devuelto. Los widgets personalizados son una forma sencilla de crear una vista de datos personalizada en un panel. Si ingresa el código de Lambda y crea el HTML, puede crear un widget personalizado útil. Además, Amazon proporciona varios widgets personalizados preconfigurados que puede crear sin ningún código.

Cuando cree una función Lambda para utilizarla como widget personalizado, se recomienda que incluya el prefijo customWidget en el nombre de la función. Le ayuda a saber cuáles de las funciones Lambda son seguras para usar cuando agrega widgets personalizados al panel.

Los widgets personalizados se comportan como otros widgets en el panel. Se pueden actualizar manualmente o automáticamente, cambiar de tamaño y también se pueden mover. Reaccionan al rango de tiempo del panel.

Si ha configurado la funcionalidad de cuentas cruzadas de la consola de CloudWatch, puede agregar un widget personalizado que ha creado en una cuenta a los paneles de otras cuentas. Para obtener más información, consulte Consola de CloudWatch para cuentas y Regiones cruzadas.

También puede utilizar widgets personalizados en su propio sitio web mediante la característica de uso compartido del panel de CloudWatch. Para obtener más información, consulte Compartir paneles de CloudWatch.

Temas

Detalles sobre los widgets personalizados

Los widgets personalizados funcionan de la siguiente manera:

1. El panel de CloudWatch llama a la función Lambda que contiene el código del widget. Se especifica en cualquier parámetro personalizado que se defina en el widget.
2. La función Lambda muestra una cadena de HTML, JSON o de Markdown. Markdown se muestra como JSON en el siguiente formato:

   ```
   {"markdown":"markdown content"}
   ```

3. El panel muestra el HTML o JSON devuelto.

Si la función devuelve el HTML, la mayoría de las etiquetas HTML serán compatibles. Utilice estilos de Cascading Style Sheets (CSS) y Gráficos vectoriales escalables (SVG) para crear vistas sofisticadas.

El estilo predeterminado de los elementos HTML, como los enlaces y las tablas, siguen el estilo de los paneles de CloudWatch. Personalice este estilo con los estilos integrados mediante la etiqueta `<style>`. También puede desactivar los estilos predeterminados si incluye un único elemento HTML con la clase de `cwdb-no-default-styles`. En el siguiente ejemplo se desactivan los estilos predeterminados: `<div class="cwdb-no-default-styles"></div>`.

Cada llamada por un widget personalizado a Lambda incluye un elemento `widgetContext` con los siguientes contenidos, para proporcionar al desarrollador de funciones Lambda información de contexto útil.

```
{
    "widgetContext": {
```

```
        "dashboardName": "Name-of-current-dashboard",
        "widgetId": "widget-16",
        "accountId": "012345678901",
        "locale": "en",
        "timezone": {
            "label": "UTC",
            "offsetISO": "+00:00",
            "offsetInMinutes": 0
        },
        "period": 300,
        "isAutoPeriod": true,
        "timeRange": {
            "mode": "relative",
            "start": 1627236199729,
            "end": 1627322599729,
            "relativeStart": 86400012,
            "zoom": {
                "start": 1627276030434,
                "end": 1627282956521
             }
        },
        "theme": "light",
        "linkCharts": true,
        "title": "Tweets for Amazon website problem",
        "forms": {
            "all": {}
        },
        "params": {
            "original": "param-to-widget"
        },
        "width": 588,
        "height": 369
    }
}
```

Estilo CSS predeterminado

Los widgets personalizados proporcionan los siguientes elementos de estilo CSS predeterminados:

- Utilice la clase CSS btn para agregar un botón. Un botón se convierte en ancla (<a>) como en el siguiente ejemplo:

```
<a class="btn" href=https://amazon.com”>Open Amazon</a>
```

- Utilice la clase CSS btn btn-primary para agregar un botón principal.
- Los siguientes elementos tienen un estilo de forma predeterminada: table (tabla),select (seleccionar), headers (H1, H2 y H3) (cabeceras), preformatted text (pre) (texto preformateado [pre]), input (entrada) y text area (área de texto).

Uso del parámetro de descripción

Se recomienda que utilice el parámetro describe (describir) en las funciones, incluso si solo muestra una cadena vacía. Si no lo admite, y se llama en el widget personalizado, muestra el contenido del widget como si fuera documentación.

Si incluye el parámetro describe (describir), la función Lambda devuelve la documentación en formato Markdown y no realiza nada más.

Cuando se crea un widget personalizado en la consola, después de seleccionar la función Lambda, aparecerá un botón Get documentation (Obtener documentación). Si elige este botón, la función se invoca con el parámetro describe (describir) y se muestra la documentación de la función. Si la documentación está bien formateada en markdown, CloudWatch analiza la primera entrada de la documentación que está rodeada de tres comillas (```) en YAML. A continuación, rellena automáticamente la documentación en los parámetros. A continuación, se muestra un ejemplo de esta documentación bien formateada.

```
``` yaml
echo: <h1>Hello world</h1>
```
```

Seguridad y JavaScript

Por razones de seguridad, JavaScript no está permitido en el HTML que se devuelve. La eliminación de JavaScript evita problemas de escalamiento de permisos, donde el programador de la función Lambda inyecta código que podría ejecutarse con permisos superiores que el usuario que ve el widget en el panel.

Si el HTML devuelto contiene algún código JavaScript u otras vulnerabilidades de seguridad conocidas, se limpia del HTML antes de que se ejecute en el panel. Por ejemplo, las etiquetas **<iframe>** y **<use>** no están permitidas y se eliminan.

Los widgets personalizados no se ejecutarán de forma predeterminada en un panel. En su lugar, debe permitir explícitamente que se ejecute un widget personalizado si confía en la función Lambda

que invoca. Puede optar por permitirlo una vez o permitirlo siempre, tanto para widgets individuales como para todo el panel. También puede denegar el permiso para widgets individuales y para todo el panel.

Interactividad en el widget personalizado

Aunque JavaScript no está permitido, hay otras formas de permitir la interactividad con el HTML devuelto.

- Cualquier elemento en el HTML devuelto se puede etiquetar con una configuración especial en una etiqueta `<cwdb-action>`, que muestra información en ventanas emergentes, pide confirmación de clics y llama a cualquier función Lambda cuando se elige ese elemento. Por ejemplo, puede definir botones que llaman a una API de AWS cualquiera mediante el uso de una función Lambda. El HTML devuelto se puede configurar para reemplazar el contenido del widget Lambda existente o para mostrarlo dentro de un modal.
- El HTML devuelto puede incluir vínculos que abren nuevas consolas, que abren otras páginas de clientes o que cargan otros paneles.
- El HTML puede incluir el atributo `title` para un elemento, que proporciona información adicional si el usuario sitúa el cursor en ese elemento.
- El elemento puede incluir selectores CSS, como `:hover`, que puede invocar animaciones u otros efectos CSS. También puede mostrar u ocultar elementos en la página.

Definición y uso de <cwdb-action>

El elemento `<cwdb-action>` define un comportamiento en el elemento inmediatamente anterior. El contenido de `<cwdb-action>` es HTML para mostrar o un bloque JSON de parámetros para pasar a una función Lambda.

A continuación se muestra un ejemplo de un elemento `<cwdb-action>`.

```
<cwdb-action
     action="call|html"
     confirmation="message"
     display="popup|widget"
     endpoint="<lambda ARN>"
     event="click|dblclick|mouseenter">

     html | params in JSON
</cwdb-action>
```

- acción: Lls valores válidos son `call`, que llama a una función Lambda, y `html`, que muestra cualquier HTML contenido en `<cwdb-action>`. El valor predeterminado es `html`.
- confirmación: muestra un mensaje de confirmación que se debe confirmar antes de que se lleve a cabo la acción, lo que le permite al cliente la cancelación.
- visualización: los valores válidos son `popup` y `widget`, que reemplaza el contenido del widget en sí. El valor predeterminado es `widget`.
- punto de enlace: el Nombre de recurso de Amazon (ARN) de la función Lambda que se va a invocar. Esto es obligatorio si la `action` es `call`.
- evento: define el evento del elemento anterior que invoca la acción. Los valores válidos son `click`, `dblclick` y `mouseenter`. El evento `mouseenter` solo se puede utilizar en combinación con la acción `html`. El valor predeterminado es `click`.

Ejemplos

A continuación, se muestra un ejemplo de cómo se utiliza la etiqueta `<cwdb-action>` para crear un botón que reinicie una instancia de Amazon EC2 mediante una llamada a función de Lambda. Muestra si la llamada fue exitosa o fallida en una ventana emergente.

```
<a class="btn">Reboot Instance</a>
<cwdb-action action="call" endpoint="arn:aws:lambda:us-
east-1:123456:function:rebootInstance" display="popup">
       { "instanceId": "i-342389adbfef" }
</cwdb-action>
```

El siguiente ejemplo muestra más información en una ventana emergente.

```
<a>Click me for more info in popup</a>
<cwdb-action display="popup">
   <h1>Big title</h1>
   More info about <b>something important</b>.
</cwdb-action>
```

Este es un ejemplo del botón Next (Siguiente) que reemplaza el contenido de un widget con una llamada a una función Lambda.

```
<a class="btn btn-primary">Next</a>
<cwdb-action action="call" endpoint="arn:aws:lambda:us-
east-1:123456:function:nextPage">
```

```
   { "pageNum": 2 }
</cwdb-action>
```

Cree un widget personalizado

Para crear un widget personalizado, puede utilizar una de las muestras proporcionadas por AWS, o puede crear una. Las muestras de AWS incluyen muestras tanto en JavaScript como en Python y las crea una pila de AWS CloudFormation. Para ver una lista de muestras, consulte Muestras de widgets personalizados.

Para crear un widget personalizado en un panel de CloudWatch

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Dashboards (Paneles) y, a continuación, elija un panel.
3. Elija el símbolo +.
4. Seleccione Custom widget (Widget personalizado).
5. Utilice uno de los siguientes métodos:
   - Para utilizar un widget personalizado de muestra que AWS ha proporcionado, realice lo siguiente:
     a. Seleccione la muestra en el cuadro desplegable.

        La consola de AWS CloudFormation se lanza en un navegador nuevo. En la consola de AWS CloudFormation, haga lo siguiente:

     b. (Opcional) Personalice el nombre de la pila de AWS CloudFormation.
     c. Realice selecciones para cualquier parámetro que la muestra utilice.
     d. Seleccione Acepto que AWS CloudFormation puede crear recursos de IAM y, luego, seleccione Crear pila.
   - Para crear su propio widget personalizado que AWS proporciona, realice lo siguiente:
     a. Elija Siguiente.
     b. Elija entre seleccionar la función de Lambda de una lista o ingrese el nombre de recurso de Amazon (ARN). Si lo selecciona de una lista, especifique también la región donde se encuentra la función y la versión que se va a utilizar.
     c. Para Parameters (Parámetros), haga selecciones para cualquier parámetro que la función utilice.

d. Ingrese un título para el widget.

e. Para Update on (Actualizar el), configure cuándo debe actualizarse el widget (cuando se debe llamar de nuevo a la función de Lambda). Puede ser uno o varios de los siguientes: Refresh (Actualizar) para actualizarlo cuando el panel se actualice automáticamente, Resize (Cambiar el tamaño) para actualizarlo cada vez que se cambie el tamaño del widget, o Time Range (Intervalo de tiempo) para actualizarlo cada vez que se ajusta el intervalo de tiempo del panel, incluso cuando se amplían los gráficos.

f. Si está satisfecho con la versión preliminar, elija Create widget (Crear widget).

Muestras de widgets personalizados

AWS proporciona widgets personalizados de muestra tanto en JavaScript como en Python. Puede crear estos widgets de muestra con el enlace para cada widget de esta lista. De forma alternativa, puede crear y personalizar un widget mediante la consola de CloudWatch. Los enlaces en esta lista abren una consola de AWS CloudFormation y usan un enlace de creación rápida de AWS CloudFormation para crear el widget personalizado.

También puede acceder a las muestras de widget personalizados en GitHub.

Si sigue esta lista, se verán ejemplos completos del widget Echo para cada idioma.

JavaScript

Widgets personalizados de muestra en JavaScript

- echo: un echo básico que puede utilizar para probar cómo aparece HTML en un widget personalizado, sin tener que ingresar un widget nuevo.
- Hello world: un widget de inicio muy básico.
- Custom widget debugger (Widget personalizado para depuración): un widget de depuración que muestra información útil sobre el entorno en tiempo de ejecución de Lambda.
- Query CloudWatch Logs Insights (Consultas de CloudWatch Logs): ejecute y edite las consultas de CloudWatch Logs Insights.
- Run Amazon Athena queries (Ejecutar las consultas de Amazon Athena): ejecute y edite consultas de Athena.
- Llamada a la API de AWS: llame a cualquier API de AWS de solo lectura y muestre los resultados en formato JSON.

- Fast CloudWatch bitmap graph (Gráfico rápido de mapa de bits de CloudWatch): ejecute los gráficos de CloudWatch en el servidor para una visualización rápida.
- Text widget from CloudWatch dashboard (Widget de texto desde el panel de CloudWatch): muestra el primer widget de texto del panel de CloudWatch que se especifique.
- CloudWatch metric data as a table (Datos métricos de CloudWatch como tabla): muestra en una tabla los datos sin procesar de métricas de CloudWatch.
- Amazon EC2 table (Tabla de Amazon EC2): muestra las instancias EC2 principales en función del uso de la CPU. Este widget también incluye un botón de reinicio que está desactivado de forma predeterminada.
- Implementaciones de AWS CodeDeploy: muestra las implementaciones de CodeDeploy.
- Informe de AWS Cost Explorer: muestra un informe sobre el costo de cada servicio de AWS para el intervalo de tiempo que se seleccione.
- Display content of external URL (Visualización del contenido de la URL externa): muestra el contenido de una URL a la que se accede de manera externa.
- Display an Amazon S3 object (Visualización de un objeto de Amazon S3): muestra un objeto en un bucket de Amazon S3 en su cuenta.
- Simple SVG pie chart (Gráfico circular SVG simple): ejemplo de un widget gráfico basado en SVG.

Python

Widgets personalizados de muestra en Python

- Echo: un echo básico que se puede utilizar para probar cómo aparece HTML en un widget personalizado, sin tener que ingresar un widget nuevo.
- Hello world: un widget de inicio muy básico.
- Custom widget debugger (Widget personalizado para depuración): un widget de depuración que muestra información útil sobre el entorno en tiempo de ejecución de Lambda.
- Llamada a la API de AWS: llame a cualquier API de AWS de solo lectura y muestre los resultados en formato JSON.
- Fast CloudWatch bitmap graph (Gráfico rápido de mapa de bits de CloudWatch): ejecute los gráficos de CloudWatch con el servidor para una visualización rápida.
- Send dashboard snapshot by email (Envío de una instantánea del panel por email): tome una instantánea del panel actual y envíela a los destinatarios por email.

- Send dashboard snapshot to Amazon S3 (Envío de una instantánea del panel a Amazon S3): tome una instantánea del panel actual y guárdela en Amazon S3.
- Text widget from CloudWatch dashboard (Widget de texto desde el panel de CloudWatch): muestra el primer widget de texto del panel de CloudWatch que se especifique.
- Display content of external URL (Visualización del contenido de la URL externa): muestra el contenido de una URL a la que se accede de manera externa.
- RSS reader (Lector RSS): muestra fuentes RSS.
- Display an Amazon S3 object (Visualización de un objeto de Amazon S3): muestra un objeto en un bucket de Amazon S3 en su cuenta.
- Simple SVG pie chart (Gráfico circular SVG simple): ejemplo de un widget gráfico basado en SVG.

Widget Echo en JavaScript

El siguiente es el widget Echo de muestra en JavaScript.

```
const DOCS = `
## Echo
A basic echo script. Anything passed in the \`\`\`echo\`\`\` parameter is returned as
 the content of the custom widget.
### Widget parameters
Param | Description
---|---
**echo** | The content to echo back

### Example parameters
\`\`\` yaml
echo: <h1>Hello world</h1>
\`\`\`
`;

exports.handler = async (event) => {
    if (event.describe) {
        return DOCS;
    }

    let widgetContext = JSON.stringify(event.widgetContext, null, 4);
    widgetContext = widgetContext.replace(/</g, '&lt;');
    widgetContext = widgetContext.replace(/>/g, '&gt;');
```

```
    return `${event.echo || ''}<pre>${widgetContext}</pre>`;
};
```

Widget Echo en Python

El siguiente es el widget Echo de muestra en Python.

```
import json

DOCS = """
## Echo
A basic echo script. Anything passed in the ```echo``` parameter is returned as the
 content of the custom widget.
### Widget parameters
Param | Description
---|---
**echo** | The content to echo back

### Example parameters
``` yaml
echo: <h1>Hello world</h1>
```"""

def lambda_handler(event, context):
    if 'describe' in event:
        return DOCS

    echo = event.get('echo', '')
    widgetContext = event.get('widgetContext')
    widgetContext = json.dumps(widgetContext, indent=4)
    widgetContext = widgetContext.replace('<', '<')
    widgetContext = widgetContext.replace('>', '>')

    return f'{echo}<pre>{widgetContext}</pre>'
```

Widget Echo en Java

El siguiente es el widget Echo de muestra en Java.

```
package example;

import com.amazonaws.services.lambda.runtime.Context;
import com.amazonaws.services.lambda.runtime.RequestHandler;
```

```
import com.google.gson.Gson;
import com.google.gson.GsonBuilder;

public class Handler implements RequestHandler<Event, String>{

  static String DOCS = ""
    + "## Echo\n"
    + "A basic echo script. Anything passed in the ```echo``` parameter is returned as
 the content of the custom widget.\n"
    + "### Widget parameters\n"
    + "Param | Description\n"
    + "---|---\n"
    + "**echo** | The content to echo back\n\n"
    + "### Example parameters\n"
    + "```yaml\n"
    + "echo: <h1>Hello world</h1>\n"
    + "```\n";

  Gson gson = new GsonBuilder().setPrettyPrinting().create();

  @Override
  public String handleRequest(Event event, Context context) {

    if (event.describe) {
      return DOCS;
    }

    return (event.echo != null ? event.echo : "") + "<pre>" +
 gson.toJson(event.widgetContext) + "</pre>";
  }
}

class Event {

    public boolean describe;
    public String echo;
    public Object widgetContext;

    public Event() {}

    public Event(String echo, boolean describe, Object widgetContext) {
        this.describe = describe;
        this.echo = echo;
```

```
        this.widgetContext = widgetContext;
    }
}
```

Agregue o elimine un widget de texto desde un panel de CloudWatch

Un widget de texto contiene un bloque de texto en formato Markdown. Puede agregar, editar o eliminar widgets de texto desde el panel de CloudWatch.

Para añadir un widget de texto a un panel

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Dashboards (Paneles) y, a continuación, elija un panel.
3. Elija el símbolo +.
4. Seleccione Texto.
5. En Markdown (Marcado), agregue texto y dele formato utilizando Markdown (Marcado) y elija Create widget (Crear widget).
6. Para hacer que el widget de texto sea transparente, seleccione Fondo transparente.
7. Elija Save dashboard (Guardar panel).

Para editar un widget de texto en un panel

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Dashboards (Paneles) y, a continuación, elija un panel.
3. Coloque el cursor en la esquina superior derecha del bloque de texto y elija Widget actions (Acciones del widget). A continuación, elija Edit (Editar).
4. Actualice el texto según sea necesario y elija Update widget (Actualizar widget).
5. Elija Save dashboard (Guardar panel).

Para eliminar un widget de texto desde un panel

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Dashboards (Paneles) y, a continuación, elija un panel.
3. Coloque el cursor en la esquina superior derecha del bloque de texto y elija Widget actions (Acciones del widget). A continuación, elija Eliminar.

4. Elija Save dashboard (Guardar panel).

Agregue o elimine un widget de alarma desde un panel de CloudWatch

Para agregar un widget de alarma a un panel, elija una de las siguientes opciones:

- Agregue una sola alarma en un widget, que muestra el gráfico de la métrica de la alarma y también el estado de la alarma.
- Agregue un widget de estado de alarma, que muestra el estado de varias alarmas en una cuadrícula. Solo se muestran los nombres de las alarmas y el estado actual; los gráficos no se muestran. Puede incluir hasta 100 alarmas en un widget de estado de alarma.

Para agregar una sola alarma a un panel, incluido el gráfico,

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación elija Alarms (Alarmas), seleccione la alarma que desea agregar y, a continuación, elija Add to Dashboard (Añadir al panel).
3. Seleccione un panel, elija un tipo de widget (Line, Stacked area o Number) y, a continuación, seleccione Add to dashboard.
4. Para ver la alarma en el panel, elija Dashboards (Paneles) en el panel de navegación y seleccione el panel.
5. (Opcional) Para hacer que el gráfico de alarma sea mayor temporalmente, seleccione el gráfico.
6. (Opcional) Para cambiar el tipo de widget, sitúe el cursor sobre el título del gráfico, elija Acciones del widget y después Tipo de widget.

Para añadir un widget de estado de alarma a un panel

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Dashboards (Paneles) y, a continuación, elija un panel.
3. Elija el símbolo +.
4. Seleccione Alarm status (Estado de la alarma).
5. Seleccione las casillas de verificación junto a las alarmas que desea agregar al widget y, luego, elija Create widget (Crear widget).
6. Elija Add to dashboard (Añadir a panel).

Para eliminar un widget de alarma de un panel

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Dashboards (Paneles) y, a continuación, elija un panel.
3. Sitúe el cursor sobre el widget, elija Acciones del widget y luego Eliminar.
4. Elija Save dashboard (Guardar panel). Si intenta salir del panel antes de guardar los cambios, el sistema le pide que los guarde o los descarte.

Cómo agregar o eliminar un widget de tabla de datos desde un panel de CloudWatch

Con el widget de tabla de datos, puede observar los puntos de datos sin procesar de su métrica y un breve resumen de esos datos sin procesar. Como el widget de tabla de datos no es un gráfico para extraer los datos reales, es más fácil entender los puntos de datos que se presentan. En los procedimientos de esta sección se describe cómo agregar y eliminar un widget de tabla de datos desde un panel de CloudWatch.

Label	Min	Max	Sum	Average	11/20 06:00	11/20 00:00	11/19 18:00	11/19 12:00	11/ 06:
TestMetric295	991	1,000	12k	998	996	1,000	997	999	
TestMetric296	995	1,000	12k	998	995	1,000	1,000	998	
TestMetric297	991	1,000	12k	998	998	1,000	999	997	
TestMetric298	994	1,000	12k	997	996	999	995	995	
TestMetric3	993	1,000	12k	998	1,000	999	999	1,000	
TestMetric299	995	999	12k	998	999	995	999	998	
TestMetric30	994	999	12k	998	999	998	999	999	
StackMetric2	99	99.9	1.2k	99.6	99.2	99.7	99.5	99.8	
StackMetric20	99	100	1.19k	99.5	100	99.1	99.4	99.4	
StackMetric21	97.5	100	1.19k	99.4	99.6	99.7	97.6	99.8	

Propiedades de la tabla

Una tabla de datos tiene un conjunto predeterminado de propiedades que no requieren cambios en las opciones o en el origen. Estas propiedades incluyen una columna de etiquetas fijas, todas las columnas de resumen activadas, los puntos de datos redondeados y las unidades convertidas.

Cada widget de tabla de datos puede tener las siguientes propiedades. La información sobre cada propiedad incluye cómo configurarla en el origen JSON del panel. Para obtener más información sobre el panel JSON, consulte Estructura y sintaxis del cuerpo del panel.

Resumen

Las columnas de resumen son una propiedad nueva que se introduce con el widget de tabla de datos. Estas columnas son un subconjunto específico de resúmenes de la tabla actual. Por ejemplo, el resumen Suma es una suma de todos los puntos de datos que se muestran en la fila. Las columnas de resumen no son las mismas que las de las estadísticas de CloudWatch. Se representan en el origen de la siguiente manera:

```
 "table": {
        "summaryColumns": [
            "MIN",
            "MAX",
            "SUM",
            "AVG"
        ]
    },
```

Umbrales

Lo puede utilizar para aplicar umbrales a la tabla. Cuando un punto de datos se encuentra dentro de un umbral, la celda se resalta con el color del umbral. Se representan en el origen de la siguiente manera:

```
"annotations": {
    "horizontal": [
        {
            "label": string,
            "value": int,
            "fill": "above" | "below"
        }
    ]
}
```

Unidad en la columna de etiquetas

Para mostrar qué unidad se asocia a la métrica, puede activar esta opción y así mostrar la unidad en la columna de etiquetas situada junto a la etiqueta. Se representan en el origen de la siguiente manera:

```
"yAxis": {
    "left": {
```

```
        "showUnits": true | false
    }
}
```

Invertir filas y columnas

Esta función transforma la tabla para que los puntos de datos cambien de columnas a filas y las métricas se conviertan en columnas. Se representan en el origen de la siguiente manera:

```
 "table": {
    "layout": "vertical" | "horizontal"
}
```

Columnas de resumen fijas

Esto hace que las columnas de resumen estén fijas para que permanezcan visibles mientras se desplaza por ellas. La etiqueta ya está fija. Se representan en el origen de la siguiente manera:

```
"table": {
    "stickySummary": true | false
}
```

Visualización solo de las columnas de resumen

Esto evita que se muestren las columnas de puntos de datos, de modo que solo se muestran las columnas de etiquetas y de resumen. Se representan en el origen de la siguiente manera:

```
 "table": {
    "showTimeSeriesData": false | true
}
```

Datos en directo

Muestra el punto de datos más reciente, incluso si aún no está completamente agregado. Se representan en el origen de la siguiente manera:

```
"liveData": true | false
```

Formato del widget numérico

Muestra todos los dígitos que pueden caber en la celda antes de redondearlos y convertirlos. Se representan en el origen de la siguiente manera:

```
"singleValueFullPrecision": true | false
```

Cómo añadir un widget de tabla de datos a un panel

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Paneles y, a continuación, elija un panel.
3. Haga clic en el botón +, seleccione Tabla de datos y elija Siguiente.
4. En la pestaña Examinar, busque o examine las métricas que desea mostrar en el widget de tabla. A continuación, seleccione las métricas.
5. (Opcional) Para cambiar el diseño de la tabla, seleccione la pestaña Opciones y seleccione Invertir filas y columnas.

   También puede utilizar la pestaña Opciones para cambiar las columnas que aparecen en la tabla y mostrar la unidad que se utiliza en la columna de Etiqueta.

   > **Tip**
   >
   > Para mostrar umbrales más precisos, seleccione Mostrar todos los dígitos que pueden caber antes de redondear.

6. (Opcional) Para cambiar el intervalo del widget de la tabla de datos, seleccione uno de los intervalos de tiempo predefinidos en la parte superior del widget. Los intervalos de tiempo oscilan entre 1 hora y 1 semana. Para configurar su propio intervalo de tiempo, elija Custom (Personalizado).
7. (Opcional) Para cambiar el intervalo del widget de la tabla de datos, seleccione uno de los intervalos de tiempo predefinidos en la parte superior del widget. Los intervalos de tiempo oscilan entre 1 hora y 1 semana. Para configurar su propio intervalo de tiempo, elija Custom (Personalizado).
8. (Opcional) Para que este widget siga utilizando el intervalo de tiempo que ha seleccionado, incluso si el intervalo de tiempo del resto del panel se modifica posteriormente, seleccione Persistir en el intervalo de tiempo.
9. Elija Crear un widget y, a continuación, elija Guardar panel.

Cómo eliminar un widget de tabla de un panel

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Dashboards (Paneles) y, a continuación, elija un panel.
3. En la esquina superior derecha del widget que desea eliminar, elija Acciones del widget, Eliminar.
4. Elija Save dashboard (Guardar panel).

Vincule y desvincule gráficos en un panel de CloudWatch

Puede vincular los gráficos de su panel conjuntamente, de modo que al ampliar o reducir la vista en un gráfico, el resto de gráficos se amplíen o reduzcan al mismo tiempo. Puedes desvincular gráficos para limitar la ampliación a un gráfico.

Para vincular los gráficos en un panel

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Dashboards (Paneles) y, a continuación, elija un panel.
3. Elija Actions (Acciones) y, a continuación, Link graphs (Vincular gráficos).

Para desvincular los gráficos en un panel

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Dashboards (Paneles) y, a continuación, elija un panel.
3. Desactive la opción Actions (Acciones) y, a continuación, Link graphs (Vincular gráficos).

Compartir paneles de CloudWatch

Puede compartir los paneles de CloudWatch con quienes no tienen acceso directo a su cuenta de AWS. Esto le permite compartir paneles entre equipos, con inversores y con personas externas a su organización. Incluso puede mostrar paneles en pantallas grandes en áreas de equipo o integrarlos en Wikis y otras páginas web.

> **Warning**
>
> A todas las personas con las que comparta el panel se les conceden los permisos enumerados en Permisos que se conceden a las personas con las que se comparte el panel para la cuenta. Si comparte el panel públicamente, todos aquellos que cuenten con el vínculo al panel tendrán estos permisos.
> Los permisos `cloudwatch:GetMetricData` y `ec2:DescribeTags` no se pueden limitar a métricas específicas o a instancias EC2, por lo que quienes cuenten con acceso al panel pueden consultar todas las métricas de CloudWatch y los nombres y etiquetas de todas las instancias EC2 de la cuenta.

Cuando comparte paneles, puede designar quién puede ver el panel de tres maneras:

- Compartir un único panel y designar hasta cinco direcciones de email de quienes pueden ver el panel. Cada uno de estos usuarios crea su propia contraseña que deben ingresar para ver el panel.
- Compartir un único panel de control de manera pública para que cualquiera que cuente con el enlace pueda verlo.
- Compartir todos los paneles de CloudWatch de su cuenta y especificar un proveedor de inicio de sesión único (SSO) de terceros para acceder al panel. Todos los usuarios que son miembros de la lista de proveedores de este SSO pueden acceder a todos los paneles de la cuenta. Para habilitarlo, integre el proveedor de SSO con Amazon Cognito. El proveedor de SSO debe admitir Security Assertion Markup Language (SAML) (Lenguaje de Marcado para Confirmaciones de Seguridad). Para obtener más información acerca de Amazon Cognito, consulte What is Amazon Cognito? (¿Qué es Amazon Cognito?)

Compartir un panel de control no conlleva cargos, pero los widgets dentro de un panel compartido sí se cobran según las tarifas estándar de CloudWatch. Para obtener más información sobre los precios de CloudWatch, consulte Precios de Amazon CloudWatch.

Al compartir un panel, los recursos de Amazon Cognito se crean en la región Este de EE. UU. (Norte de Virginia).

Important

No modifique los nombres ni los identificadores de los recursos creados por el proceso de uso compartido del panel. Esto incluye los recursos de Amazon Cognito e IAM. Modificar estos recursos puede provocar un funcionamiento inesperado e incorrecto de los paneles compartidos.

Note

Si comparte un panel que tiene widgets de métricas con anotaciones de alarmas, las personas con las que comparte el panel no verán esos widgets. En su lugar, verán un widget en blanco con texto que indica que el widget no está disponible. Podrá seguir viendo widgets de métricas con anotaciones de alarma cuando usted mismo vea el panel.

Permisos necesarios para compartir un panel

Para poder compartir paneles con cualquiera de los siguientes métodos y ver qué paneles ya se han compartido, debe iniciar sesión en un usuario de IAM o rol de IAM que tenga determinados permisos.

Para poder compartir paneles, su usuario o rol de IAM debe incluir los permisos incluidos en la siguiente declaración de políticas:

```
{
    "Effect": "Allow",
    "Action": [
        "iam:CreateRole",
        "iam:CreatePolicy",
        "iam:AttachRolePolicy",
        "iam:PassRole"
    ],
    "Resource": [
        "arn:aws:iam::*:role/service-role/CWDBSharing*",
        "arn:aws:iam::*:policy/*"
    ]
},
{
    "Effect": "Allow",
    "Action": [
```

```
        "cognito-idp:*",
        "cognito-identity:*",
    ],
    "Resource": [
        "*"
    ]
},
{
    "Effect": "Allow",
    "Action": [
        "cloudwatch:GetDashboard",
    ],
    "Resource": [
        "*"
        // or the ARNs of dashboards that you want to share
    ]
}
```

Para poder ver qué paneles se comparten, pero no compartir paneles, su usuario o rol de IAM puede incluir una declaración de políticas similar a la siguiente:

```
{
    "Effect": "Allow",
    "Action": [
        "cognito-idp:*",
        "cognito-identity:*"
    ],
    "Resource": [
        "*"
    ]
},
{
    "Effect": "Allow",
    "Action": [
        "cloudwatch:ListDashboards",
    ],
    "Resource": [
        "*"
    ]
}
```

Permisos que se conceden a las personas con las que se comparte el panel

Cuando se comparte un panel, CloudWatch crea un rol de IAM en la cuenta que otorga los siguientes permisos a las personas con las que se comparte el panel:

- `cloudwatch:GetInsightRuleReport`
- `cloudwatch:GetMetricData`
- `cloudwatch:DescribeAlarms`
- `ec2:DescribeTags`

> **Warning**
>
> Todas las personas con las que comparte el panel de control reciben estos permisos para la cuenta. Si comparte el panel públicamente, todas las personas que cuenten con el enlace al panel tendrán estos permisos.
> Los permisos `cloudwatch:GetMetricData` y `ec2:DescribeTags` no se pueden limitar a métricas específicas o a instancias EC2, por lo que quienes cuenten con acceso al panel pueden consultar todas las métricas de CloudWatch y los nombres y etiquetas de todas las instancias EC2 de la cuenta.

Cuando comparte un panel, de forma predeterminada, los permisos que crea CloudWatch restringen el acceso solo a las alarmas y a las reglas de Contributor Insights que se encuentran en el panel cuando se comparte. Si agrega nuevas alarmas o reglas de Contributor Insights al panel y desea que también las vean las personas con las que compartió el panel, debe actualizar la política para permitir estos recursos.

Comparta un único panel con usuarios específicos

Siga los pasos de esta sección para compartir un panel con hasta cinco direcciones de email que usted elija.

> **Note**
>
> De forma predeterminada, los widgets de CloudWatch Logs del panel no son visibles para las personas con las que comparte el panel. Para obtener más información, consulte Se concede

permiso a las personas con las que se comparten los paneles para que vean los widgets de las tablas de registros.
De forma predeterminada, los widgets de alarma compuestos del panel no son visibles para las personas con las que comparte el panel. Para obtener más información, consulte Permitir ver alarmas compuestas a las personas con las que comparte .

Para compartir un panel con usuarios específicos

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Dashboards (Paneles).
3. Elija el nombre del panel.
4. Seleccione Actions (Acciones), luego Share dashboard (Compartir panel).
5. Junto a Share your dashboard and require a username and password (Compartir panel y solicitar un nombre de usuario y una contraseña), elija Start sharing (Comenzar a compartir).
6. En Add email addresses (Agregar direcciones de email), ingrese las direcciones de email con las que desea compartir el panel. Puede incluir hasta cinco direcciones de email.
7. Cuando haya ingresado todas las direcciones de email, lea el acuerdo y seleccione la casilla de confirmación. A continuación, elija Review policy (Revisar política).
8. Confirme que los recursos que se compartirán son los que desea y elija Confirm and generate shareable link (Confirmar y generar enlaces compartibles).
9. En la siguiente página, elija Copy link to clipboard (Copiar enlace al portapapeles). A continuación, puede pegar este enlace en el email y enviarlo a los usuarios invitados. Los usuarios reciben automáticamente un email por separado con el nombre de usuario y una contraseña temporal para conectarse al panel.

Comparta un único panel de manera pública

Siga los pasos de esta sección para compartir un panel públicamente. Puede ser útil para mostrar el panel en una pantalla grande en una sala de equipo o integrarlo en una página Wiki.

> **Important**
>
> Compartir un panel de manera pública hace que sea accesible para cualquier persona que tenga el enlace, sin autenticación. Realícelo sólo con paneles que no contienen información confidencial.

> **Note**
>
> De forma predeterminada, los widgets de CloudWatch Logs del panel no son visibles para las personas con las que comparte el panel. Para obtener más información, consulte Se concede permiso a las personas con las que se comparten los paneles para que vean los widgets de las tablas de registros.
> De forma predeterminada, los widgets de alarma compuestos del panel no son visibles para las personas con las que comparte el panel. Para obtener más información, consulte Permitir ver alarmas compuestas a las personas con las que comparte .

Para compartir un panel de forma pública

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Dashboards (Paneles).
3. Elija el nombre del panel.
4. Seleccione Actions (Acciones), luego Share dashboard (Compartir panel).
5. Junto a Share your dashboard publicly (Compartir panel públicamente), elija Start sharing (Comenzar a compartir).
6. Escriba **`Confirm`** en el cuadro de texto.
7. Lea el acuerdo y seleccione la casilla de confirmación. A continuación, elija Review policy (Revisar política).
8. Confirme que los recursos que se compartirán son los que desea y elija Confirm and generate shareable link (Confirmar y generar enlaces compartibles).
9. En la siguiente página, elija Copy link to clipboard (Copiar enlace al portapapeles). A continuación, puede compartir este enlace. Cualquier persona con la que comparta el vínculo puede acceder al panel, sin proporcionar credenciales.

Comparta todos los paneles de CloudWatch en la cuenta mediante SSO

Siga los pasos de esta sección para compartir todos los paneles de su cuenta con los usuarios mediante el inicio de sesión único (SSO).

> **Note**
>
> De forma predeterminada, los widgets de CloudWatch Logs del panel no son visibles para las personas con las que comparte el panel. Para obtener más información, consulte Se concede permiso a las personas con las que se comparten los paneles para que vean los widgets de las tablas de registros.
> De forma predeterminada, los widgets de alarma compuestos del panel no son visibles para las personas con las que comparte el panel. Para obtener más información, consulte Permitir ver alarmas compuestas a las personas con las que comparte .

Para compartir los paneles de CloudWatch con los usuarios que se encuentran en la lista de proveedores de SSO

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Dashboards (Paneles).
3. Elija el nombre del panel.
4. Seleccionar Actions (Acciones), Share dashboard (Compartir panel).
5. Seleccione Go to CloudWatch Settings (Ir a la configuración de CloudWatch).
6. Si el proveedor SSO que desea no aparece en Available SSO providers (Proveedores SSO disponibles), elija Manage SSO providers (Administrar proveedores SSO) y siga las instrucciones en Configure SSO para compartir el panel de CloudWatch.

   A continuación, vuelva a la consola de CloudWatch y actualice el navegador. Ahora debería aparecer en la lista el proveedor SSO que ha habilitado
7. Elija el proveedor SSO que desee en la lista Available SSO providers (Proveedores SSO disponibles).
8. Elija Guardar cambios.

Configure SSO para compartir el panel de CloudWatch

Para configurar el uso compartido del panel a través de un proveedor de inicio de sesión único de terceros que admita SAML, siga estos pasos.

> **Important**
>
> Se recomienda que no comparta paneles mediante un proveedor SSO que no sea SAML. Al hacerlo, se corre el riesgo de permitir que terceros accedan inadvertidamente a los paneles de su cuenta.

Para configurar un proveedor SSO para habilitar el uso compartido del panel

1. Integre el proveedor SSO con Amazon Cognito. Para obtener más información, consulte Integrating Third-Party SAML Identity Providers with Amazon Cognito User Pools (Integración de proveedores de identidad SAML de terceros con Grupos de usuarios de Amazon Cognito).
2. Descargue el archivo XML de metadatos del proveedor SSO.
3. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
4. En el panel de navegación, seleccione Configuración.
5. En la sección Dashboard sharing (Uso compartido de paneles), elija Configure (Configurar).
6. Seleccione Manage SSO providers (Administrar proveedores SSO).

   De esta forma, abre la consola de Amazon Cognito en la región Este de EE. UU. (Norte de Virginia) (us-east-1). Si no ve User Pools (Grupos de usuarios), es posible que la consola de Amazon Cognito se haya abierto en una Región diferente. Si es así, cambie la Región a EE. UU. Este (Norte de Virginia) us-east-1 y continúe con los siguientes pasos.
7. Elija el grupo CloudWatchDashboardSharing.
8. En el panel de navegación, elija Proveedores de identidades.
9. Elija SAML.
10. Ingrese un nombre para el proveedor SSO en Provider name (Nombre del proveedor).
11. Elija Select file (Seleccionar archivo) y seleccione el archivo XML de metadatos que descargó en el paso 1.
12. Elija Create provider (Crear proveedor).

Vea cuántos de sus paneles se comparten

Puede utilizar la consola de CloudWatch para ver cuántos de sus paneles de CloudWatch se están compartiendo actualmente con otros usuarios.

Para ver cuántos de sus paneles se están compartiendo

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, seleccione Configuración.
3. La sección Dashboard sharing (Uso compartido de paneles) muestra cuántos paneles se comparten.
4. Para ver qué paneles se comparten, elija ***number*** dashboards shared (Número de paneles compartidos) en Username and password (Nombre de usuario y contraseña) y en Public dashboards (Paneles públicos).

Vea qué paneles se comparten

Puede usar la consola de CloudWatch para ver qué paneles se están compartiendo actualmente con otros usuarios.

Para ver qué paneles se están compartiendo

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Dashboards (Paneles).
3. En la lista de paneles, consulte la columna Share (Compartir). Los paneles que tienen rellenado el icono de esta columna se están compartiendo actualmente.
4. Para ver con qué usuarios se comparte un panel, elija el nombre del panel y, a continuación, elija Actions (Acciones), Share dashboards (Compartir panel).

   La página Share dashboard ***dashboard name*** (Compartir panel [nombre del panel]) muestra cómo se comparte el panel. Si lo desea, puede dejar de compartir el panel al seleccionar Stop sharing (Dejar de compartir).

Detener el uso compartido de uno o varios paneles

Puede dejar de compartir un solo panel o dejar de compartir todos los paneles a la vez.

Para detener el uso compartido de un panel

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Dashboards (Paneles).
3. Elija el nombre del panel compartido.
4. Seleccione Actions (Acciones), Share dashboard (Compartir panel).
5. Seleccione Stop sharing (Dejar de compartir).
6. En el cuadro de confirmación, elija Stop sharing (Dejar de compartir).

Para detener el uso compartido de todos los paneles

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, seleccione Configuración.
3. En la sección Dashboard sharing (Uso compartido de paneles), elija Stop sharing all dashboards (Dejar de compartir todos los paneles).
4. En el cuadro de confirmación, elija Stop sharing all dashboards (Dejar de compartir todos los paneles).

Revise los permisos de los paneles compartidos y cambie el alcance de permisos

Siga los pasos descritos en esta sección si desea revisar los permisos de los usuarios de los paneles compartidos o cambiar el alcance de los permisos de los paneles compartidos.

Para revisar los permisos de los paneles compartidos

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Dashboards (Paneles).
3. Elija el nombre del panel compartido.
4. Elija Actions (Acciones) y, luego, Share dashboard (Compartir panel).
5. En Resources (Recursos), elija Rol de IAM.
6. En la consola de IAM, elija la política que se muestra.
7. (Opcional) Para limitar las alarmas que pueden ver los usuarios del panel compartido, elija Edit policy (Editar política) y mueva el permiso `cloudwatch:DescribeAlarms` desde su posición

actual a una declaración nueva de `Allow` que enumera los ARN de sólo las alarmas que desea que los usuarios del panel compartido vean. Consulte el siguiente ejemplo.

```
{
   "Effect": "Allow",
    "Action": "cloudwatch:DescribeAlarms",
    "Resource": [
        "AlarmARN1",
        "AlarmARN2"
    ]
}
```

Si hace esto, asegúrese de eliminar el permiso `cloudwatch:DescribeAlarms` desde una sección de la política actual que se ve así:

```
{
   "Effect": "Allow",
    "Action": [
        "cloudwatch:GetInsightRuleReport",
        "cloudwatch:GetMetricData",
        "cloudwatch:DescribeAlarms",
        "ec2:DescribeTags"
    ],
    "Resource": "*"
}
```

8. (Opcional) Para limitar el alcance de las reglas de Contributor Insights que los usuarios del panel compartido pueden ver, elija Edit policy (Editar política) y mueva `cloudwatch:GetInsightRuleReport` desde su posición actual a una declaración nueva de `Allow` que enumera los ARN de sólo las reglas de Contributor Insights que desea que los usuarios del panel compartido vean. Consulte el siguiente ejemplo.

```
{
   "Effect": "Allow",
    "Action": "cloudwatch:GetInsightRuleReport",
    "Resource": [
        "PublicContributorInsightsRuleARN1",
        "PublicContributorInsightsRuleARN2"
    ]
}
```

Si lo hace, asegúrese de eliminar `cloudwatch:GetInsightRuleReport` de una sección de la política actual que tiene un aspecto similar al siguiente:

```
{
            "Effect": "Allow",
            "Action": [
                "cloudwatch:GetInsightRuleReport",
                "cloudwatch:GetMetricData",
                "cloudwatch:DescribeAlarms",
                "ec2:DescribeTags"
            ],
            "Resource": "*"
        }
```

Permitir ver alarmas compuestas a las personas con las que comparte

Cuando comparte un panel, de forma predeterminada, los widgets de alarmas compuestas del panel no son visibles para las personas con las que comparte el panel. Para que los widgets de alarmas compuestas sean visibles, debe agregar un permiso `DescribeAlarms: *` a la política de uso compartido del panel. Ese permiso tendría el siguiente aspecto:

```
{
    "Effect": "Allow",
    "Action": "cloudwatch:DescribeAlarms",
    "Resource": "*"
}
```

Warning

La declaración de política anterior da acceso a todas las alarmas de la cuenta. Para reducir el alcance de `cloudwatch:DescribeAlarms`, debe utilizar una declaración de `Deny`. Puede añadir una declaración de `Deny` a la política y especificar los ARN de las alarmas que desea bloquear. Esa declaración de denegación debería tener un aspecto similar al siguiente:

```
{
    "Effect": "Allow",
    "Action": "cloudwatch:DescribeAlarms",
```

```
    "Resource": "*"
},
{
    "Effect": "Deny",
    "Action": "cloudwatch:DescribeAlarms",
    "Resource": [
        "SensitiveAlarm1ARN",
        "SensitiveAlarm1ARN"
    ]
}
```

Se concede permiso a las personas con las que se comparten los paneles para que vean los widgets de las tablas de registros

Cuando comparte un panel, de forma predeterminada los widgets de CloudWatch Logs Insights que se encuentran en el panel no son visibles para las personas con las que comparte el panel. Esto afecta tanto a los widgets de CloudWatch Logs Insights existentes como a los que se agregan al panel después de compartirlo.

Si desea que puedan ver los widgets de CloudWatch Logs, debe agregar permisos al rol de IAM para el uso compartido de paneles.

Para permitir que las personas con las que comparte un panel vean los widgets de CloudWatch Logs

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Dashboards (Paneles).
3. Elija el nombre del panel compartido.
4. Elija Actions (Acciones) y, luego, Share dashboard (Compartir panel).
5. En Resources (Recursos), elija Rol de IAM.
6. En la consola de IAM, elija la política que se muestra.
7. Seleccione Edit policy (Editar política) y agregue la siguiente declaración. En la nueva declaración, se recomienda que especifique los ARN de sólo los grupos de registro que desea compartir. Consulte el siguiente ejemplo.

```
{
            "Effect": "Allow",
```

```
            "Action": [
                "logs:FilterLogEvents",
                "logs:StartQuery",
                "logs:StopQuery",
                "logs:GetLogRecord",
                "logs:DescribeLogGroups"
            ],
            "Resource": [
                "SharedLogGroup1ARN",
                "SharedLogGroup2ARN"
           ]
        },
```

8. Elija Save changes (Guardar cambios).

Si la política de IAM para el uso compartido de paneles ya incluye esos cinco permisos con * como el recurso, se recomienda encarecidamente que cambie la política y especifique solo los ARN de los grupos de registro que desea compartir. Por ejemplo, si la sección `Resource` para estos permisos fuera la siguiente:

```
"Resource": "*"
```

Cambie la política para especificar sólo los ARN de los grupos de registros que desea compartir, como en el siguiente ejemplo:

```
"Resource": [
  "SharedLogGroup1ARN",
  "SharedLogGroup2ARN"
]
```

Se concede permiso a las personas con las que comparte para que vean los widgets personalizados

Cuando comparte un panel, de forma predeterminada los widgets personalizados que están en el panel no son visibles para las personas con las que comparte el panel. Esto afecta tanto a los widgets personalizados existentes como a los que se agregan al panel después de compartirlo.

Si desea que puedan ver los widgets personalizados, agregue permisos al rol de IAM para compartir el panel.

Para permitir que las personas con las que comparte un panel vean los widgets personalizados

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Dashboards (Paneles).
3. Elija el nombre del panel compartido.
4. Elija Actions (Acciones) y, luego, Share dashboard (Compartir panel).
5. En Resources (Recursos), elija Rol de IAM.
6. En la consola de IAM, elija la política que se muestra.
7. Seleccione Edit policy (Editar política) y agregue la siguiente declaración. En la nueva declaración, se recomienda que especifique los ARN de sólo las funciones de Lambda que desea compartir. Consulte el siguiente ejemplo.

```
{
  "Sid": "Invoke",
  "Effect": "Allow",
  "Action": [
      "lambda:InvokeFunction"
  ],
  "Resource": [
    "LambdaFunction1ARN",
    "LambdaFunction2ARN"
  ]
 }
```

8. Elija Save changes (Guardar cambios).

Si la política de IAM para compartir el panel ya incluye ese permiso con * como recurso, se le recomienda encarecidamente que cambie la política y especifique sólo los ARN de las funciones de Lambda que desea compartir. Por ejemplo, si la sección `Resource` para estos permisos fuera la siguiente:

```
"Resource": "*"
```

Cambie la política para especificar sólo los ARN de los widgets personalizados que desea compartir, como en el ejemplo siguiente:

```
"Resource": [
  "LambdaFunction1ARN",
```

```
  "LambdaFunction2ARN"
]
```

Utilizar datos en directo

Puede elegir si los widgets de métricas muestran datos en directo. Los datos en directo son aquellos publicados en el último minuto que no se han agregado por completo.

- Si dichos datos están desactivados, solo se muestran los puntos de datos con un período de agregación de al menos un minuto en el pasado. Por ejemplo, cuando se utilizan períodos de cinco minutos, el punto de datos para las 12:35 se agregaría de 12:35 a 12:40 y se mostraría a las 12:41.
- Si los datos en directo están activados, se muestra el punto de datos más reciente tan pronto como se publiquen datos en el intervalo de agregación correspondiente. El punto de datos más reciente puede cambiar cada vez que actualice la pantalla según se publiquen nuevos datos en ese período de agregación. Si utiliza una estadística acumulativa como Sum (Suma) o Sample count (Recuento de muestras), el uso de datos en directo puede provocar un descenso al final del gráfico.

Tiene la opción de habilitar datos en directo para un panel completo o para widgets individuales en el panel.

Para elegir si se van a utilizar datos en directo en todo el panel

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Dashboards (Paneles) y, a continuación, elija un panel.
3. Para activar o desactivar de forma permanente los datos en directo de todos los widgets del panel, realice lo siguiente:
   a. Elija Actions (Acciones), Settings (Configuración), Bulk update live data (Actualización por lotes de datos en directo).
   b. Elija Live Data on (Datos en directo activados) o Live Data off (Datos en directo desactivados) y, a continuación, Set (Establecer).
4. Para anular temporalmente la configuración de datos en directo de cada widget, elija Actions (Acciones). A continuación, en Overrides (Anulaciones), junto a Live data (Datos en directo), realice una de las siguientes operaciones:

- Elija On (Activar) para activar temporalmente los datos en directo de todos los widgets.
- Elija Off (Desactivar) para desactivar temporalmente los datos en directo de todos los widgets.
- Elija Do not override (No invalidar) para conservar la configuración de datos en directo de cada widget.

Para elegir si se van a utilizar datos en directo en un único widget

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Dashboards (Paneles) y, a continuación, elija un panel.
3. Seleccione un widget y elija Actions (Acciones), Edit (Editar).
4. Elija la pestaña Graph options (Opciones del gráfico).
5. Seleccione o borre la casilla en Live Data (Datos en directo).

Visualización de un panel animado

Puede ver un panel animado que repite los datos métricos de CloudWatch que se capturaron con el paso del tiempo. Esto puede ayudarle a ver tendencias, a hacer presentaciones o a analizar problemas después de que se produzcan.

Los widgets animados del panel incluyen widgets de líneas, de área apilada, de números y widgets de explorador de métricas. Los gráficos circulares, los gráficos de barras, los widgets de texto y los widgets de registros se muestran en el panel, pero no son animados.

Para ver un panel animado

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Dashboards (Paneles).
3. Elija el nombre del panel.
4. Seleccione Actions (Acciones), Replay dashboard (Repetir panel).
5. (Opcional) De forma predeterminada, al comenzar la animación, aparece como una ventana deslizante. Si desea que la animación aparezca como una animación punto por punto, elija el icono de lupa mientras la animación está en pausa y restablezca el zoom.
6. Para comenzar la animación, elija el botón Reproducir. También puede elegir los botones atrás y adelante para desplazarse a otros puntos en el tiempo.

7. (Opcional) Para cambiar la ventana de tiempo de la animación, elija el calendario y seleccione el período de tiempo.
8. Para cambiar la velocidad de la animación, elija Auto speed (Velocidad automática) y seleccione la nueva velocidad.
9. Cuando haya terminado, seleccione Exit animate (Salir de la animación).

Agregue un panel de CloudWatch a la lista de favoritos

En la consola de CloudWatch, puede agregar paneles, alarmas y grupos de registros a una lista de favoritos. Puede acceder a la lista de favoritos desde el panel de navegación. En el siguiente procedimiento se describe cómo agregar un panel a la lista de favoritos.

Para agregar un panel a la lista de favoritos

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Dashboards (Paneles).
3. En la lista de paneles, seleccione el símbolo de estrella situado junto al nombre del panel que desea seleccionar como favorito.
   - (Opcional) También puede seleccionar un panel de la lista y elegir el símbolo de estrella junto al nombre del panel para seleccionarlo como favorito.
4. Para acceder a la lista de favoritos, elija Favorites and recents (Favoritos y recientes) en el panel de navegación. El menú contiene dos columnas. Una contiene los paneles, alarmas y grupos de registros favoritos, y la otra columna contiene los paneles, alarmas y grupos de registros que visitó recientemente.

> **Tip**
>
> Puede seleccionar paneles como favoritos, así como alarmas y grupos de registros, desde el menú Favorites and recents (Favoritos y recientes) del panel de navegación de la consola de CloudWatch. En la columna Recently visited (Visitados recientemente), coloque el cursor sobre el panel que desea marcar como favorito y elija el símbolo de estrella junto a él.

Cambie la configuración de anulación del periodo o del intervalo de actualización para el panel de CloudWatch

Puede especificar cómo se conserva o modifica la configuración del periodo de los gráficos que se añaden a este panel.

Cuando se aplica un periodo automático o un intervalo de tiempo persistente a un widget, el intervalo de tiempo general del gráfico puede afectar a los periodos que haya establecido.

- Si el intervalo de tiempo es de un día o menos, la configuración del periodo no cambia.
- Si el intervalo de tiempo está comprendido entre uno y tres días, los periodos establecidos por debajo de los cinco minutos se cambian a 5 minutos.
- Si el intervalo de tiempo es superior a tres días, los periodos establecidos por debajo de una hora se cambian a una hora.

En los siguientes pasos se explica cómo usar la consola para cambiar las opciones de anulación del período. También puede cambiarlas mediante el campo `period0verride` en la estructura JSON del panel. Para obtener más información, consulte Estructura general del cuerpo del panel.

Para cambiar las opciones de anulación del periodo

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. Elija Acciones.
3. En Period override (Anulación del periodo), elija una de las siguientes opciones:
   - Elija Auto para que el periodo de las métricas de cada gráfico se adapte automáticamente al intervalo de tiempo del panel.
   - Elija Do not override (No anular) para garantizar que la configuración del periodo de cada gráfico se aplica siempre.
   - Elija una de las otras opciones para que los gráficos que se añadan al panel siempre tomen esa hora elegida como configuración del periodo.

   La opción Period override (Anulación del periodo) siempre vuelve a Auto cuando se cierra el panel o se actualiza el navegador. No es posible guardar diferentes configuraciones para Period override (Anulación del periodo).

Puede cambiar la frecuencia con la que se actualizan los datos del panel de CloudWatch.

Para cambiar el intervalo de actualización del panel

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Dashboards (Paneles) y, a continuación, elija un panel.
3. En el menú Refresh options (Opciones de actualización) (esquina superior derecha), elija 10 Seconds (10 segundos), 1 Minute (1 minuto), 2 Minutes (2 minutos), 5 Minutes (5 minutos) o 15 Minutes (15 minutos).

Cambie el intervalo de tiempo o el formato de zona horaria en un panel de CloudWatch

Puede cambiar el intervalo de tiempo para mostrar los datos del panel en minutos, horas, días o semanas. También puede cambiar el formato de la zona horaria para mostrar los datos del panel en UTC o en la hora local. La hora local es la zona horaria que se especifica en el sistema operativo de la computadora.

> **Note**
>
> Si crea un panel con gráficos que contienen 100 métricas de alta resolución o más, le recomendamos no establecer el intervalo de tiempo en un valor superior a una hora. Para obtener más información, consulte Métricas de alta resolución.

> **Note**
>
> Si el intervalo de tiempo de un panel es más corto que el periodo utilizado para un widget en el panel, ocurre lo siguiente:
>
> - El widget se modifica para mostrar la cantidad de datos correspondiente a un periodo completo para ese widget, aunque sea más largo que el intervalo de tiempo del panel. Esto garantiza que haya al menos un punto de datos en el gráfico.
> - La hora de inicio del periodo de este punto de datos se ajusta hacia atrás para garantizar que se pueda mostrar al menos un punto de datos.

New console

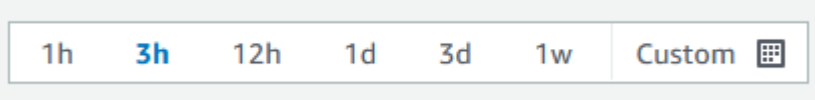
Para cambiar el intervalo de tiempo del panel

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Dashboards (Paneles) y, a continuación, elija un panel.
3. Desde la pantalla del panel, realice una de las siguientes operaciones:
   - En el área superior del panel, seleccione uno de los rangos de tiempo predefinidos. Estos abarcan de 1 hora a 1 semana (1 h, 3 h, 12 h, 1 día o 1 sem.).
   - Como alternativa, puede elegir una de las siguientes opciones de intervalos de tiempo personalizados:
     - Elija Custom (Personalizado) y, luego, seleccione la pestaña Relative (Relativo). Elija un intervalo de tiempo de 1 minuto a 15 meses.
     - Elija Custom (Personalizado) y, a continuación, seleccione la pestaña Absolute (Absoluto). Utilice el calendario o los campos de texto para especificar el intervalo de tiempo.

> **Tip**
>
> Si el periodo de agregación está establecido en Auto (Automático), cuando cambie el intervalo de tiempo de un gráfico, CloudWatch podrá cambiar el periodo. Para configurar el periodo de forma manual, elija el menú desplegable Actions (Acciones) y, a continuación, elija Period override (Anulación del periodo).

Para cambiar el formato de la zona horaria del panel

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Dashboards (Paneles) y, a continuación, elija un panel.
3. En la esquina superior derecha de la pantalla del panel, seleccione Personalizado.



4. En la esquina superior derecha del cuadro que aparezca, elija UTC (Tiempo universal coordinado) o Local time (Hora local) en el menú desplegable.

5. Seleccione Aplicar.

Old console

Para cambiar el intervalo de tiempo del panel

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Dashboards (Paneles) y, a continuación, elija un panel.
3. Desde la pantalla del panel, realice una de las siguientes operaciones:
   - En el área superior del panel, seleccione uno de los rangos de tiempo predefinidos. Estos abarcan de 1 hora a 1 semana (1 h, 3 h, 12 h, 1 día, 3 días o 1 sem.).
   - Como alternativa, puede elegir una de las siguientes opciones de intervalos de tiempo personalizados:
     - Elija el menú desplegable Custom (Personalizado) y, luego, seleccione la pestaña Relative (Relativo). Seleccione uno de los intervalos predefinidos, que abarcan desde 1 minuto hasta 15 meses.
     - Elija el menú desplegable Custom (Personalizado) y, luego, seleccione la pestaña Absolute (Absoluto). Utilice el calendario o los campos de texto para especificar el intervalo de tiempo.

> **Tip**
>
> Si el periodo de agregación está establecido en Auto (Automático), cuando cambie el intervalo de tiempo de un gráfico, CloudWatch podrá cambiar el periodo. Para configurar el periodo de forma manual, elija el menú desplegable Actions (Acciones) y, a continuación, elija Period override (Anulación del periodo).

Para cambiar el formato de la zona horaria del panel

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Dashboards (Paneles) y, a continuación, elija un panel.
3. En la esquina superior derecha de la pantalla del panel, elija el menú desplegable Custom (Personalizado).

4. En la esquina superior derecha del cuadro que aparezca, seleccione UTC (Tiempo universal coordinado) o Local timezone (Zona horaria local) en el menú desplegable.

Uso de métricas de Amazon CloudWatch

Las métricas son los datos sobre el desempeño de los sistemas. De forma predeterminada, diversos servicios ofrecen métricas gratuitas para recursos (tales como las instancias de Amazon EC2, los volúmenes de Amazon EBS y las instancias de base de datos de Amazon RDS). También puede habilitar la supervisión detallada para algunos recursos, como las instancias de Amazon EC2, o publicar las métricas de las aplicaciones que usted utiliza. Amazon CloudWatch puede cargar todas las métricas en su cuenta (tanto las métricas de los recursos AWS como las métricas de las aplicaciones que proporcione) para búsquedas, representación de gráficos y para alarmas.

Los datos de las métricas se guardan durante 15 meses, lo que le permite ver datos actualizados y datos históricos.

Para representar métricas en la consola, puede utilizar CloudWatch Metrics Insights, un motor de consultas SQL de alto rendimiento que puede utilizar para identificar tendencias y patrones dentro de todas sus métricas en tiempo real.

Contenido

Supervisión básica y supervisión detallada

CloudWatch proporciona dos categorías de supervisión: supervisión básica y supervisión detallada.

Muchos servicios AWS ofrecen supervisión básica publicando un conjunto predeterminado de métricas en CloudWatch sin cargo para los clientes. De forma predeterminada, cuando empiezas a utilizar uno de estos Servicios de AWS, se habilita automáticamente la supervisión básica. Para obtener una lista de los servicios que ofrecen supervisión básica, consulte Servicios de AWS que publican métricas de CloudWatch.

Solo algunos servicios ofrecen una supervisión detallada. También incurre en cargos. Para utilizarlo para un servicio AWS, debe elegir activarlo. Para obtener más información acerca de los precios, consulte Precios de Amazon CloudWatch.

Las opciones de supervisión detalladas difieren según los servicios que lo ofrecen. Por ejemplo, la supervisión detallada de Amazon EC2 proporciona métricas más frecuentes, publicadas a intervalos de un minuto, en lugar de los intervalos de cinco minutos utilizados en la supervisión básica de Amazon EC2. La supervisión detallada de Simple Storage Service (Amazon S3) y Amazon Managed Streaming para Apache Kafka significa métricas más detalladas.

En diferentes servicios AWS, la supervisión detallada también tiene nombres diferentes. Por ejemplo, en Amazon EC2 se denomina supervisión detallada, en AWS Elastic Beanstalk se denomina supervisión mejorada y en Simple Storage Service (Amazon S3) se denomina métricas de solicitudes.

El uso de la supervisión detallada de Amazon EC2 le ayuda a administrar mejor los recursos de Amazon EC2, de modo que pueda encontrar tendencias y actuar con mayor rapidez. Para Simple Storage Service (Amazon S3), las métricas de solicitudes están disponibles a intervalos de un minuto para ayudarle a identificar rápidamente los problemas operativos y actuar en consecuencia. En Amazon MSK, cuando habilita el nivel de supervisión `PER_BROKER`, `PER_TOPIC_PER_BROKER` o bien `PER_TOPIC_PER_PARTITION`, obtiene métricas adicionales que proporcionan más visibilidad.

En la siguiente tabla se muestra una lista de los servicios que ofrecen una supervisión detallada. También incluye enlaces a los documentos de los servicios que explican más sobre la supervisión detallada y proporcionan instrucciones sobre cómo activarlo.

Servicio	Documentación
Amazon API Gateway	Dimensiones de las métricas de API Gateway

Servicio	Documentación
Amazon CloudFront	Visualización de métricas de distribución adicionales de CloudFront
Amazon EC2	Activar o desactivar la supervisión detallada para las instancias
Elastic Beanstalk	Informes y supervisión de estado mejorados
Amazon Kinesis Data Streams	Métricas ampliadas de nivel de fragmento
Amazon MSK	Supervisión de Amazon MSK con las métricas de Amazon CloudWatch
Amazon S3	Métricas de solicitud de Simple Storage Service (Amazon S3) en CloudWatch

Servicio	Documentación
Amazon SES	Recopile métricas de supervisión detalladas de CloudWatch mediante la publicación de eventos de Amazon SES.

Además, CloudWatch ofrece soluciones de supervisión listas para usar con métricas más detalladas y paneles creados previamente para algunos servicios de AWS, como se muestra en la siguiente tabla.

Servicio	Documentación de característica
Lambda	Lambda Insights
Amazon ECS	Container Insights para Amazon ECS
Amazon EKS	Container Insights para Amazon EKS y Kubernetes

Consulte sus métricas con CloudWatch Metrics Insights

CloudWatch Metrics Insights es un potente motor de consultas SQL de alto rendimiento que puede utilizar para consultar sus métricas a escala. Puede identificar tendencias y patrones dentro de todas sus métricas de CloudWatch, en tiempo real.

También puede configurar alarmas en cualquier consulta de Información de métricas que devuelva una sola serie temporal. Esto puede resultar especialmente útil para crear alarmas que controlen las métricas agregadas de una flota de su infraestructura o aplicaciones. Cree la alarma una vez y verá que se ajusta dinámicamente a medida que se agreguen o se eliminen recursos de la flota.

Puede realizar una consulta de Información de métricas de CloudWatch en la consola con el editor de consultas de la Información de métricas de CloudWatch. También puede realizar una consulta de Información de métricas de CloudWatch con la AWS CLI o SDK de AWS al ejecutar `GetMetricData` o `PutDashboard`. Las consultas que ejecute con el editor de consultas de Información de métricas de CloudWatch son gratuitas. Para obtener más información sobre los precios de CloudWatch, consulte Precios de Amazon CloudWatch.

Con el editor de consultas de Información de métricas de CloudWatch, puede elegir entre una variedad de consultas de ejemplo prediseñadas y también crear sus propias consultas. Al crear las consultas, puede utilizar una vista de generador para examinar las métricas y dimensiones existentes. También puede usar una vista de editor para escribir las consultas de forma manual.

También puede utilizar lenguaje natural para crear consultas de Información de métricas de CloudWatch. Para ello, pregunte o describa los datos que busca. Esta función asistida por IA genera una consulta según una petición y proporciona una explicación línea por línea sobre cómo funciona la consulta. Para obtener más información, consulte Usar lenguaje natural para generar y actualizar consultas de Información de métricas de CloudWatch.

Puede ejecutar consultas a escala con Información de métricas. Con la cláusula AGRUPAR POR, puede agrupar sus métricas en tiempo real, en series temporales separadas por valor de dimensión específico. Como las consultas de Información de métricas incluyen la función ORDENAR POR, puede utilizar Información de métricas para realizar consultas del tipo “N principales”. Por ejemplo, las consultas del tipo “N principales” pueden analizar millones de métricas de su cuenta y arrojar las 10 instancias que consumen más CPU. Esto puede ayudarlo a identificar y solucionar los problemas de latencia en sus aplicaciones.

Temas

- Inferencia en SQL
- Consultas de ejemplo de Metrics Insights
- Límites de Metrics Insights
- Glosario de Metrics Insights
- Solución de problemas de Metrics Insights

Creación de sus consultas

Puede ejecutar una consulta de CloudWatch Metrics Insights mediante la consola de CloudWatch, AWS CLI o SDK de AWS. Las consultas que se realizan en la consola son gratuitas. Para obtener más información sobre los precios de CloudWatch, consulte Precios de Amazon CloudWatch.

Para obtener más información acerca del uso de SDK de AWS para realizar una consulta de Metrics Insights, consulte GetMetricData.

Para ejecutar una consulta mediante la consola de CloudWatch, siga estos pasos:

Para consultar las métricas mediante Metrics Insights

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, seleccione Métricas y, a continuación, Todas las métricas.
3. Elija la pestaña Queries (Consultas).
4. (Opcional) Para ejecutar una consulta de ejemplo prediseñada, elija Add query (Agregar consulta) y seleccione la consulta que se va a ejecutar. Si está satisfecho con esta consulta, puede omitir el resto del procedimiento. De lo contrario, puede elegir Editor para editar la consulta de ejemplo y luego, Run (Ejecutar) para ejecutar la consulta modificada.
5. Para crear su propia consulta, puede utilizar la vista de Builder (Generador), la vista de Editor y, también, una combinación de ambas. Puede cambiar entre las dos vistas en cualquier momento y ver su trabajo en curso en ambas vistas.

   En la vista Builder (Generador), puede examinar y seleccionar el espacio de nombre de métrica, el nombre de métrica, el filtro, el grupo y las opciones de pedido. Para cada una de estas opciones, el generador de consultas le ofrece una lista de posibles selecciones de su entorno de donde puede elegir.

   En la vista Editor, puede comenzar a escribir su consulta. A medida que escribe, el editor ofrece sugerencias basadas en los caracteres que ha escrito hasta el momento.

6. Cuando esté satisfecho con la consulta, elija Save (Guardar).
7. (Opcional) Otra forma de editar una consulta que haya representado es seleccionar la pestaña Graphed metrics (Representación gráfica de métricas) y elegir el icono de edición junto a la fórmula de consulta en la columna Details (Detalles).
8. (Opcional) Para quitar una consulta del gráfico, elija Graphed metrics (Representación gráfica de métricas) y, luego, el ícono de X en la parte derecha de la fila en la que se muestra la consulta.

Componentes de consulta y sintaxis de Metrics Insights

La sintaxis de CloudWatch Metrics Insights se detalla a continuación.

```
SELECT FUNCTION(metricName)
FROM namespace | SCHEMA(...)
[ WHERE labelKey OPERATOR labelValue [AND ... ] ]
[ GROUP BY labelKey [ , ... ] ]
[ ORDER BY FUNCTION() [ DESC | ASC ] ]
[ LIMIT number ]
```

Las cláusulas posibles de una consulta de Metrics Insights se detallan a continuación. Ninguna de las palabras clave distingue mayúsculas de minúsculas, al contrario de los identificadores como los nombres de las métricas, los espacios de nombres y las dimensiones.

SELECT

Obligatorio. Especifica la función que se va a utilizar para agregar observaciones en cada bucket de tiempo (determinada por el periodo proporcionado). También especifica el nombre de la métrica que se va a consultar.

Los valores válidos de FUNCTION (FUNCIÓN) son `AVG`, `COUNT`, `MAX`, `MIN`, y `SUM`.

- `AVG` calcula el promedio de las observaciones que coinciden con la consulta.
- `COUNT` devuelve el recuento de las observaciones que coinciden con la consulta.
- `MAX` devuelve el valor máximo de las observaciones que coinciden con la consulta.
- `MIN` devuelve el valor mínimo de las observaciones que coinciden con la consulta.
- `SUM` calcula la suma de las observaciones que coinciden con la consulta.

FROM

Obligatorio. Especifica el origen de la métrica. Puede especificar el espacio de nombres de métrica que contiene la métrica que se va a consultar o una función de tabla de SCHEMA (ESQUEMA). Algunos ejemplos de espacios de nombres de métrica incluyen `"AWS/EC2"`, `"AWS/Lambda"` y los espacios de nombres de métricas que usted creó para sus métricas personalizadas.

Los espacios de nombres de métrica que incluyen / o cualquier otro carácter que no sea una letra, un número o un guion bajo debe estar rodeado de comillas dobles. Para obtener más información, consulte ¿Dónde se deben colocar comillas o caracteres de escape?.

SCHEMA

Función de tabla opcional que se puede utilizar dentro de una cláusula de FROM (DESDE). Use SCHEMA (ESQUEMA) para reducir el alcance de los resultados de la consulta solo a las métricas que coinciden exactamente con una lista de dimensiones o métricas que no tienen dimensiones.

Si usa una cláusula de SCHEMA (ESQUEMA), esta debe contener al menos un argumento y este primer argumento debe ser el espacio de nombres de métrica que se consulta. Si especifica el SCHEMA (ESQUEMA) solo con este argumento de espacio de nombres, los resultados se reducen únicamente a métricas que no tienen dimensiones.

Si especifica SCHEMA (ESQUEMA) con argumentos adicionales, los argumentos adicionales posteriores al argumento de espacio de nombres deben ser claves de etiqueta. Las claves de etiqueta deben ser nombres de dimensión. Si especifica una o más de estas claves de etiqueta, los resultados se reducen únicamente a las métricas que tienen ese conjunto exacto de dimensiones. El orden de estas claves de etiqueta no es importante.

Por ejemplo:

- SELECCIONAR AVG(CPUUtilization) DESDE “AWS/EC2” coincide con todas las métricas de `CPUUtilization` en el espacio de nombres de `AWS/EC2`, independientemente de sus dimensiones, y devuelve una única serie temporal acumulada.
- SELECCIONAR AVG(CPUUtilization) DESDE ESQUEMA DE (“AWS/EC2”) solo coincide con las métricas de `CPUUtilization` en el espacio de nombres de `AWS/EC2` que no tiene ninguna dimensión definida.
- SELECCIONAR AVG(CPUUtilization) DESDE ESQUEMA DE (“AWS/EC2”, InstanceId) solo coincide con las métricas de `CPUUtilization` que se notificaron a CloudWatch con exactamente una dimensión, `InstanceId`.

- SELECCIONAR SUM(RequestCount) DESDE ESQUEMA DE (“AWS/ApplicationELB”, LoadBalancer, AvailabilityZone) solo coincide con las métricas de `RequestCount` que se han notificado a CloudWatch desde `AWS/ApplicationELB` con exactamente dos dimensiones, `LoadBalancer` y `AvailabilityZone`.

WHERE

Opcional. Filtra los resultados solo a aquellas métricas que coinciden con la expresión especificada mediante valores de etiqueta específicos para una o más claves de etiqueta. Por ejemplo, WHERE InstanceType = 'c3.4xlarge' (DONDE InstanceType = 'c3.4xlarge') filtra los resultados únicamente a los tipos de instancia `c3.4xlarge` y WHERE InstanceType != 'c3.4xlarge' (DONDE InstanceType != 'c3.4xlarge') filtra los resultados en todos los tipos de instancias, excepto `c3.4xlarge`.

Cuando ejecute una consulta en una cuenta de supervisión, puede utilizar `WHERE AWS.AccountId` para limitar los resultados únicamente a la cuenta que especifique. Por ejemplo, las métricas de consulta `WHERE AWS.AccountId=444455556666` únicamente de la cuenta `444455556666`. Para limitar la consulta solo a las métricas de la propia cuenta de supervisión, utilice `WHERE AWS.AccountId=CURRENT_ACCOUNT_ID()`.

Los valores de etiqueta siempre deben rodearse de comillas simples.

Operadores admitidos

La cláusula WHERE (DONDE) es compatible con los siguientes operadores:

- = El valor de etiqueta debe coincidir con la cadena especificada.
- != El valor de etiqueta no debe coincidir con la cadena especificada.
- AND (Y) Ambas condiciones especificadas deben ser verdaderas para que coincidan. Puede usar varias palabras clave AND (Y) para especificar dos o más condiciones.

GROUP BY

Opcional. Agrupa los resultados de la consulta en varias series temporales y cada una de ellas corresponde a un valor diferente para la clave o claves de etiqueta especificadas. Por ejemplo, si utiliza `GROUP BY InstanceId`, devuelve una serie temporal diferente para cada valor de `InstanceId`. El uso de `GROUP BY ServiceName, Operation` crea una serie temporal diferente para cada combinación posible de los valores de `ServiceName` y `Operation`.

Si utiliza la cláusula GROUP BY (AGRUPAR POR), los resultados se ordenan en orden alfabético ascendente de forma predeterminada, a través de la secuencia de etiquetas especificada en la

cláusula GROUP BY (AGRUPAR POR). Para cambiar el orden de los resultados, agregue una cláusula ORDER BY (AGRUPAR POR) a su consulta.

Cuando ejecuta una consulta en una cuenta de supervisión, puede utilizar `GROUP BY AWS.AccountId` para agrupar los resultados en función de las cuentas de las que provienen.

> **Note**
>
> Si algunas de las métricas coincidentes no incluyen una clave de etiqueta especificada en la cláusula GROUP BY (AGRUPAR POR), esto devuelve un grupo nulo denominado `Other`. Por ejemplo, si especifica `GROUP BY ServiceName, Operation` y algunas de las métricas devueltas no incluyen `ServiceName` como dimensión, entonces esas métricas se muestran con el valor `Other` para `ServiceName`.

ORDER BY

Opcional. Especifica el orden que se va a utilizar para la serie temporal devuelta, si la consulta devuelve más de una serie temporal. El orden se basa en los valores que encuentra la FUNCTION (FUNCIÓN) que usted especifique en la cláusula ORDER BY (AGRUPAR POR). La FUNCTION (FUNCIÓN) se utiliza para calcular un único valor escalar de cada serie temporal devuelta y ese valor se utiliza para determinar el orden.

También debe especificar si se va a utilizar orden ascendente ASC o descendente DESC. Si omite esto, el valor predeterminado es ascendente ASC.

Por ejemplo, al agregar una cláusula `ORDER BY MAX() DESC`, los resultados se ordenan según el punto de datos máximo observado dentro del intervalo de tiempo, en orden descendente, lo que significa que la serie temporal que tiene el punto de datos máximo más alto se devuelve primero.

Las funciones válidas que se deben utilizar dentro de una cláusula ORDER BY (AGRUPAR POR) son `AVG()`, `COUNT()`, `MAX()`, `MIN()`, y `SUM()`.

Si usa una cláusula de AGRUPAR POR con una cláusula de LÍMITE, la consulta resultante será una consulta “Top N”. ORDER BY (AGRUPAR POR) también es útil para consultas que pueden devolver un gran número de métricas porque cada consulta no puede devolver más de 500 series temporales. Si una consulta coincide con más de 500 series temporales y usted utiliza una cláusula de ORDER BY (AGRUPAR POR), las series temporales se ordenan y luego se devuelven las 500 series temporales que aparecen primero en el orden de clasificación.

LIMIT

Opcional. Limita la cantidad de series temporales devueltas según la consulta al valor especificado. El valor máximo que puede especificar es 500 y una consulta que no especifica un LIMIT (LÍMITE) también puede devolver hasta 500 series temporales.

El uso de una cláusula de LÍMITE con una de AGRUPAR POR le proporcionará una consulta “Top N”.

¿Dónde se deben colocar comillas o caracteres de escape?

En una consulta, los valores de etiqueta siempre deben estar rodeados de comillas simples. Por ejemplo, SELECCIONAR MAX(CPUUtilization) DESDE “AWS/EC2” DONDE AutoScalingGroupName = 'my-production-fleet'.

Los espacios de nombres de métrica, los nombres de métricas y las claves de etiqueta que contienen caracteres que no son letras, números y guiones bajos (_) deben estar rodeados de comillas dobles. Por ejemplo, SELECCIONAR MAX(“My.Metric”).

Si uno de ellos contiene comillas dobles o comillas simples en sí (como `Bytes"Input"`), debe escapar cada comilla con una barra invertida, como en SELECCIONAR AVG(“Bytes\“Input\””).

Si un espacio de nombres de métrica, un nombre de métrica o una clave de etiqueta contienen una palabra que es una palabra clave reservada en Metrics Insights, también debe rodearse de comillas dobles. Por ejemplo, si tiene una métrica denominada `LIMIT`, utilizaría `SELECT AVG("LIMIT")`. También es válido incluir cualquier espacio de nombres, nombre de métrica o etiqueta entre comillas dobles, incluso si no contiene una palabra clave reservada.

Para obtener una lista completa de palabras clave reservadas, consulte Palabras clave reservadas.

Crear una consulta enriquecida paso a paso

En esta sección se ilustra la creación de un ejemplo completo en el que se utilizan todas las cláusulas posibles, paso a paso.

Iniciamos con la siguiente consulta, que acumula todas las métricas de `RequestCount` del Application Load Balancer que se recopilan con ambas dimensiones: `LoadBalancer` y `AvailabilityZone`.

```
SELECT SUM(RequestCount)
```

```
FROM SCHEMA("AWS/ApplicationELB", LoadBalancer, AvailabilityZone)
```

Ahora, si deseamos ver métricas solo de un balanceador de carga específico, podemos agregar una cláusula WHERE (DONDE) para limitar las métricas devueltas solo a aquellas métricas donde el valor de la dimensión `LoadBalancer` es `app/load-balancer-1`.

```
SELECT SUM(RequestCount)
FROM SCHEMA("AWS/ApplicationELB", LoadBalancer, AvailabilityZone)
WHERE LoadBalancer = 'app/load-balancer-1'
```

La consulta mencionada acumula las métricas `RequestCount` de todas las zonas de disponibilidad para este equilibrador de carga en una serie temporal. Si queremos ver series temporales diferentes para cada zona de disponibilidad, podemos agregar una cláusula GROUP BY (AGRUPAR POR).

```
SELECT SUM(RequestCount)
FROM SCHEMA("AWS/ApplicationELB", LoadBalancer, AvailabilityZone)
WHERE LoadBalancer = 'app/load-balancer-1'
GROUP BY AvailabilityZone
```

A continuación, es posible que deseemos ordenar estos resultados para ver los valores más altos en primer lugar. La siguiente cláusula ORDER BY (AGRUPAR POR) ordena la serie temporal en orden descendente, según el valor máximo notificado por cada serie temporal durante el intervalo de tiempo de consulta:

```
SELECT SUM(RequestCount)
FROM SCHEMA("AWS/ApplicationELB", LoadBalancer, AvailabilityZone)
WHERE LoadBalancer = 'app/load-balancer-1'
GROUP BY AvailabilityZone
ORDER BY MAX() DESC
```

Por último, si nuestro interés principal es un tipo de consulta “Top N”, podemos utilizar una cláusula LÍMITE. Este último ejemplo limita los resultados únicamente a las series temporales con los cinco valores `MAX` más altos.

```
SELECT SUM(RequestCount)
FROM SCHEMA("AWS/ApplicationELB", LoadBalancer, AvailabilityZone)
WHERE LoadBalancer = 'app/load-balancer-1'
GROUP BY AvailabilityZone
ORDER BY MAX() DESC
LIMIT 5
```

Ejemplos de consultas entre cuentas

Estos ejemplos son válidos cuando se ejecutan en una cuenta configurada como cuenta de supervisión en la observabilidad entre cuentas de CloudWatch.

En el siguiente ejemplo se buscan todas las instancias de Amazon EC2 de la cuenta de origen 123456789012 y se indica un promedio.

```
SELECT AVG(CpuUtilization)
FROM "AWS/EC2"
WHERE AWS.AccountId ='123456789012'
```

En el siguiente ejemplo, se consulta la métrica de `CPUUtilization` en `AWS/EC2` en todas las cuentas de origen vinculadas y se agrupan los resultados por ID de cuenta y tipo de instancia.

```
SELECT AVG(CpuUtilization)
FROM "AWS/EC2"
GROUP BY AWS.AccountId, InstanceType
```

En el siguiente ejemplo, se consulta `CPUUtilization` en la propia cuenta de supervisión.

```
SELECT AVG(CpuUtilization)
FROM "AWS/EC2"
WHERE AWS.AccountId = CURRENT_ACCOUNT_ID()
```

Palabras clave reservadas

Las siguientes son palabras clave reservadas en CloudWatch Metrics Insights. Si alguna de estas palabras se encuentra en un espacio de nombres, un nombre de métrica o una clave de etiqueta de una consulta, debe rodearlas de comillas dobles. Las palabras clave reservadas no distinguen mayúsculas y minúsculas.

```
"ABORT" "ABORTSESSION" "ABS" "ABSOLUTE" "ACCESS" "ACCESSIBLE" "ACCESS_LOCK" "ACCOUNT"
 "ACOS" "ACOSH" "ACTION" "ADD" "ADD_MONTHS"
"ADMIN" "AFTER" "AGGREGATE" "ALIAS" "ALL" "ALLOCATE" "ALLOW" "ALTER" "ALTERAND" "AMP"
 "ANALYSE" "ANALYZE" "AND" "ANSIDATE" "ANY" "ARE" "ARRAY",
"ARRAY_AGG" "ARRAY_EXISTS" "ARRAY_MAX_CARDINALITY" "AS" "ASC" "ASENSITIVE" "ASIN"
 "ASINH" "ASSERTION" "ASSOCIATE" "ASUTIME" "ASYMMETRIC" "AT",
"ATAN" "ATAN2" "ATANH" "ATOMIC" "AUDIT" "AUTHORIZATION" "AUX" "AUXILIARY" "AVE"
 "AVERAGE" "AVG" "BACKUP" "BEFORE" "BEGIN" "BEGIN_FRAME" "BEGIN_PARTITION",
```

```
"BETWEEN" "BIGINT" "BINARY" "BIT" "BLOB" "BOOLEAN" "BOTH" "BREADTH" "BREAK" "BROWSE"
 "BT" "BUFFERPOOL" "BULK" "BUT" "BY" "BYTE" "BYTEINT" "BYTES" "CALL",
"CALLED" "CAPTURE" "CARDINALITY" "CASCADE" "CASCADED" "CASE" "CASESPECIFIC" "CASE_N"
 "CAST" "CATALOG" "CCSID" "CD" "CEIL" "CEILING" "CHANGE" "CHAR",
"CHAR2HEXINT" "CHARACTER" "CHARACTERS" "CHARACTER_LENGTH" "CHARS" "CHAR_LENGTH" "CHECK"
 "CHECKPOINT" "CLASS" "CLASSIFIER" "CLOB" "CLONE" "CLOSE" "CLUSTER",
"CLUSTERED" "CM" "COALESCE" "COLLATE" "COLLATION" "COLLECT" "COLLECTION" "COLLID"
 "COLUMN" "COLUMN_VALUE" "COMMENT" "COMMIT" "COMPLETION" "COMPRESS" "COMPUTE",
"CONCAT" "CONCURRENTLY" "CONDITION" "CONNECT" "CONNECTION" "CONSTRAINT" "CONSTRAINTS"
 "CONSTRUCTOR" "CONTAINS" "CONTAINSTABLE" "CONTENT" "CONTINUE" "CONVERT",
"CONVERT_TABLE_HEADER" "COPY" "CORR" "CORRESPONDING" "COS" "COSH" "COUNT" "COVAR_POP"
 "COVAR_SAMP" "CREATE" "CROSS" "CS" "CSUM" "CT" "CUBE" "CUME_DIST",
"CURRENT" "CURRENT_CATALOG" "CURRENT_DATE" "CURRENT_DEFAULT_TRANSFORM_GROUP"
 "CURRENT_LC_CTYPE" "CURRENT_PATH" "CURRENT_ROLE" "CURRENT_ROW" "CURRENT_SCHEMA",
"CURRENT_SERVER" "CURRENT_TIME" "CURRENT_TIMESTAMP" "CURRENT_TIMEZONE"
 "CURRENT_TRANSFORM_GROUP_FOR_TYPE" "CURRENT_USER" "CURRVAL" "CURSOR" "CV" "CYCLE"
 "DATA",
"DATABASE" "DATABASES" "DATABLOCKSIZE" "DATE" "DATEFORM" "DAY" "DAYS" "DAY_HOUR"
 "DAY_MICROSECOND" "DAY_MINUTE" "DAY_SECOND" "DBCC" "DBINFO" "DEALLOCATE" "DEC",
"DECFLOAT" "DECIMAL" "DECLARE" "DEFAULT" "DEFERRABLE" "DEFERRED" "DEFINE" "DEGREES"
 "DEL" "DELAYED" "DELETE" "DENSE_RANK" "DENY" "DEPTH" "DEREF" "DESC" "DESCRIBE",
"DESCRIPTOR" "DESTROY" "DESTRUCTOR" "DETERMINISTIC" "DIAGNOSTIC" "DIAGNOSTICS"
 "DICTIONARY" "DISABLE" "DISABLED" "DISALLOW" "DISCONNECT" "DISK" "DISTINCT",
"DISTINCTROW" "DISTRIBUTED" "DIV" "DO" "DOCUMENT" "DOMAIN" "DOUBLE" "DROP" "DSSIZE"
 "DUAL" "DUMP" "DYNAMIC" "EACH" "ECHO" "EDITPROC" "ELEMENT" "ELSE" "ELSEIF",
"EMPTY" "ENABLED" "ENCLOSED" "ENCODING" "ENCRYPTION" "END" "END-EXEC" "ENDING"
 "END_FRAME" "END_PARTITION" "EQ" "EQUALS" "ERASE" "ERRLVL" "ERROR" "ERRORFILES",
"ERRORTABLES" "ESCAPE" "ESCAPED" "ET" "EVERY" "EXCEPT" "EXCEPTION" "EXCLUSIVE" "EXEC"
 "EXECUTE" "EXISTS" "EXIT" "EXP" "EXPLAIN" "EXTERNAL" "EXTRACT" "FALLBACK
"FALSE" "FASTEXPORT" "FENCED" "FETCH" "FIELDPROC" "FILE" "FILLFACTOR" "FILTER" "FINAL"
 "FIRST" "FIRST_VALUE" "FLOAT" "FLOAT4" "FLOAT8" "FLOOR"
"FOR" "FORCE" "FOREIGN" "FORMAT" "FOUND" "FRAME_ROW" "FREE" "FREESPACE" "FREETEXT"
 "FREETEXTTABLE" "FREEZE" "FROM" "FULL" "FULLTEXT" "FUNCTION"
"FUSION" "GE" "GENERAL" "GENERATED" "GET" "GIVE" "GLOBAL" "GO" "GOTO" "GRANT" "GRAPHIC"
 "GROUP" "GROUPING" "GROUPS" "GT" "HANDLER" "HASH"
"HASHAMP" "HASHBAKAMP" "HASHBUCKET" "HASHROW" "HAVING" "HELP" "HIGH_PRIORITY" "HOLD"
 "HOLDLOCK" "HOUR" "HOURS" "HOUR_MICROSECOND" "HOUR_MINUTE"
"HOUR_SECOND" "IDENTIFIED" "IDENTITY" "IDENTITYCOL" "IDENTITY_INSERT" "IF" "IGNORE"
 "ILIKE" "IMMEDIATE" "IN" "INCLUSIVE" "INCONSISTENT" "INCREMENT"
"INDEX" "INDICATOR" "INFILE" "INHERIT" "INITIAL" "INITIALIZE" "INITIALLY" "INITIATE"
 "INNER" "INOUT" "INPUT" "INS" "INSENSITIVE" "INSERT" "INSTEAD"
"INT" "INT1" "INT2" "INT3" "INT4" "INT8" "INTEGER" "INTEGERDATE" "INTERSECT"
 "INTERSECTION" "INTERVAL" "INTO" "IO_AFTER_GTIDS" "IO_BEFORE_GTIDS"
```

```
"IS" "ISNULL" "ISOBID" "ISOLATION" "ITERATE" "JAR" "JOIN" "JOURNAL" "JSON_ARRAY"
 "JSON_ARRAYAGG" "JSON_EXISTS" "JSON_OBJECT" "JSON_OBJECTAGG"
"JSON_QUERY" "JSON_TABLE" "JSON_TABLE_PRIMITIVE" "JSON_VALUE" "KEEP" "KEY" "KEYS"
 "KILL" "KURTOSIS" "LABEL" "LAG" "LANGUAGE" "LARGE" "LAST"
"LAST_VALUE" "LATERAL" "LC_CTYPE" "LE" "LEAD" "LEADING" "LEAVE" "LEFT" "LESS" "LEVEL"
 "LIKE" "LIKE_REGEX" "LIMIT" "LINEAR" "LINENO" "LINES"
"LISTAGG" "LN" "LOAD" "LOADING" "LOCAL" "LOCALE" "LOCALTIME" "LOCALTIMESTAMP" "LOCATOR"
 "LOCATORS" "LOCK" "LOCKING" "LOCKMAX" "LOCKSIZE" "LOG"
"LOG10" "LOGGING" "LOGON" "LONG" "LONGBLOB" "LONGTEXT" "LOOP" "LOWER" "LOW_PRIORITY"
 "LT" "MACRO" "MAINTAINED" "MAP" "MASTER_BIND"
"MASTER_SSL_VERIFY_SERVER_CERT" "MATCH" "MATCHES" "MATCH_NUMBER" "MATCH_RECOGNIZE"
 "MATERIALIZED" "MAVG" "MAX" "MAXEXTENTS" "MAXIMUM" "MAXVALUE"
"MCHARACTERS" "MDIFF" "MEDIUMBLOB" "MEDIUMINT" "MEDIUMTEXT" "MEMBER" "MERGE" "METHOD"
 "MICROSECOND" "MICROSECONDS" "MIDDLEINT" "MIN" "MINDEX"
"MINIMUM" "MINUS" "MINUTE" "MINUTES" "MINUTE_MICROSECOND" "MINUTE_SECOND" "MLINREG"
 "MLOAD" "MLSLABEL" "MOD" "MODE" "MODIFIES" "MODIFY"
"MODULE" "MONITOR" "MONRESOURCE" "MONSESSION" "MONTH" "MONTHS" "MSUBSTR" "MSUM"
 "MULTISET" "NAMED" "NAMES" "NATIONAL" "NATURAL" "NCHAR" "NCLOB"
"NE" "NESTED_TABLE_ID" "NEW" "NEW_TABLE" "NEXT" "NEXTVAL" "NO" "NOAUDIT" "NOCHECK"
 "NOCOMPRESS" "NONCLUSTERED" "NONE" "NORMALIZE" "NOT" "NOTNULL"
"NOWAIT" "NO_WRITE_TO_BINLOG" "NTH_VALUE" "NTILE" "NULL" "NULLIF" "NULLIFZERO" "NULLS"
 "NUMBER" "NUMERIC" "NUMPARTS" "OBID" "OBJECT" "OBJECTS"
"OCCURRENCES_REGEX" "OCTET_LENGTH" "OF" "OFF" "OFFLINE" "OFFSET" "OFFSETS" "OLD"
 "OLD_TABLE" "OMIT" "ON" "ONE" "ONLINE" "ONLY" "OPEN" "OPENDATASOURCE"
"OPENQUERY" "OPENROWSET" "OPENXML" "OPERATION" "OPTIMIZATION" "OPTIMIZE"
 "OPTIMIZER_COSTS" "OPTION" "OPTIONALLY" "OR" "ORDER" "ORDINALITY" "ORGANIZATION"
"OUT" "OUTER" "OUTFILE" "OUTPUT" "OVER" "OVERLAPS" "OVERLAY" "OVERRIDE" "PACKAGE" "PAD"
 "PADDED" "PARAMETER" "PARAMETERS" "PART" "PARTIAL" "PARTITION"
"PARTITIONED" "PARTITIONING" "PASSWORD" "PATH" "PATTERN" "PCTFREE" "PER" "PERCENT"
 "PERCENTILE" "PERCENTILE_CONT" "PERCENTILE_DISC" "PERCENT_RANK" "PERIOD" "PERM"
"PERMANENT" "PIECESIZE" "PIVOT" "PLACING" "PLAN" "PORTION" "POSITION" "POSITION_REGEX"
 "POSTFIX" "POWER" "PRECEDES" "PRECISION" "PREFIX" "PREORDER"
"PREPARE" "PRESERVE" "PREVVAL" "PRIMARY" "PRINT" "PRIOR" "PRIQTY" "PRIVATE"
 "PRIVILEGES" "PROC" "PROCEDURE" "PROFILE" "PROGRAM" "PROPORTIONAL"
"PROTECTION" "PSID" "PTF" "PUBLIC" "PURGE" "QUALIFIED" "QUALIFY" "QUANTILE" "QUERY"
 "QUERYNO" "RADIANS" "RAISERROR" "RANDOM" "RANGE" "RANGE_N" "RANK"
"RAW" "READ" "READS" "READTEXT" "READ_WRITE" "REAL" "RECONFIGURE" "RECURSIVE" "REF"
 "REFERENCES" "REFERENCING" "REFRESH" "REGEXP" "REGR_AVGX" "REGR_AVGY"
"REGR_COUNT" "REGR_INTERCEPT" "REGR_R2" "REGR_SLOPE" "REGR_SXX" "REGR_SXY" "REGR_SYY"
 "RELATIVE" "RELEASE" "RENAME" "REPEAT" "REPLACE" "REPLICATION"
"REPOVERRIDE" "REQUEST" "REQUIRE" "RESIGNAL" "RESOURCE" "RESTART" "RESTORE" "RESTRICT"
 "RESULT" "RESULT_SET_LOCATOR" "RESUME" "RET" "RETRIEVE" "RETURN"
"RETURNING" "RETURNS" "REVALIDATE" "REVERT" "REVOKE" "RIGHT" "RIGHTS" "RLIKE" "ROLE"
 "ROLLBACK" "ROLLFORWARD" "ROLLUP" "ROUND_CEILING" "ROUND_DOWN"
```

```
"ROUND_FLOOR" "ROUND_HALF_DOWN" "ROUND_HALF_EVEN" "ROUND_HALF_UP" "ROUND_UP" "ROUTINE"
 "ROW" "ROWCOUNT" "ROWGUIDCOL" "ROWID" "ROWNUM" "ROWS" "ROWSET"
"ROW_NUMBER" "RULE" "RUN" "RUNNING" "SAMPLE" "SAMPLEID" "SAVE" "SAVEPOINT" "SCHEMA"
 "SCHEMAS" "SCOPE" "SCRATCHPAD" "SCROLL" "SEARCH" "SECOND" "SECONDS"
"SECOND_MICROSECOND" "SECQTY" "SECTION" "SECURITY" "SECURITYAUDIT" "SEEK" "SEL"
 "SELECT" "SEMANTICKEYPHRASETABLE" "SEMANTICSIMILARITYDETAILSTABLE"
"SEMANTICSIMILARITYTABLE" "SENSITIVE" "SEPARATOR" "SEQUENCE" "SESSION" "SESSION_USER"
 "SET" "SETRESRATE" "SETS" "SETSESSRATE" "SETUSER" "SHARE" "SHOW"
"SHUTDOWN" "SIGNAL" "SIMILAR" "SIMPLE" "SIN" "SINH" "SIZE" "SKEW" "SKIP" "SMALLINT"
 "SOME" "SOUNDEX" "SOURCE" "SPACE" "SPATIAL" "SPECIFIC" "SPECIFICTYPE"
"SPOOL" "SQL" "SQLEXCEPTION" "SQLSTATE" "SQLTEXT" "SQLWARNING" "SQL_BIG_RESULT"
 "SQL_CALC_FOUND_ROWS" "SQL_SMALL_RESULT" "SQRT" "SS" "SSL" "STANDARD"
"START" "STARTING" "STARTUP" "STAT" "STATE" "STATEMENT" "STATIC" "STATISTICS" "STAY"
 "STDDEV_POP" "STDDEV_SAMP" "STEPINFO" "STOGROUP" "STORED" "STORES"
"STRAIGHT_JOIN" "STRING_CS" "STRUCTURE" "STYLE" "SUBMULTISET" "SUBSCRIBER" "SUBSET"
 "SUBSTR" "SUBSTRING" "SUBSTRING_REGEX" "SUCCEEDS" "SUCCESSFUL"
"SUM" "SUMMARY" "SUSPEND" "SYMMETRIC" "SYNONYM" "SYSDATE" "SYSTEM" "SYSTEM_TIME"
 "SYSTEM_USER" "SYSTIMESTAMP" "TABLE" "TABLESAMPLE" "TABLESPACE" "TAN"
"TANH" "TBL_CS" "TEMPORARY" "TERMINATE" "TERMINATED" "TEXTSIZE" "THAN" "THEN"
 "THRESHOLD" "TIME" "TIMESTAMP" "TIMEZONE_HOUR" "TIMEZONE_MINUTE" "TINYBLOB"
"TINYINT" "TINYTEXT" "TITLE" "TO" "TOP" "TRACE" "TRAILING" "TRAN" "TRANSACTION"
 "TRANSLATE" "TRANSLATE_CHK" "TRANSLATE_REGEX" "TRANSLATION" "TREAT"
"TRIGGER" "TRIM" "TRIM_ARRAY" "TRUE" "TRUNCATE" "TRY_CONVERT" "TSEQUAL" "TYPE" "UC"
 "UESCAPE" "UID" "UNDEFINED" "UNDER" "UNDO" "UNION" "UNIQUE"
"UNKNOWN" "UNLOCK" "UNNEST" "UNPIVOT" "UNSIGNED" "UNTIL" "UPD" "UPDATE" "UPDATETEXT"
 "UPPER" "UPPERCASE" "USAGE" "USE" "USER" "USING" "UTC_DATE"
"UTC_TIME" "UTC_TIMESTAMP" "VALIDATE" "VALIDPROC" "VALUE" "VALUES" "VALUE_OF"
 "VARBINARY" "VARBYTE" "VARCHAR" "VARCHAR2" "VARCHARACTER" "VARGRAPHIC"
"VARIABLE" "VARIADIC" "VARIANT" "VARYING" "VAR_POP" "VAR_SAMP" "VCAT" "VERBOSE"
 "VERSIONING" "VIEW" "VIRTUAL" "VOLATILE" "VOLUMES" "WAIT" "WAITFOR"
"WHEN" "WHENEVER" "WHERE" "WHILE" "WIDTH_BUCKET" "WINDOW" "WITH" "WITHIN"
 "WITHIN_GROUP" "WITHOUT" "WLM" "WORK" "WRITE" "WRITETEXT" "XMLCAST" "XMLEXISTS"
"XMLNAMESPACES" "XOR" "YEAR" "YEARS" "YEAR_MONTH" "ZEROFILL" "ZEROIFNULL" "ZONE"
```

Creación de alarmas en las consultas de Información de métricas

Puede crear alarmas en las consultas de Información de métricas Esto le permite tener alarmas que rastreen varios recursos sin necesidad de actualizarlas más adelante. La consulta detecta los nuevos recursos y aquellos que cambien. Por ejemplo, puede crear una alarma que controle el uso de la CPU de la flota y que evalúe automáticamente las nuevas instancias que lance después de crearla.

En una cuenta de supervisión que esté configurada para la observabilidad entre cuentas de CloudWatch, sus alarmas de Información de métricas pueden supervisar los recursos en las cuentas

de origen y en la propia cuenta de supervisión. Para obtener más información sobre cómo limitar las consultas de alarmas a una cuenta específica o cómo agrupar los resultados por ID de cuenta, consulte las secciones `WHERE` y `GROUP BY` de Componentes de consulta y sintaxis de Metrics Insights.

Contenido

Creación de una alarma Información de métricas

Creación de una alarma en una consulta de Información de métricas mediante la consola

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, seleccione Métricas y, a continuación, Todas las métricas.
3. Elija la pestaña Queries (Consultas).
4. (Opcional) Para ejecutar una consulta de ejemplo prediseñada, elija Add query (Agregar consulta) y seleccione la consulta que se va a ejecutar. De lo contrario, puede elegir Editor para editar la consulta de ejemplo y luego, Run (Ejecutar) para ejecutar la consulta modificada.
5. Para crear su propia consulta, puede utilizar la vista de Builder (Generador), la vista de Editor (Editor) y, también, una combinación de ambas. Puede cambiar entre las dos vistas en cualquier momento y ver su trabajo en curso en ambas vistas.

   En la vista Builder (Generador), puede examinar y seleccionar el espacio de nombre de métrica, el nombre de métrica, el filtro, el grupo y las opciones de pedido. Para cada una de estas opciones, el generador de consultas le ofrece una lista de posibles selecciones de su entorno de donde puede elegir.

   En la vista Editor, puede comenzar a escribir su consulta. A medida que escribe, el editor ofrece sugerencias basadas en los caracteres que ha escrito hasta el momento.

   > **Important**
   >
   > Para configurar una alarma en una consulta de Información de métricas, la consulta debe devolver una sola serie temporal. Si contiene una instrucción GROUP BY, tenga en

cuenta que esta debe incluirse en una expresión matemática de métrica que devuelva una única serie temporal como resultado final de la expresión.

6. Cuando esté satisfecho con la consulta, elija Save (Guardar).
7. Elija Crear alarma.
8. En Conditions (Condiciones), especifique lo siguiente:

   a. En Whenever ***`metric`*** is (Siempre que la métrica sea), especifique si la métrica debe ser mayor que, menor que o igual al umbral. En than... (que...), especifique el valor de umbral.

   b. Elija Configuración adicional. Para Puntos de datos para alarma, especifique el número de periodos de evaluación (puntos de datos) que deben tener el estado `ALARM` para que se active la alarma. Si estos dos valores coinciden, creará una alarma que pasará al estado `ALARM` si se infringen muchos periodos consecutivos.

      Para crear una alarma M de N, especifique un número menor para el primer valor que el especificado para el segundo valor. Para obtener más información, consulte Evaluación de una alarma.

   c. En Missing data treatment (Tratamiento de datos que faltan), elija cómo debe comportarse la alarma cuando falten algunos puntos de datos. Para obtener más información, consulte Configuración de la forma en la que las alarmas de CloudWatch tratan los datos que faltan.

9. Elija Siguiente.
10. En Notification (Notificación), seleccione el tema de SNS al que desee enviar la notificación cuando la alarma tenga el estado `ALARM`, `OK` o `INSUFFICIENT_DATA`.

    Para que la alarma envíe varias notificaciones para el mismo estado de alarma o para estados de alarma diferentes, seleccione Add notificación (Añadir notificación).

    Para que la alarma no envíe notificaciones, elija Remove (Eliminar).
11. Para que la alarma realice acciones de Auto Scaling, EC2 o de Systems Manager, elija el botón correspondiente y elija el estado de la alarma y la acción que se debe realizar. Las alarmas solo pueden realizar acciones de Systems Manager cuando entran en el estado ALARMA. Para obtener más información sobre las acciones de Systems Manager, consulte Configuración de CloudWatch para crear OpsItems desde alarmas y Creación de incidentes.

> **Note**
>
> Para crear una alarma que realice una acción de SSM Incident Manager, debe contar con determinados permisos. Para obtener más información, consulte Ejemplos de políticas basadas en identidades del Administrador de incidentes de AWS Systems Manager.

12. Cuando haya terminado, elija Next (Siguiente).
13. Escriba un nombre y la descripción de la alarma. El nombre solo debe contener caracteres ASCII. A continuación, elija Next.
14. En Preview and create (Obtener vista previa y crear), confirme que la información y las condiciones son las que desea y, a continuación, elija Create alarm (Crear alarma).

Creación de una alarma en una consulta de Información de métricas mediante AWS CLI

- Utilice el comando `put-metric-alarm` y especifique una consulta de Información de métricas en el parámetro `metrics`. Por ejemplo, el siguiente comando establece una alarma que pase al estado ALARM si alguna de las instancias supera el 50 % de uso de la CPU.

```
aws cloudwatch put-metric-alarm --alarm-name Metrics-Insights-alarm --
evaluation-periods 1 --comparison-operator GreaterThanThreshold --metrics
 '[{"Id":"m1","Expression":"SELECT MAX(CPUUtilization) FROM SCHEMA(\"AWS/EC2\",
 InstanceId)", "Period":60}]' --threshold 50
```

Casos de datos parciales

Si la consulta de Información de métricas que se use en la alarma coincide con más de 10 000 métricas, la alarma se evalúa en función de las 10 000 primeras métricas que encuentre la consulta. Esto significa que la alarma se evalúa con datos parciales.

Puede utilizar los siguientes métodos para averiguar si una alarma de Información de métricas está evaluando actualmente su estado de alarma según los datos parciales:

- En la consola, si elige una alarma para ver la página de detalles, aparecerá el mensaje Evaluation warning: Not evaluating all data (Advertencia de evaluación: no se están evaluando todos los datos) en la misma página.

- Verá el valor `PARTIAL_DATA` en el campo `EvaluationState` cuando use el comando AWS CLI describe-alarms o la API DescribeAlarms.

Las alarmas también publican eventos en Amazon EventBridge cuando pasa al estado de datos parciales, por lo que puede crear una regla de EventBridge para controlar estos eventos. En estos casos, el campo `evaluationState` tiene el valor `PARTIAL_DATA`. A continuación, se muestra un ejemplo.

```
{
    "version": "0",
    "id": "12345678-3bf9-6a09-dc46-12345EXAMPLE",
    "detail-type": "CloudWatch Alarm State Change",
    "source": "aws.cloudwatch",
    "account": "123456789012",
    "time": "2022-11-08T11:26:05Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:cloudwatch:us-east-1:123456789012:alarm:my-alarm-name"
    ],
    "detail": {
        "alarmName": "my-alarm-name",
        "state": {
            "value": "ALARM",
            "reason": "Threshold Crossed: 3 out of the last 3 datapoints [20000.0
 (08/11/22 11:25:00), 20000.0 (08/11/22 11:24:00), 20000.0 (08/11/22 11:23:00)] were
 greater than the threshold (0.0) (minimum 1 datapoint for OK -> ALARM transition).",
            "reasonData": "{\"version\":\"1.0\",\"queryDate\":
\"2022-11-08T11:26:05.399+0000\",\"startDate\":\"2022-11-08T11:23:00.000+0000\",
\"period\":60,\"recentDatapoints\":[20000.0,20000.0,20000.0],\"threshold\":0.0,
\"evaluatedDatapoints\":[{\"timestamp\":\"2022-11-08T11:25:00.000+0000\",\"value
\":20000.0}]}",
            "timestamp": "2022-11-08T11:26:05.401+0000",
            "evaluationState": "PARTIAL_DATA"
        },
        "previousState": {
            "value": "INSUFFICIENT_DATA",
            "reason": "Unchecked: Initial alarm creation",
            "timestamp": "2022-11-08T11:25:51.227+0000"
        },
        "configuration": {
            "metrics": [
                {
```

```
                    "id": "m2",
                    "expression": "SELECT SUM(PartialDataTestMetric) FROM
 partial_data_test",
                    "returnData": true,
                    "period": 60
                }
            ]
        }
    }
}
```

Si la consulta de la alarma incluye una expresión GROUP BY que inicialmente devuelve más de 500 series temporales, la alarma se evalúa en función de las 500 primeras series temporales que encuentre la consulta. Sin embargo, si usa una cláusula ORDER BY, todas las series temporales que encuentre la consulta se ordenan y las 500 que tengan los valores más altos o más bajos según su cláusula ORDER BY se usan para evaluar la alarma.

Uso de consultas de Metrics Insights con matemáticas de métricas

Puede utilizar una consulta de Metrics Insights como entrada de una función matemática métrica. Para obtener más información acerca de las matemáticas de métricas, consulte Uso de la calculadora de métricas.

Una consulta de Metrics Insights que no incluye una cláusula GROUP BY (AGRUPAR POR) devuelve una sola serie temporal. Por lo tanto, los resultados devueltos se pueden utilizar con cualquier función matemática métrica que tome como entrada una única serie temporal.

Una consulta de Metrics Insights que incluye una cláusula GROUP BY (AGRUPAR POR) devuelve varias series temporales. Por lo tanto, los resultados devueltos se pueden utilizar con cualquier función matemática métrica que tome como entrada una matriz de series temporales.

Por ejemplo, la siguiente consulta devuelve el número total de bytes descargados para cada bucket de la región como una matriz de series temporales:

```
SELECT SUM(BytesDownloaded)
FROM SCHEMA("AWS/S3", BucketName, FilterId)
WHERE FilterId = 'EntireBucket'
GROUP BY BucketName
```

En un gráfico de la consola o en una operación de GetMetricData, los resultados de esta consulta son q1. Esta consulta devuelve el resultado en bytes, por lo que si desea ver el resultado como MB, puede utilizar la siguiente función matemática:

```
q1/1024/1024
```

Uso de lenguaje natural para generar y actualizar consultas de Información de métricas de CloudWatch

Esta característica está en versión preliminar en las regiones Este de EE. UU. (Norte de Virginia), Oeste de EE. UU. (Oregón) y Asia-Pacífico (Tokio) para CloudWatch y está sujeta a cambios.

CloudWatch admite la función de consulta en lenguaje natural que le ayuda a generar y actualizar consultas para Información de métricas de CloudWatch e Información de registros de CloudWatch.

Con esta función, puede hacer preguntas o describir los datos de CloudWatch que busca en un lenguaje sencillo. Esta función de lenguaje natural genera una consulta según una petición presentada y proporciona una explicación línea por línea sobre cómo funciona la consulta. También puede actualizar la consulta para investigar más a fondo los datos.

Según el entorno, puede introducir peticiones como “¿Qué instancia de Amazon Elastic Compute Cloud presenta la mayor salida de red?” y “Mostrarme las 10 tablas principales de Amazon DynamoDB por lecturas consumidas”.

Para generar una consulta de Información de métricas de CloudWatch con esta función, abra el editor de consultas de Información de métricas de CloudWatch en la vista del generador o editor y seleccione Generar consulta.

Important

Para usar la función de consulta en lenguaje natural, debe usar la política CloudwatchFullAccess, CloudwatchReadOnlyAccess, CloudWatchFullAccessV2,AdministratorAccess o ReadOnlyAccess. También puede incluir la acción `cloudwatch:GenerateQuery` en una política integrada o administrada por el cliente, nueva o existente.

Consultas de ejemplo

Los ejemplos en esta sección describen cómo generar y actualizar consultas mediante la función de lenguaje natural.

> **Note**
>
> Para obtener más información sobre el editor y la sintaxis de consultas de Información de métricas de CloudWatch, consulte Componentes y sintaxis de las consultas de Información de métricas de CloudWatch.

Ejemplo: generar una consulta en lenguaje natural

Para generar una consulta en lenguaje natural, introduzca una petición y seleccione Generar nueva consulta. En este ejemplo se muestra una consulta que realiza una búsqueda básica.

Prompt

A continuación, se muestra un ejemplo de una petición que indica la función de buscar las 10 tablas principales de DynamoDB que consumen más capacidad de lectura.

```
Show top 10 DynamoDB Tables by consumed reads
```

Consultar

El siguiente es un ejemplo de una consulta que la función de lenguaje natural genera según la petición. Observe cómo se muestra la petición en un comentario antes de la consulta. Tras la consulta, puede leer una explicación que describe cómo funciona la consulta.

```
# Show top 10 DynamoDB Tables by consumed reads
SELECT SUM("ConsumedReadCapacityUnits")
FROM "AWS/DynamoDB"
GROUP BY TableName
ORDER BY SUM() DESC
LIMIT 10
# This query selects the sum of consumed read capacity units for each DynamoDB table,
 groups the results by table name, orders the results from highest to lowest read
 capacity consumption, and limits the results to the top 10 tables.
```

> **Note**
>
> Para desactivar el aspecto de la petición y la explicación de cómo funciona la consulta, use el icono de engranaje del editor.

Ejemplo: actualizar una consulta en lenguaje natural

Puede actualizar una consulta al editar la petición inicial y, a continuación, seleccionar Actualizar consulta.

Petición actualizada

El siguiente ejemplo muestra una versión actualizada de la petición anterior. En lugar de una solicitud que busca las 10 tablas principales de DynamoDB que consumen más capacidad de lectura, esta petición ahora indica la capacidad de ordenar los resultados por el número de bytes proporcionados.

```
Sort by bytes returned instead
```

Consulta actualizada

A continuación, se muestra un ejemplo de una consulta actualizada. Observe cómo se muestra la petición actualizada en un comentario antes de la consulta actualizada. Tras la consulta, puede leer una explicación que describe cómo se actualizó la consulta original.

```
# Sort by bytes returned instead
SELECT SUM("ReturnedBytes")
FROM "AWS/DynamoDB"
GROUP BY TableName
ORDER BY SUM() DESC
LIMIT 10
# This query modifies the original query to select the sum of returned bytes instead
 of consumed read capacity units, and orders the results from highest to lowest sum of
 returned bytes, limiting the results to the top 10 tables.
```

Optar por no utilizar sus datos para mejorar el servicio

Los datos de la petición en lenguaje natural que proporciona para entrenar el modelo de IA y generar consultas relevantes se utilizan únicamente para proporcionar y mantener su servicio. Estos datos podrían usarse para mejorar la calidad de Información de métricas de CloudWatch. La confianza

y privacidad, como así también la seguridad de su contenido, son nuestra máxima prioridad. Para obtener más información, consulte Condiciones del servicio de AWS y Política de IA responsable de AWS.

Puede optar por que su contenido no se utilice para desarrollar o mejorar la calidad de las consultas en lenguaje natural mediante la creación de una política de exclusión de los servicios de IA. Para excluirse de la recopilación de datos para todas las características de IA de CloudWatch, incluida la función de generación de consultas, debe crear una política de exclusión para CloudWatch. Para obtener más información, consulte Políticas de exclusión de servicios de IA en la Guía del usuario de AWS Organizations.

Inferencia en SQL

En CloudWatch Metrics Insights se utilizan varios mecanismos para inferir la intención de una consulta en SQL determinada.

Temas

- Bucket de tiempo
- Proyección de campos
- Acumulación global de ORDER BY (AGRUPAR POR)

Bucket de tiempo

Los puntos de datos de serie temporal resultantes de una consulta se acumulan en buckets de tiempo según el periodo solicitado. Para acumular valores en SQL estándar, se debe definir una cláusula GROUP BY explícita para recopilar todas las observaciones de un periodo determinado juntas. Debido a que esta es la forma estándar de consultar datos de serie temporal, CloudWatch Metrics Insights deduce el bucket de tiempo sin necesidad de que se exprese la cláusula GROUP BY de forma explícita.

Por ejemplo, cuando se realiza una consulta con un periodo de un minuto, todas las observaciones pertenecientes a ese minuto hasta el siguiente (excluido) se acumulan hasta la hora de inicio del bucket de tiempo. Esto hace que las sentencias SQL de Metrics Insights sean más concisas y menos detalladas.

```
SELECT AVG(CPUUtilization)
FROM SCHEMA("AWS/EC2", InstanceId)
```

La consulta previa devuelve una única serie temporal (pares de marca temporal y valor), lo que representa la utilización promedio de la CPU de todas las instancias de Amazon EC2. Suponiendo que el periodo solicitado sea de un minuto, cada punto de datos devuelto representa el promedio de todas las observaciones medidas dentro de un intervalo específico de un minuto (se incluye la hora de inicio y no la hora de finalización). La marca temporal relacionada con el punto de datos específico es la hora de inicio del bucket.

Proyección de campos

Las consultas de Metrics Insights siempre devuelven la proyección de marca temporal. No es necesario especificar una columna de marca temporal en la cláusula SELECT para obtener la marca temporal de cada valor de punto de datos correspondiente. Para obtener información detallada acerca de cómo se calcula la marca temporal, consulte Bucket de tiempo.

Cuando se utiliza GROUP BY, el nombre de cada grupo también se deduce y proyecta en el resultado, para que usted pueda agrupar las series temporales devueltas.

```
SELECT AVG(CPUUtilization)
FROM SCHEMA("AWS/EC2", InstanceId)
GROUP BY InstanceId
```

La consulta anterior devuelve una serie temporal para cada instancia de Amazon EC2. Cada serie temporal se etiqueta después del valor del ID de instancia.

Acumulación global de ORDER BY (AGRUPAR POR)

Cuando se utiliza ORDER BY (AGRUPAR POR), FUNCTION() (FUNCIÓN) deduce la función de acumulación según la cual desea ordenar (los valores de punto de datos de las métricas consultadas). La operación de acumulación se realiza en todos los puntos de datos coincidentes de cada serie temporal individual en el periodo en el que se realizó la consulta.

```
SELECT AVG(CPUUtilization)
FROM SCHEMA("AWS/EC2", InstanceId)
GROUP BY InstanceId
ORDER BY MAX()
LIMIT 10
```

La consulta anterior devuelve la utilización de la CPU para cada instancia de Amazon EC2, lo que limita el conjunto de resultados a 10 entradas. Los resultados se ordenan en función del valor máximo de las series temporales individuales dentro del periodo solicitado. La cláusula ORDER BY

(AGRUPAR POR) se aplica antes de LIMIT (LÍMITE), de modo que el pedido se calcula con más de 10 series temporales.

Consultas de ejemplo de Metrics Insights

Esta sección contiene ejemplos de consultas útiles de CloudWatch Metrics Insights que puede copiar y utilizar directamente o copiar y modificar en el editor de consultas. Algunos de estos ejemplos ya están disponibles en la consola; para obtener acceso a ellos, elija Add query (Agregar consulta) en la vista de Metrics (Métricas).

Ejemplos de Application Load Balancer

Total de solicitudes en todos los balanceadores de carga

```
SELECT SUM(RequestCount)
FROM SCHEMA("AWS/ApplicationELB", LoadBalancer)
```

Los 10 balanceadores de carga más activos

```
SELECT MAX(ActiveConnectionCount)
FROM SCHEMA("AWS/ApplicationELB", LoadBalancer)
GROUP BY LoadBalancer
ORDER BY SUM() DESC
LIMIT 10
```

Ejemplos de uso de la API de AWS

Las 20 API principales de AWS según la cantidad de llamadas de su cuenta

```
SELECT COUNT(CallCount)
FROM SCHEMA("AWS/Usage", Class, Resource, Service, Type)
WHERE Type = 'API'
GROUP BY Service, Resource
ORDER BY COUNT() DESC
LIMIT 20
```

API de CloudWatch ordenadas por llamadas

```
SELECT COUNT(CallCount)
FROM SCHEMA("AWS/Usage", Class, Resource, Service, Type)
WHERE Type = 'API' AND Service = 'CloudWatch'
```

```
GROUP BY Resource
ORDER BY COUNT() DESC
```

Ejemplos de DynamoDB

Las 10 tablas principales según lecturas consumidas

```
SELECT SUM(ProvisionedWriteCapacityUnits)
FROM SCHEMA("AWS/DynamoDB", TableName)
GROUP BY TableName
ORDER BY MAX() DESC LIMIT 10
```

Las 10 tablas principales según bytes devueltos

```
SELECT SUM(ReturnedBytes)
FROM SCHEMA("AWS/DynamoDB", TableName)
GROUP BY TableName
ORDER BY MAX() DESC LIMIT 10
```

Las 10 tablas principales según errores de usuario

```
SELECT SUM(UserErrors)
FROM SCHEMA("AWS/DynamoDB", TableName)
GROUP BY TableName
ORDER BY MAX() DESC LIMIT 10
```

Ejemplos de Amazon Elastic Block Store

Los 10 volúmenes principales de Amazon EBS según bytes escritos

```
SELECT SUM(VolumeWriteBytes)
FROM SCHEMA("AWS/EBS", VolumeId)
GROUP BY VolumeId
ORDER BY SUM() DESC
LIMIT 10
```

Tiempo promedio de escritura del volumen de Amazon EBS

```
SELECT AVG(VolumeTotalWriteTime)
FROM SCHEMA("AWS/EBS", VolumeId)
```

Ejemplos de Amazon EC2

Utilización de la CPU de instancias EC2 clasificadas según la más alta

```
SELECT AVG(CPUUtilization)
  FROM SCHEMA("AWS/EC2", InstanceId)
  GROUP BY InstanceId
  ORDER BY AVG() DESC
```

Utilización promedio de la CPU en toda la flota

```
SELECT AVG(CPUUtilization)
FROM SCHEMA("AWS/EC2", InstanceId)
```

Las 10 instancias principales según la utilización más alta de la CPU

```
SELECT MAX(CPUUtilization)
FROM SCHEMA("AWS/EC2", InstanceId)
GROUP BY InstanceId
ORDER BY MAX() DESC
LIMIT 10
```

En este caso, el agente de CloudWatch está recopilando una métrica de **`CPUUtilization`** por aplicación. Esta consulta filtra el promedio de esta métrica para un nombre de aplicación específico.

```
SELECT AVG(CPUUtilization)
FROM "AWS/CWAgent"
WHERE ApplicationName = 'eCommerce'
```

Ejemplos de Amazon Elastic Container Service

Utilización promedio de la CPU en todos los clústeres de ECS

```
SELECT AVG(CPUUtilization)
FROM SCHEMA("AWS/ECS", ClusterName)
```

Los 10 clústeres principales según la utilización de memoria

```
SELECT AVG(MemoryUtilization)
FROM SCHEMA("AWS/ECS", ClusterName)
```

```
GROUP BY ClusterName
ORDER BY AVG() DESC
LIMIT 10
```

Los 10 servicios principales según la utilización de la CPU

```
SELECT AVG(CPUUtilization)
FROM SCHEMA("AWS/ECS", ClusterName, ServiceName)
GROUP BY ClusterName, ServiceName
ORDER BY AVG() DESC
LIMIT 10
```

Los 10 servicios principales según la ejecución de tareas (Información de contenedores)

```
SELECT AVG(RunningTaskCount)
FROM SCHEMA("ECS/ContainerInsights", ClusterName, ServiceName)
GROUP BY ClusterName, ServiceName
ORDER BY AVG() DESC
LIMIT 10
```

Ejemplos de Información de contenedores de Amazon Elastic Kubernetes Service

Utilización promedio de la CPU en todos los clústeres de EKS

```
SELECT AVG(pod_cpu_utilization)
FROM SCHEMA("ContainerInsights", ClusterName)
```

Los 10 clústeres principales según la utilización de CPU de nodo

```
SELECT AVG(node_cpu_utilization)
FROM SCHEMA("ContainerInsights", ClusterName)
GROUP BY ClusterName
ORDER BY AVG() DESC LIMIT 10
```

Los 10 clústeres principales según la utilización de la memoria de pod

```
SELECT AVG(pop_memory_utilization)
FROM SCHEMA("ContainerInsights", ClusterName)
GROUP BY ClusterName
ORDER BY AVG() DESC LIMIT 10
```

Los 10 nodos principales según la utilización de la CPU

```
SELECT AVG(node_cpu_utilization)
FROM SCHEMA("ContainerInsights", ClusterName, NodeName)
GROUP BY ClusterName, NodeName
ORDER BY AVG() DESC LIMIT 10
```

Los 10 pods principales según la utilización de la memoria

```
SELECT AVG(pod_memory_utilization)
FROM SCHEMA("ContainerInsights", ClusterName, PodName)
GROUP BY ClusterName, PodName
ORDER BY AVG() DESC LIMIT 10
```

Ejemplos de EventBridge

Las 10 reglas principales según invocaciones

```
SELECT SUM(Invocations)
FROM SCHEMA("AWS/Events", RuleName)
GROUP BY RuleName
ORDER BY MAX() DESC LIMIT 10
```

Las 10 reglas principales según invocaciones fallidas

```
SELECT SUM(FailedInvocations)
FROM SCHEMA("AWS/Events", RuleName)
GROUP BY RuleName
ORDER BY MAX() DESC LIMIT 10
```

Las 10 reglas principales según reglas coincidentes

```
SELECT SUM(MatchedEvents)
FROM SCHEMA("AWS/Events", RuleName)
GROUP BY RuleName
ORDER BY MAX() DESC LIMIT 10
```

Ejemplos de Kinesis

Las 10 secuencias principales según bytes escritos

```
SELECT SUM("PutRecords.Bytes")
FROM SCHEMA("AWS/Kinesis", StreamName)
GROUP BY StreamName
ORDER BY SUM() DESC LIMIT 10
```

Las 10 secuencias principales según los primeros elementos de la secuencia

```
SELECT MAX("GetRecords.IteratorAgeMilliseconds")
FROM SCHEMA("AWS/Kinesis", StreamName)
GROUP BY StreamName
ORDER BY MAX() DESC LIMIT 10
```

Ejemplos de Lambda

Funciones de Lambda ordenadas por número de invocaciones

```
SELECT SUM(Invocations)
FROM SCHEMA("AWS/Lambda", FunctionName)
GROUP BY FunctionName
ORDER BY SUM() DESC
```

Las 10 funciones principales de Lambda según tiempo de ejecución más largo

```
SELECT AVG(Duration)
FROM SCHEMA("AWS/Lambda", FunctionName)
GROUP BY FunctionName
ORDER BY MAX() DESC
LIMIT 10
```

Las 10 funciones principales de Lambda según recuento de errores

```
SELECT SUM(Errors)
FROM SCHEMA("AWS/Lambda", FunctionName)
GROUP BY FunctionName
ORDER BY SUM() DESC
LIMIT 10
```

Ejemplos de CloudWatch

Los 10 grupos principales de registros según eventos entrantes

```
SELECT SUM(IncomingLogEvents)
FROM SCHEMA("AWS/Logs", LogGroupName)
GROUP BY LogGroupName
ORDER BY SUM() DESC LIMIT 10
```

Los 10 grupos principales de registros según bytes escritos

```
SELECT SUM(IncomingBytes)
FROM SCHEMA("AWS/Logs", LogGroupName)
GROUP BY LogGroupName
ORDER BY SUM() DESC LIMIT 10
```

Ejemplos de Amazon RDS

Las 10 instancias principales de Amazon RDS según la mayor utilización de la CPU

```
SELECT MAX(CPUUtilization)
FROM SCHEMA("AWS/RDS", DBInstanceIdentifier)
GROUP BY DBInstanceIdentifier
ORDER BY MAX() DESC
LIMIT 10
```

Los 10 clústeres principales de Amazon RDS según escrituras

```
SELECT SUM(WriteIOPS)
FROM SCHEMA("AWS/RDS", DBClusterIdentifier)
GROUP BY DBClusterIdentifier
ORDER BY MAX() DESC
LIMIT 10
```

Ejemplos de Amazon Simple Storage Service

Latencia promedio según bucket

```
SELECT AVG(TotalRequestLatency)
FROM SCHEMA("AWS/S3", BucketName, FilterId)
WHERE FilterId = 'EntireBucket'
GROUP BY BucketName
ORDER BY AVG() DESC
```

Los 10 buckets principales según bytes descargados

```
SELECT SUM(BytesDownloaded)
FROM SCHEMA("AWS/S3", BucketName, FilterId)
WHERE FilterId = 'EntireBucket'
GROUP BY BucketName
ORDER BY SUM() DESC
LIMIT 10
```

Ejemplos de Amazon Simple Notification Service

Total de mensajes publicados según temas de SNS

```
SELECT SUM(NumberOfMessagesPublished)
FROM SCHEMA("AWS/SNS", TopicName)
```

Los 10 temas principales según mensajes publicados

```
SELECT SUM(NumberOfMessagesPublished)
FROM SCHEMA("AWS/SNS", TopicName)
GROUP BY TopicName
ORDER BY SUM() DESC
LIMIT 10
```

Los 10 temas principales según fallos en la entrega de mensajes

```
SELECT SUM(NumberOfNotificationsFailed)
FROM SCHEMA("AWS/SNS", TopicName)
GROUP BY TopicName
ORDER BY SUM() DESC
LIMIT 10
```

Ejemplos de Amazon SQS

Las 10 colas principales según el número de mensajes visibles

```
SELECT AVG(ApproximateNumberOfMessagesVisible)
FROM SCHEMA("AWS/SQS", QueueName)
GROUP BY QueueName
ORDER BY AVG() DESC
LIMIT 10
```

Las 10 colas más activas

```
SELECT SUM(NumberOfMessagesSent)
FROM SCHEMA("AWS/SQS", QueueName)
GROUP BY QueueName
ORDER BY SUM() DESC
LIMIT 10
```

Las 10 colas principales según antigüedad del primer mensaje

```
SELECT AVG(ApproximateAgeOfOldestMessage)
FROM SCHEMA("AWS/SQS", QueueName)
GROUP BY QueueName
ORDER BY AVG() DESC
LIMIT 10
```

Límites de Metrics Insights

En la actualidad, CloudWatch Metrics Insights tiene los siguientes límites:

- En la actualidad, solo puede consultar las tres horas de datos más recientes.
- Una única consulta puede procesar un máximo de 10 000 métricas. Esto significa que si las cláusulas SELECT (SELECCIONAR), FROM (DESDE) y WHERE (DONDE) coinciden con más de 10 000 métricas, la consulta solo procesa las primeras 10 000 métricas que encuentra.
- Una única consulta puede devolver un máximo de 500 series temporales. Esto significa que si la consulta devuelve más de 500 métricas, no se devolverán todas las métricas en los resultados de la consulta. Si usa una cláusula de ORDER BY (AGRUPAR POR), luego se ordenan todas las métricas que se procesan y se devuelven las 500 que tienen los valores más altos o más bajos según su cláusula de ORDER BY (AGRUPAR POR).

  Si no incluye una cláusula de ORDER BY (AGRUPAR POR), no puede controlar qué 500 métricas coincidentes se devuelven.
- Puede tener hasta 200 alarmas de Información de métricas por región.
- Metrics Insights no admite datos de alta resolución; esto es, datos de métricas que se han notificado con una granularidad de menos de un minuto. Si solicita datos de alta resolución, la solicitud no devuelve un error, pero el resultado se agrega con una granularidad de un minuto.
- Cada operación de GetMetricData puede tener solo una consulta, pero puede tener varios widgets en un panel y que cada uno incluya una consulta.

Glosario de Metrics Insights

etiqueta

En Metrics Insights, una etiqueta es un par clave-valor que se utiliza para alcanzar una consulta para devolver un conjunto determinado de datos o para definir criterios según los cuales los resultados de la consulta se deben separar en series temporales independientes. Una clave de etiqueta es similar al nombre de una columna en SQL. Actualmente, las etiquetas deben ser dimensiones métricas de CloudWatch.

observación

Una observación es un valor registrado para una métrica determinada en un momento dado.

Solución de problemas de Metrics Insights

Los resultados incluyen “Otro”, pero no lo tengo como dimensión

Esto significa que la consulta incluye una cláusula de GROUP BY que especifica una clave de etiqueta que no se utiliza en algunas de las métricas devueltas por la consulta. En este caso, se devuelve un grupo nulo denominado `Other` (Otro). Las métricas que no incluyen esa clave de etiqueta probablemente sean métricas acumuladas que devuelven valores acumulados en todos los valores de esa clave de etiqueta.

Por ejemplo, supongamos que tenemos la siguiente consulta:

```
SELECT AVG(Faults)
FROM MyCustomNamespace
GROUP BY Operation, ServiceName
```

Si algunas de las métricas devueltas no incluyen `ServiceName` como dimensión, entonces esas métricas se muestran con el valor `Other` para `ServiceName`.

Para evitar ver “Other” (Otro) en los resultados, use SCHEMA (ESQUEMA) en la cláusula de FROM (DESDE), como en el siguiente ejemplo:

```
SELECT AVG(Faults)
FROM SCHEMA(MyCustomNamespace, Operation)
GROUP BY Operation, ServiceName
```

Esto limita los resultados devueltos solo a las métricas que tienen ambas dimensiones: `Operation` y `ServiceName`.

La marca temporal más antigua de mi gráfica tiene un valor métrico inferior al de las demás

CloudWatch Metrics Insights actualmente solo admite las últimas tres horas de datos. Cuando se realiza un gráfico con un periodo superior a un minuto, puede haber casos en que el punto de datos más antiguo difiera del valor esperado. Esto se debe a que las consultas de Metrics Insights solo devuelven los datos de las últimas 3 horas. En este caso, el punto de datos más antiguo de la consulta devuelve solo las observaciones que se midieron dentro del límite de las últimas tres horas, en lugar de devolver todas las observaciones dentro del período de ese punto de datos.

Uso del explorador de métricas para monitorear los recursos según sus etiquetas y propiedades

El explorador de métricas es una herramienta basada en etiquetas que le permite filtrar, agregar y visualizar las métricas por etiquetas y propiedades de recurso para mejorar la observabilidad de los servicios. Esto le proporciona una experiencia de solución de problemas flexible y dinámica, de modo que puede crear varios gráficos a la vez y utilizarlos para crear paneles con el estado de las aplicaciones.

Las visualizaciones del explorador de métricas son dinámicas, por lo que si se crea un recurso concordante después de crear un widget de explorador de métricas y agregarlo a un panel de CloudWatch, el nuevo recurso aparecerá automáticamente en el widget del explorador.

Por ejemplo, si todas las instancias de producción de EC2 tienen la pestaña **`production`**, puede usar el explorador de métricas para filtrar y agregar métricas de todas estas instancias para comprender el estado y rendimiento. Si posteriormente se crea una nueva instancia con una etiqueta coincidente, se agrega al widget del explorador de métricas de manera automática.

> **Note**
>
> El explorador de métricas proporciona una experiencia a un momento dado. Los recursos que se han cancelado o que ya no existen con la propiedad o etiqueta que especificó no se muestran en la visualización. Sin embargo, aún puede encontrar las métricas de estos recursos en las vistas de métricas de CloudWatch.

Con el explorador de métricas, puede elegir cómo se agregan las métricas de los recursos que concuerden con los criterios, y si desea mostrarlas todas en un solo gráfico o en gráficos diferentes dentro de un widget de explorador de métricas.

El explorador de métricas incluye plantillas que puede utilizar para ver gráficos de visualización útiles con un solo clic, y también puede ampliar estas plantillas para crear widgets de explorador de métricas completamente personalizados.

El explorador de métricas admite métricas emitidas por AWS y las métricas de EC2 publicadas por el agente de CloudWatch, incluidas las métricas de memoria, disco y CPU. Para utilizar el explorador de métricas para ver las métricas publicadas por el agente de CloudWatch, es posible que tenga que actualizar el archivo de configuración del agente de CloudWatch. Para obtener más información, consulte Configuración del agente de CloudWatch para el explorador de métricas.

Para crear una visualización con el explorador de métricas y, opcionalmente, agregarla a un panel, siga estos pasos.

Para crear una visualización con el explorador de métricas

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, seleccione Explorer (Explorador).
3. Haga una de las siguientes acciones:
   - Para usar una plantilla, selecciónela en el cuadro que en este momento muestra Empty Explorer (Explorador vacío).

     De acuerdo a cuál sea la plantilla, el explorador puede mostrar inmediatamente gráficos de métricas. Si no lo hace, elija una o más etiquetas o propiedades en el cuaro From (Desde) y, a continuación, los datos deben aparecer. Si no es así, utilice las opciones ubicadas en la parte superior de la página para mostrar un intervalo de tiempo más largo en los gráficos.
   - Para crear una visualización personalizada, bajo el título Metrics (Métricas), elija una métrica o todas las métricas disponibles de un servicio.

     Después de elegir una métrica, puede optar por repetir este paso para agregar más métricas.
4. Para cada métrica seleccionada, CloudWatch muestra la estadística que utilizará inmediatamente después del nombre de métrica. Para cambiarlo, elija el nombre de la estadística y, luego, elija la estadística que desee.
5. En From (Desde), elija una etiqueta o una propiedad de recurso para filtrar los resultados.

Después de hacer esto, puede optar por repetir este paso para elegir más etiquetas o propiedades de recurso.

Si elige varios valores de la misma propiedad, como dos tipos de instancias de EC2, el explorador mostrará todos los recursos que concuerden con cualquiera de las propiedades elegidas. Se trata como a una operación lógica OR.

Si elige diferentes propiedades o etiquetas, como la etiqueta **`Production`** y el tipo de instancia M5, sólo se mostrarán los recursos que concuerden con todas estas selecciones. Se trata como a una operación lógica AND.

6. (Opcional) Para Aggregate by (Agregar por), elija una estadística para agregar las métricas. Luego, junto a for (Para), elija cómo agregar la métrica de la lista. Puede agregar todos los recursos que se muestran actualmente o agregarlos mediante una sola etiqueta o propiedad de recurso.

   De acuerdo al modo que elija para agregar, el resultado puede ser una sola serie temporal o varias series temporales.
7. Bajo el título Split by (Dividir por), puede elegir dividir un único gráfico con varias series temporales en diferentes gráficos. La división se puede hacer por una variedad de criterios que usted elige enSplit by (Dividir por).
8. En Graph options (Opciones de gráficos), puede refinar el gráfico al cambiar el periodo, el tipo de gráfico, la ubicación de la leyenda y el diseño.
9. Para agregar esta visualización como widget a un panel de CloudWatch, elija Add to dashboard (Añadir al panel).

Configuración del agente de CloudWatch para el explorador de métricas

Para permitir que el explorador de métricas descubra métricas de EC2 que el agente de CloudWatch publica, asegúrese de que el archivo de configuración del agente de CloudWatch contenga los siguientes valores:

- En la sección `metrics`, asegúrese de que el parámetro `aggregation_dimensions` incluya `["InstanceId"]`. También puede contener otras dimensiones.
- En la sección `metrics`, asegúrese de que el parámetro `append_dimensions` incluya una línea `{"InstanceId":"${aws:InstanceId}"}`. También puede contener otras líneas.

- En la sección `metrics`, dentro de la sección `metrics_collected`, verifique las secciones de cada tipo de recurso que desea que el explorador de métricas descubra, como las secciones `cpu`, `disk`, y `memory`. Asegúrese de que cada una de estas secciones tenga `"resources": [ "*"] line.`.
- En la sección `cpu` de la sección `metrics_collected`, asegúrese de que haya una línea `"totalcpu": true`.
- Para las métricas recopiladas por el agente de CloudWatch, debe utilizar el espacio de nombres predeterminado `CWAgent` en lugar de un espacio de nombres personalizado.

Las configuraciones de la lista anterior hacen que el agente de CloudWatch publique métricas agregadas para discos, CPU y otros recursos que se pueden trazar en el explorador de métricas para todas las instancias que lo utilizan.

Estas configuraciones volverán a publicar las métricas que había configurado previamente para que se publicaran con varias dimensiones, lo que aumentaría los costos de la métrica.

Para obtener más información sobre cómo se crea el archivo de configuración del agente de CloudWatch, consulte Cree o edite de forma manual el archivo de configuración del agente de CloudWatch.

Uso de flujos métricos

Puede utilizar flujos métricos para transmitir continuamente las métricas de CloudWatch a un destino de su elección, con entrega casi en tiempo real y latencia baja. Los destinos admitidos incluyen destinos de AWS como Amazon Simple Storage Service y varios destinos de proveedores de servicios de terceros.

Existen tres situaciones de uso principales para los flujos métricos de CloudWatch:

- Configuración personalizada con Firehose: cree un flujo métrico y diríjalo a un flujo de entrega de Amazon Data Firehose que entregue sus métricas de CloudWatch a donde usted quiera que vayan. Puede transmitirlas a un lago de datos, como Amazon S3, o a cualquier destino o punto de conexión compatible con Firehose, incluidos los proveedores de terceros. Los formatos JSON, OpenTelemetry 1.0.0 y OpenTelemetry 0.7.0 son compatibles de forma nativa, o puede configurar transformaciones en el flujo de entrega de Firehose para convertir los datos a un formato diferente, como Parquet. Esto permite actualizar continuamente los datos de supervisión o combinar estos datos de métrica de CloudWatch con datos de facturación y rendimiento para crear conjuntos de

datos abundantes. A continuación, puede utilizar herramientas como Amazon Athena para obtener información acerca de la optimización de costos, el rendimiento de los recursos y la utilización de los recursos.

- Configuración rápida de S3: transmisión a Amazon Simple Storage Service mediante un proceso de configuración rápido. De forma predeterminada, CloudWatch crea los recursos necesarios para la transmisión. Los formatos admitidos son JSON, OpenTelemetry 1.0.0 y OpenTelemetry 0.7.0.
- Configuración rápida de socios de AWS: CloudWatch ofrece una experiencia de configuración rápida para algunos socios externos. Puede utilizar proveedores de servicios de terceros para supervisar, solucionar problemas y analizar las aplicaciones mediante los datos transmitidos de CloudWatch. Cuando utiliza el flujo de trabajo de configuración rápida para socios, solo tiene que proporcionar una URL de destino y una clave de API para su destino, y CloudWatch se encarga del resto de la configuración. La configuración rápida de socios está disponible para los siguientes proveedores externos:

  - Datadog
  - Dynatrace
  - New Relic
  - Splunk Observability Cloud
  - SumoLogic

Puede transmitir todas sus métricas de CloudWatch o usar filtros para transmitir solo las métricas especificadas. Cada flujo métrico puede incluir hasta 1000 filtros que incluyen o excluyen espacios de nombres de métricas o métricas específicas. Un único flujo métrico puede solo incluir o excluir filtros, pero no ambas opciones.

Después de crear un flujo métrico, si se crean nuevas métricas que coinciden con los filtros existentes, las nuevas métricas se incluyen automáticamente en el flujo.

No hay límite en el número de flujos métricos por cuenta o por Región, ni límite en el número de actualizaciones de métricas que se estén transmitiendo.

Cada flujo puede usar el formato JSON, OpenTelemetry 1.0.0 u OpenTelemetry 0.7.0. Puede editar el formato de salida de un flujo métrico en cualquier momento, por ejemplo, para actualizar de OpenTelemetry 0.7.0 a OpenTelemetry 1.0.0. Para obtener más información acerca de los formatos de salida, consulte Formatos de salida de flujos métricos.

Para los flujos de métricas en las cuentas de monitoreo, puede elegir si desea incluir métricas de las cuentas de origen vinculadas a esa cuenta de monitoreo. Para obtener más información, consulte Observabilidad entre cuentas de CloudWatch.

Las secuencias métricas siempre incluyen las estadísticas `Minimum`, `Maximum`, `SampleCount` y `Sum`. También puede elegir incluir estadísticas adicionales por un cargo adicional. Para obtener más información, consulte Estadísticas que se pueden transmitir en streaming.

El precio de los flujos métricos se basa en el número de actualizaciones de métrica. También incurre en cargos de Firehose por el flujo de entrega que se utilice para el flujo métrico. Para obtener más información, consulte Precios de Amazon CloudWatch.

Temas

Configuración de un flujo métrico

Siga los pasos descritos en las siguientes secciones para configurar flujos métricos de CloudWatch.

Después de crear un flujo métrico, el tiempo que tarden en aparecer los datos de métrica en el destino depende de la configuración de almacenamiento en búfer en el flujo de entrega de Firehose. El almacenamiento en búfer se expresa en tamaño máximo de carga o en el tiempo de espera máximo, lo que se alcance primero. Si se establecen en los valores mínimos (60 segundos, 1 MB), la latencia esperada es en 3 minutos si los espacios de nombres de CloudWatch seleccionados tienen activadas las actualizaciones de métrica.

En un flujo métrico de CloudWatch, los datos se envían cada un minuto. Los datos pueden llegar al destino final fuera de servicio. Todas las métricas en los espacios de nombres especificados se envían en el flujo métrico, excepto las métricas con una marca de tiempo de más de dos días de antigüedad.

Para cada combinación de nombre de métrica y espacio de nombres que transmite, se distribuyen en streaming todas las combinaciones de dimensiones de ese nombre de métrica y espacio de nombres.

Para los flujos de métricas en las cuentas de monitoreo, puede elegir si desea incluir métricas de las cuentas de origen vinculadas a esa cuenta de monitoreo. Para obtener más información, consulte Observabilidad entre cuentas de CloudWatch.

Para crear y administrar un flujo métrico, debe haber iniciado sesión en una cuenta que tenga la política CloudWatchFullAccess y el permiso `iam:PassRole`, o en una cuenta que tenga la siguiente lista de permisos:

- `iam:PassRole`
- `cloudwatch:PutMetricStream`
- `cloudwatch:DeleteMetricStream`
- `cloudwatch:GetMetricStream`
- `cloudwatch:ListMetricStreams`
- `cloudwatch:StartMetricStreams`
- `cloudwatch:StopMetricStreams`

Si va a hacer que CloudWatch configure el rol de IAM necesario para los flujos métricos, también debe tener los permisos `iam:CreateRole` y `iam:PutRolePolicy`.

> **Important**
>
> Un usuario con `cloudwatch:PutMetricStream` tiene acceso a los datos de métrica de CloudWatch que se están transmitiendo, incluso si no tiene el permiso `cloudwatch:GetMetricData`.

Temas

- Configuración personalizada con Firehose
- Utilice la configuración rápida de Amazon S3
- Configuración rápida de socios

Configuración personalizada con Firehose

Utilice este método para crear un flujo métrico y diríjalo a un flujo de entrega de Amazon Data Firehose que entregue sus métricas de CloudWatch a donde usted quiera que vayan. Puede transmitirlas a un lago de datos, como Amazon S3, o a cualquier destino o punto de conexión compatible con Firehose, incluidos los proveedores de terceros.

Los formatos JSON, OpenTelemetry 1.0.0 y OpenTelemetry 0.7.0 son compatibles de forma nativa, o puede configurar transformaciones en el flujo de entrega de Firehose para convertir los datos a un formato diferente, como Parquet. Esto permite actualizar continuamente los datos de supervisión o combinar estos datos de métrica de CloudWatch con datos de facturación y rendimiento para crear conjuntos de datos abundantes. A continuación, puede utilizar herramientas como Amazon Athena para obtener información acerca de la optimización de costos, el rendimiento de los recursos y la utilización de los recursos.

Puede usar la consola de CloudWatch, la AWS CLI, el AWS CloudFormation, o el AWS Cloud Development Kit (AWS CDK) para configurar un flujo métrico.

El flujo de entrega de Firehose que utilice para el flujo métrico debe estar en la misma cuenta y región en la que configuró el flujo métrico. Para lograr la funcionalidad entre regiones, puede configurar el flujo de entrega de Firehose para que transmita a un destino final que se encuentre en una cuenta diferente o región diferente.

Consola de CloudWatch

En esta sección, se describe cómo utilizar la consola de CloudWatch para configurar un flujo métrico mediante Firehose.

Para configurar un flujo métrico personalizado mediante Firehose

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. Elija Metrics (Métricas), Streams (Flujos) en el panel de navegación. Luego, elija Create metric stream (Crear flujo métrico).
3. (Opcional) Si inicia sesión en una cuenta configurada como cuenta de monitoreo para la observabilidad entre cuentas de CloudWatch, puede optar por incluir las métricas desde una cuenta de origen vinculada en este flujo de métricas. Para incluir las métricas de las cuentas de origen, seleccione Incluir métricas de las cuentas de origen.
4. Elija Configuración personalizada con Firehose.

5. En Seleccionar el flujo de entrega de Kinesis Data Firehose, seleccione el flujo de entrega de Firehose que desea utilizar. Debe estar en la misma cuenta. El formato predeterminado de esta opción es OpenTelemetry 0.7.0, pero puede cambiar el formato más adelante en este procedimiento.

   A continuación, seleccione el flujo de entrega de Firehose que desee utilizar en Seleccionar el flujo de entrega de Firehose.
6. (Opcional) Puede elegir Seleccionar un rol de servicio existente para utilizar un rol de IAM existente en lugar de hacer que CloudWatch cree uno nuevo.
7. (Opcional) Para cambiar el formato de salida predeterminado para su situación, elija Change output format (Cambiar formato de salida). Los formatos admitidos son JSON, OpenTelemetry 1.0.0 y OpenTelemetry 0.7.0.
8. En Métricas para el flujo, seleccione Todas las métricas o Seleccionar métricas.

   Si elige Todas las métricas, todas las métricas de esta cuenta se incluirán en el flujo.

   Considere con cuidado si desea transmitir todas las métricas, ya que cuantas más métricas transmita, mayores serán los cargos por transmisión de métricas.

   Si elige Seleccionar métrica, realice una de las operaciones siguientes:

   - Para transmitir la mayoría de los espacios de nombres de métricas, elija Excluir y seleccione los espacios de nombres o las métricas que desea excluir. Al especificar un espacio de nombres en Excluir, de forma opcional, puede seleccionar algunas métricas específicas de ese espacio de nombres para excluirlas. Si elige excluir un espacio de nombres, pero no selecciona las métricas de ese espacio de nombres, se excluyen todas las métricas de ese espacio de nombres.
   - Para incluir solo algunos espacios de nombres de métricas o métricas en el flujo de métricas, elija Incluir y, a continuación, seleccione los espacios de nombres que desea incluir. Si elige incluir un espacio de nombres, pero no selecciona las métricas de ese espacio de nombres, se incluyen todas las métricas de ese espacio de nombres.
9. (Opcional) Para transmitir estadísticas adicionales para algunas de estas métricas más allá de mínimo, máximo, recuento de muestras y suma, elija Agregar estadísticas adicionales. Elija Add recommended metrics (Agregar métricas recomendadas) para agregar algunas estadísticas de uso común o seleccionar manualmente el espacio de nombres y el nombre de métrica para transmitir estadísticas adicionales. A continuación, seleccione las estadísticas adicionales que desea transmitir.

Para elegir otro grupo de métricas para transmitir un conjunto diferente de estadísticas adicionales, elija Add additional statistics (Agregar una estadística adicional). Cada métrica puede incluir hasta 20 estadísticas adicionales y hasta 100 métricas dentro de un flujo de métricas pueden incluir estadísticas adicionales.

El streaming de estadísticas adicionales conlleva más cargos. Para obtener más información, consulte Estadísticas que se pueden transmitir en streaming.

Para obtener definiciones de las estadísticas adicionales, consulte Definiciones de estadísticas de CloudWatch.

10. (Opcional) Personalice el nombre del nuevo flujo métrico en Metric stream name (Nombre de flujo métrico).
11. Elija Create metric stream (Crear flujo métrico).

La AWS CLI o la API de AWS

Siga los pasos a continuación, para crear un flujo métrico de CloudWatch.

Para utilizar la AWS CLI o la API de AWS para crear un flujo métrico

1. Si está transmitiendo a Amazon S3, primero cree el bucket. Para obtener más información, consulte Creating a bucket (Creación de un bucket).
2. Cree un flujo de entrega de Firehose. Para obtener más información, consulte Creación de un flujo de Firehose.
3. Cree un rol de IAM que permita a CloudWatch escribir en el flujo de entrega de Firehose. Para obtener más información acerca del contenido de este rol, consulte Confianza entre CloudWatch y Firehose.
4. Use el comando CLI `aws cloudwatch put-metric-stream` o el la API `PutMetricStream` para crear el flujo métrico de CloudWatch.

AWS CloudFormation

Puede usar el AWS CloudFormation para configurar un flujo métrico. Para obtener más información, consulte AWS::CloudWatch::MetricStream

Para utilizar el AWS CloudFormation para crear un flujo métrico

1. Si está transmitiendo a Amazon S3, primero cree el bucket. Para obtener más información, consulte Creating a bucket (Creación de un bucket).
2. Cree un flujo de entrega de Firehose. Para obtener más información, consulte Creación de un flujo de Firehose.
3. Cree un rol de IAM que permita a CloudWatch escribir en el flujo de entrega de Firehose. Para obtener más información acerca del contenido de este rol, consulte Confianza entre CloudWatch y Firehose.
4. Cree la secuencia en AWS CloudFormation. Para obtener más información, consulte AWS::CloudWatch::MetricStream.

AWS Cloud Development Kit (AWS CDK)

Puede usar el AWS Cloud Development Kit (AWS CDK) para configurar un flujo métrico.

Para utilizar el AWS CDK para crear un flujo métrico

1. Si está transmitiendo a Amazon S3, primero cree el bucket. Para obtener más información, consulte Creating a bucket (Creación de un bucket).
2. Cree un flujo de entrega de Firehose. Para obtener más información, consulte Creación de un flujo de entrega de Amazon Data Firehose.
3. Cree un rol de IAM que permita a CloudWatch escribir en el flujo de entrega de Firehose. Para obtener más información acerca del contenido de este rol, consulte Confianza entre CloudWatch y Firehose.
4. Cree el flujo métrico. El recurso del flujo métrico está disponible en AWS CDK como un modelo nivel 1 (L1) llamado `CfnMetricStream`. Para obtener más información, consulte Using L1 constructs (Uso de modelos L1).

Utilice la configuración rápida de Amazon S3

El método Configuración rápida de S3 funciona bien si desea configurar rápidamente un flujo en Amazon S3 y no necesita ninguna transformación de formato más allá de los formatos JSON, OpenTelemetry 1.0.0 y OpenTelemetry 0.7.0 admitidos. CloudWatch creará todos los recursos necesarios, incluidos el flujo de entrega de Firehose y los roles de IAM. El formato predeterminado de esta opción es JSON, pero puede cambiar el formato mientras configura la transmisión.

Como opción, si desea que el formato final sea formato Parquet o el formato Optimized Row Columnar (ORC), debe seguir los pasos descritos en Configuración personalizada con Firehose.

Consola de CloudWatch

En esta sección se describe cómo utilizar la consola de CloudWatch para configurar un flujo métrico de Amazon S3 mediante la configuración rápida de S3.

Para configurar un flujo métrico mediante la configuración rápida de S3

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. Elija Metrics (Métricas), Streams (Flujos) en el panel de navegación. Luego, elija Create metric stream (Crear flujo métrico).
3. (Opcional) Si inicia sesión en una cuenta configurada como cuenta de monitoreo para la observabilidad entre cuentas de CloudWatch, puede optar por incluir las métricas desde una cuenta de origen vinculada en este flujo de métricas. Para incluir las métricas de las cuentas de origen, seleccione Incluir métricas de las cuentas de origen.
4. Elija la Configuración rápida de S3. CloudWatch creará todos los recursos necesarios, incluidos el flujo de entrega de Firehose y los roles de IAM. El formato predeterminado de esta opción es JSON, pero puede cambiar el formato más adelante en este procedimiento.
5. (Opcional) Elija Seleccionar recursos existentes para usar un bucket de S3 existente o los roles de IAM existentes en lugar de hacer que CloudWatch cree otros nuevos para usted.
6. (Opcional) Para cambiar el formato de salida predeterminado para su situación, elija Change output format (Cambiar formato de salida). Los formatos admitidos son JSON, OpenTelemetry 1.0.0 y OpenTelemetry 0.7.0.
7. En Métricas para el flujo, seleccione Todas las métricas o Seleccionar métricas.

   Si elige Todas las métricas, todas las métricas de esta cuenta se incluirán en el flujo.

   Considere con cuidado si desea transmitir todas las métricas, ya que cuantas más métricas transmita, mayores serán los cargos por transmisión de métricas.

   Si elige Seleccionar métrica, realice una de las operaciones siguientes:

   - Para transmitir la mayoría de los espacios de nombres de métricas, elija Excluir y seleccione los espacios de nombres o las métricas que desea excluir. Al especificar un espacio de nombres en Excluir, de forma opcional, puede seleccionar algunas métricas específicas de ese espacio de nombres para excluirlas. Si elige excluir un espacio de nombres, pero no

selecciona las métricas de ese espacio de nombres, se excluyen todas las métricas de ese espacio de nombres.

- Para incluir solo algunos espacios de nombres de métricas o métricas en el flujo de métricas, elija Incluir y, a continuación, seleccione los espacios de nombres que desea incluir. Si elige incluir un espacio de nombres, pero no selecciona las métricas de ese espacio de nombres, se incluyen todas las métricas de ese espacio de nombres.

8. (Opcional) Para transmitir estadísticas adicionales para algunas de estas métricas más allá de mínimo, máximo, recuento de muestras y suma, elija Agregar estadísticas adicionales. Elija Add recommended metrics (Agregar métricas recomendadas) para agregar algunas estadísticas de uso común o seleccionar manualmente el espacio de nombres y el nombre de métrica para transmitir estadísticas adicionales. A continuación, seleccione las estadísticas adicionales que desea transmitir.

   Para elegir otro grupo de métricas para transmitir un conjunto diferente de estadísticas adicionales, elija Add additional statistics (Agregar una estadística adicional). Cada métrica puede incluir hasta 20 estadísticas adicionales y hasta 100 métricas dentro de un flujo de métricas pueden incluir estadísticas adicionales.

   El streaming de estadísticas adicionales conlleva más cargos. Para obtener más información, consulte Estadísticas que se pueden transmitir en streaming.

   Para obtener definiciones de las estadísticas adicionales, consulte Definiciones de estadísticas de CloudWatch.

9. (Opcional) Personalice el nombre del nuevo flujo métrico en Metric stream name (Nombre de flujo métrico).
10. Elija Create metric stream (Crear flujo métrico).

Configuración rápida de socios

CloudWatch proporciona una experiencia de configuración rápida para los siguientes socios externos. Para usar este flujo de trabajo, solo debe proporcionar una URL de destino y una clave de API para su destino. CloudWatch se encarga del resto de la configuración, incluida la creación del flujo de entrega de Firehose y los roles de IAM necesarios.

> **Important**
>
> Antes de utilizar la configuración rápida de socios para crear un flujo métrico, recomendamos encarecidamente leer la documentación de ese socio, cuyo enlace se incluye en la siguiente lista.

- Datadog
- Dynatrace
- New Relic
- Splunk Observability Cloud
- SumoLogic

Cuando configura un flujo métrico para uno de estos socios, este se crea con algunos ajustes predeterminados, como se indica en las siguientes secciones.

Temas

Configurar un flujo métrico mediante la configuración rápida de socios

CloudWatch ofrece una opción de configuración rápida para algunos socios externos. Antes de comenzar con los pasos en esta sección, debe disponer de cierta información para el socio. Esta información puede incluir una URL de destino o una clave de API para el destino de su socio. También debe leer la documentación en el sitio web del socio vinculado en la sección anterior y los valores predeterminados para ese socio que aparecen en las siguientes secciones.

Para transmitir a un destino de terceros que no sea compatible con la configuración rápida, puede seguir las instrucciones en Siga las instrucciones en Configuración personalizada con Firehose para

configurar un flujo con Firehose y, a continuación, enviar esas métricas desde Firehose su destino final.

Para utilizar la configuración rápida de socios para crear un flujo métrico para un proveedor externo

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. Elija Metrics (Métricas), Streams (Flujos) en el panel de navegación. Luego, elija Create metric stream (Crear flujo métrico).
3. (Opcional) Si inicia sesión en una cuenta configurada como cuenta de monitoreo para la observabilidad entre cuentas de CloudWatch, puede optar por incluir las métricas desde una cuenta de origen vinculada en este flujo de métricas. Para incluir las métricas de las cuentas de origen, seleccione Incluir métricas de las cuentas de origen.
4. Elija Configuración rápida para socios de Amazon Web Services
5. Seleccione el nombre del socio al que desea transmitir las métricas.
6. En la URL del punto de conexión, introduzca la URL de destino.
7. Para la clave de acceso o la clave de API, introduzca la clave de acceso del socio. No todos los socios requieren una clave de acceso.
8. En Métricas para el flujo, seleccione Todas las métricas o Seleccionar métricas.

   Si elige Todas las métricas, todas las métricas de esta cuenta se incluirán en el flujo.

   Considere con cuidado si desea transmitir todas las métricas, ya que cuantas más métricas transmita, mayores serán los cargos por transmisión de métricas.

   Si elige Seleccionar métrica, realice una de las operaciones siguientes:

   - Para transmitir la mayoría de los espacios de nombres de métricas, elija Excluir y seleccione los espacios de nombres o las métricas que desea excluir. Al especificar un espacio de nombres en Excluir, de forma opcional, puede seleccionar algunas métricas específicas de ese espacio de nombres para excluirlas. Si elige excluir un espacio de nombres, pero no selecciona las métricas de ese espacio de nombres, se excluyen todas las métricas de ese espacio de nombres.
   - Para incluir solo algunos espacios de nombres de métricas o métricas en el flujo de métricas, elija Incluir y, a continuación, seleccione los espacios de nombres que desea incluir. Si elige incluir un espacio de nombres, pero no selecciona las métricas de ese espacio de nombres, se incluyen todas las métricas de ese espacio de nombres.

9. (Opcional) Para transmitir estadísticas adicionales para algunas de estas métricas más allá de mínimo, máximo, recuento de muestras y suma, elija Agregar estadísticas adicionales. Elija Add recommended metrics (Agregar métricas recomendadas) para agregar algunas estadísticas de uso común o seleccionar manualmente el espacio de nombres y el nombre de métrica para transmitir estadísticas adicionales. A continuación, seleccione las estadísticas adicionales que desea transmitir.

   Para elegir otro grupo de métricas para transmitir un conjunto diferente de estadísticas adicionales, elija Add additional statistics (Agregar una estadística adicional). Cada métrica puede incluir hasta 20 estadísticas adicionales y hasta 100 métricas dentro de un flujo de métricas pueden incluir estadísticas adicionales.

   El streaming de estadísticas adicionales conlleva más cargos. Para obtener más información, consulte Estadísticas que se pueden transmitir en streaming.

   Para obtener definiciones de las estadísticas adicionales, consulte Definiciones de estadísticas de CloudWatch.
10. (Opcional) Personalice el nombre del nuevo flujo métrico en Metric stream name (Nombre de flujo métrico).
11. Elija Create metric stream (Crear flujo métrico).

Valores de flujo predeterminados de Datadog

Los flujos de configuración rápida de socios para Datadog utilizan los siguientes valores predeterminados:

- Formato de salida: OpenTelemetry 0.7.0
- Codificación de contenido del flujo de Firehose GZIP
- Opciones de almacenamiento en búfer del flujo de Firehose Intervalo de 60 segundos, tamaño de 4 MB
- Opción de reintento del flujo de Firehose Duración de 60 segundos

Cuando utiliza la configuración rápida de socios para crear un flujo métrico a Datadog y transmite determinadas métricas, esas métricas incluyen algunas estadísticas adicionales de forma predeterminada. La transmisión de estadísticas adicionales puede conllevar cargos adicionales. Para obtener más información acerca de las estadísticas y sus costos, consulte Estadísticas que se pueden transmitir en streaming.

La siguiente lista muestra las métricas con estadísticas adicionales transmitidas de forma predeterminada, si decide transmitir esas métricas. Puede anular la selección de estas estadísticas adicionales antes de iniciar la transmisión.

- **`Duration`** en **`AWS/Lambda`**: p50, p80, p95, p99, p99.9
- **`PostRuntimeExtensionDuration`** en **`AWS/Lambda`**: p50, p99
- **`FirstByteLatency`** y **`TotalRequestLatency`** en **`AWS/S3`**: p50, p90, p95, p99, p99.9
- **`ResponseLatency`** en **`AWS/Polly`** y **`TargetResponseTime`** en AWS/ApplicationELB: p50, p90, p95, p99
- **`Latency`** y **`IntegrationLatency`** en **`AWS/ApiGateway`**: p90, p95, p99
- **`Latency`** y **`TargetResponseTime`** en **`AWS/ELB`**: p95, p99
- **`RequestLatency`** en **`AWS/AppRunner`**: p50, p95, p99
- **`ActivityTime`**, **`ExecutionTime`**, **`LambdaFunctionRunTime`**, **`LambdaFunctionScheduleTime`**, **`LambdaFunctionTime`**, **`ActivityRunTime`** y **`ActivityScheduleTime`** en **`AWS/States`**: p95, p99
- **`EncoderBitRate`**, **`ConfiguredBitRate`** y **`ConfiguredBitRateAvailable`** en **`AWS/MediaLive`**: p90
- **`Latency`** en **`AWS/AppSync`**: p90

Valores de flujo predeterminados de Dynatrace

Los flujos de configuración rápida de socios para Dynatrace utilizan los siguientes valores predeterminados:

- Formato de salida: OpenTelemetry 0.7.0
- Codificación de contenido del flujo de Firehose GZIP
- Opciones de almacenamiento en búfer del flujo de Firehose Intervalo de 60 segundos, tamaño de 5 MB
- Opción de reintento del flujo de Firehose Duración de 600 segundos

Valores predeterminados de flujo de New Relic

Los flujos de configuración rápida de socios para New Relic utilizan los siguientes valores predeterminados:

- Formato de salida: OpenTelemetry 0.7.0
- Codificación de contenido del flujo de Firehose GZIP
- Opciones de almacenamiento en búfer del flujo de Firehose Intervalo de 60 segundos, tamaño de 1 MB
- Opción de reintento del flujo de Firehose Duración de 60 segundos

Valores predeterminados del flujo de Splunk Observability Cloud

Los flujos de configuración rápida de socios para Splunk Observability Cloud utilizan los siguientes valores predeterminados:

- Formato de salida: OpenTelemetry 0.7.0
- Codificación de contenido del flujo de Firehose GZIP
- Opciones de almacenamiento en búfer del flujo de Firehose Intervalo de 60 segundos, tamaño de 1 MB
- Opción de reintento del flujo de Firehose Duración de 300 segundos

Valores predeterminados de flujo de Sumo Logic

Los flujos de configuración rápida de socios para Sumo Logic utilizan los siguientes valores predeterminados:

- Formato de salida: OpenTelemetry 0.7.0
- Codificación de contenido del flujo de Firehose GZIP
- Opciones de almacenamiento en búfer del flujo de Firehose Intervalo de 60 segundos, tamaño de 1 MB
- Opción de reintento del flujo de Firehose Duración de 60 segundos

Estadísticas que se pueden transmitir en streaming

Los flujos métricos siempre incluyen las siguientes estadísticas: `Minimum`, `Maximum`, `SampleCount` y `Sum`. También puede elegir incluir las siguientes estadísticas adicionales en un flujo de métricas. Esta elección se realiza por métricas. Para obtener más información acerca de estas estadísticas, consulte Definiciones de estadísticas de CloudWatch.

- Valores de percentil como p95 o p99 (para flujos con formato JSON o OpenTelemetry)
- Media recortada (solo para flujos con formato JSON)
- Media Winsorized (solo para flujos con formato JSON)
- Recuento recortado (solo para flujos con formato JSON)
- Suma recortada (solo para flujos con formato JSON)
- Clasificación de percentiles (solo para flujos con formato JSON)
- Media intercuartil (solo para flujos con formato JSON)

El streaming de estadísticas adicionales conlleva cargos adicionales. El streaming entre una y cinco de estas estadísticas adicionales para una métrica concreta se factura como actualización de métrica adicional. A partir de entonces, cada conjunto adicional de hasta cinco de estas estadísticas se facturará como otra actualización de métrica.

Por ejemplo, supongamos que para una métrica está transmitiendo las seis estadísticas adicionales siguientes: p95, p99, p99.9, Media recortada, Media Winsorizada y Suma recortada. Cada actualización de esta métrica se factura como tres actualizaciones de métricas: una para la actualización de métricas que incluye las estadísticas predeterminadas, una para las cinco primeras estadísticas adicionales y otra para la sexta estadística adicional. Sumar hasta cuatro estadísticas adicionales para un total de diez no aumentaría la facturación, pero lo haría una undécima estadística adicional.

Cuando especifica un nombre de métrica y una combinación de espacio de nombres para transmitir estadísticas adicionales, todas las combinaciones de dimensiones del nombre de métrica y el espacio de nombres se distribuyen en streaming con las estadísticas adicionales.

Las secuencias métricas de CloudWatch publican una nueva métrica, `TotalMetricUpdate`, que refleja el número base de actualizaciones de métricas más actualizaciones de métricas adicionales en las que se produce el streaming de estadísticas adicionales. Para obtener más información, consulte Supervisión del flujo métrico con las métricas de CloudWatch.

Para obtener más información, consulte Precios de Amazon CloudWatch.

> **Note**
>
> Algunas métricas no admiten percentiles. Las estadísticas de percentiles de estas métricas se excluyen del flujo y no incurren en cargos de flujo métrico. Un ejemplo de estas

estadísticas que no admiten percentiles son algunas métricas del espacio de nombres AWS/ECS.

Las estadísticas adicionales que configura se transmiten en streaming solo si coinciden con los filtros de la transmisión. Por ejemplo, si crea un flujo que solo tiene EC2 y RDS en los filtros de inclusión y, a continuación, las listas de configuración de estadísticas EC2 y Lambda, a continuación, el flujo incluye EC2 métricas con estadísticas adicionales, métricas RDS con solo las estadísticas predeterminadas y no incluye estadísticas Lambda en absoluto.

Operación y mantenimiento del flujo métrico

Los flujos métricos siempre están en uno de los dos estados, Running (En ejecución) o Stopped (Detenido).

- En ejecución: El flujo métrico se está ejecutando correctamente. Es posible que no haya datos de métrica transmitidos al destino debido a los filtros del flujo.
- Detenido: El flujo métrico ha sido detenido explícitamente, y no por un error. Puede ser útil detener la transmisión para pausar temporalmente el streaming de datos sin eliminar el flujo.

Si detiene y reinicia un flujo métrico, los datos de la métrica que se publicaron en CloudWatch mientras se detuvo el flujo métrico no se rellenan en el flujo métrico.

Si cambia el formato de salida de un flujo métrico, en algunos casos podrá ver una pequeña cantidad de datos de la métrica escritos en el destino tanto en el formato antiguo como en el nuevo. Para evitar esta situación, puede crear un nuevo flujo de entrega de Firehose con la misma configuración que la actual y, a continuación, cambiar al nuevo flujo de entrega de Firehose y cambiar el formato de salida al mismo tiempo. De esta forma, los registros Kinesis con diferentes formatos de salida se almacenan en Amazon S3 en objetos separados. Más tarde, puede dirigir el tráfico de vuelta al flujo de entrega original de Firehose y eliminar el segundo flujo de entrega.

Para ver, editar, detener e iniciar los flujos métricos

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. Elija Metrics (Métricas), Streams (Flujos) en el panel de navegación.

   Aparecerá la lista de flujos, y la columna Status (Estado) muestra si cada flujo está en ejecución o detenido.

3. Para detener o iniciar un flujo métrico, seleccione el flujo y elija Stop (Detener) o Start (Comenzar).
4. Para ver los detalles sobre un flujo métrico, seleccione el flujo y elija View details (Ver detalles).
5. Para cambiar el formato de salida, los filtros, la secuencia de Firehose de destino o los roles del flujo, elija Editar y realice los cambios que desee.

   Si cambia los filtros, es posible que haya algunos espacios vacíos en los datos de la métrica durante la transición.

Supervisión del flujo métrico con las métricas de CloudWatch

Los flujos métricos emiten métricas de CloudWatch sobre el estado y el funcionamiento en el espacio de nombres `AWS/CloudWatch/MetricStreams`. Se emiten las siguientes métricas. Estas métricas se emiten con una dimensión `MetricStreamName` y sin dimensión. Puede utilizar las métricas sin dimensiones para ver métricas agregadas para todas los flujos métricos. Puede utilizar las métricas con la dimensión `MetricStreamName` para ver las métricas sólo sobre ese flujo métrico.

Para todas estas métricas, los valores se emiten solo para los flujos métricos que se encuentran en el estado Running (En ejecución).

Métrica	Descripción
`MetricUpdate`	Las actualizaciones de métricas numéricas enviadas al flujo métrico. Si no se transmite ninguna actualización de métrica durante un período de tiempo, esta métrica no se emite durante ese período de tiempo. Si detiene el flujo métrico, esta métrica deja de emitirse hasta que se inicie de nuevo el flujo métrico. Estadística válida: `Sum` Unidades: ninguna
`TotalMetricUpdate`	Esto se calcula como MetricUpdate + un número basado en estadísticas adicionales que se están transmitiendo. Para cada combinación única de nombres de espacio de nombres y métricas, el streaming de 1 a 5 estadísticas adicionales agrega 1 a la

Métrica	Descripción
	`TotalMetricUpdate` , estadísticas adicionales streaming 6-10 agrega 2 a `TotalMetricUpdate` , y así sucesivamente. Estadística válida: `Sum` Unidades: ninguna
`PublishErrorRate`	El número de errores irrecuperables que se producen al introducir datos en el flujo de entrega de Firehose. Si no se producen errores durante un periodo de tiempo, esta métrica no se emite durante ese periodo de tiempo. Si detiene el flujo métrico, esta métrica deja de emitirse hasta que se inicie de nuevo el flujo métrico. Estadísticas válidas: `Average` para ver la tasa de actualizaciones de métricas que no se pueden registrar. Este valor debe estar entre 0,0 and 1,0 Unidades: ninguna

Confianza entre CloudWatch y Firehose

El flujo de entrega de Firehose debe confiar en CloudWatch a través de un rol de IAM que tenga permisos de escritura para Firehose. Estos permisos se pueden limitar al único flujo de entrega de Firehose que utiliza el flujo métrico de CloudWatch. El Rol de IAM debe confiar en el servicio principal `streams.metrics.cloudwatch.amazonaws.com`.

Si utiliza la consola de CloudWatch para crear un flujo métrico, puede hacer que CloudWatch cree el rol con los permisos que corresponden. Si utiliza otro método para crear un flujo métrico o desea crear el Rol de IAM en sí, debe contener la siguiente política de permisos y la política de confianza.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "firehose:PutRecord",
```

```
                "firehose:PutRecordBatch"
            ],
            "Effect": "Allow",
            "Resource": "arn:aws:firehose:region:account-id:deliverystream/*"
        }
    ]
}
```

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "streams.metrics.cloudwatch.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

CloudWatch transmite datos de métrica al flujo de entrega de Firehose de destino en nombre de la fuente que posee el recurso del flujo métrico.

Formatos de salida de flujos métricos

Los datos de un flujo métrico de CloudWatch pueden estar en formato JSON o en formato OpenTelemetry. Actualmente, se admiten los formatos OpenTelemetry 1.0.0 y 0.7.0.

Contenido

Formato JSON

En un flujo métrico de CloudWatch que utiliza el formato JSON, cada registro de Firehose contiene varios objetos JSON separados por un carácter de línea nueva (\n). Cada objeto incluye un único punto de datos de una sola métrica.

El formato JSON que se utiliza es totalmente compatible con AWS Glue y con Amazon Athena. Si tiene un flujo de entrega de Firehose y una tabla de AWS Glue formateada correctamente, el formato se puede transformar de manera automática en formato Parquet u Optimized Row Columnar (ORC) antes de almacenarse en S3. Para obtener más información acerca de cómo transformar el formato, consulte Cómo convertir el formato de registro de entrada en Firehose. Para obtener más información acerca del formato correcto para AWS Glue, consulte ¿Qué esquema de AWS Glue debería usar para obtener el formato de salida JSON?.

En el formato JSON, los valores válidos para `unit` son iguales que para el valor de `unit` en la estructura de la API `MetricDatum`. Para obtener más información, consulte MetricDatum. El valor para el campo `timestamp` está en milisegundos de la fecha de inicio, como `1616004674229`.

A continuación, se muestra un ejemplo del formato. En este ejemplo, se formatea JSON para facilitar la lectura, pero en la práctica todo el formato está en una única línea.

```
{
    "metric_stream_name": "MyMetricStream",
    "account_id": "1234567890",
    "region": "us-east-1",
    "namespace": "AWS/EC2",
    "metric_name": "DiskWriteOps",
    "dimensions": {
        "InstanceId": "i-123456789012"
    },
    "timestamp": 1611929698000,
    "value": {
        "count": 3.0,
        "sum": 20.0,
        "max": 18.0,
        "min": 0.0,
        "p99": 17.56,
        "p99.9": 17.8764,
        "TM(25%:75%)": 16.43
    },
    "unit": "Seconds"
```

```
}
```

¿Qué esquema de AWS Glue debería usar para obtener el formato de salida JSON?

A continuación, se muestra un ejemplo de una representación JSON del `StorageDescriptor` para una tabla de AWS Glue, que luego sería utilizado por Firehose. Para obtener más información acerca de `StorageDescriptor`, consulte StorageDescriptor.

```
{
  "Columns": [
    {
      "Name": "metric_stream_name",
      "Type": "string"
    },
    {
      "Name": "account_id",
      "Type": "string"
    },
    {
      "Name": "region",
      "Type": "string"
    },
    {
      "Name": "namespace",
      "Type": "string"
    },
    {
      "Name": "metric_name",
      "Type": "string"
    },
    {
      "Name": "timestamp",
      "Type": "timestamp"
    },
    {
      "Name": "dimensions",
      "Type": "map<string,string>"
    },
    {
      "Name": "value",
      "Type":
 "struct<min:double,max:double,count:double,sum:double,p99:double,p99.9:double>"
    },
```

```
    {
      "Name": "unit",
      "Type": "string"
    }
  ],
  "Location": "s3://my-s3-bucket/",
  "InputFormat": "org.apache.hadoop.mapred.TextInputFormat",
  "OutputFormat": "org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat",
  "SerdeInfo": {
    "SerializationLibrary": "org.apache.hive.hcatalog.data.JsonSerDe"
  },
  "Parameters": {
    "classification": "json"
  }
}
```

El ejemplo anterior es para los datos registrados en Amazon S3 en formato JSON. Reemplace los valores de los siguientes campos con los valores indicados para almacenar los datos en formato Parquet o en formato Optimized Row Columnar (ORC).

- Parquet:
  - inputFormat: org.apache.hadoop.hive.ql.io.parquet.MapredParquetInputFormat
  - outputFormat: org.apache.hadoop.hive.ql.io.parquet.MapredParquetOutputFormat
  - SerDeInfo.serializationLib: org.apache.hadoop.hive.ql.io.parquet.serde.ParquetHiveSerDe
  - parameters.classification: parquet
- ORC:
  - inputFormat: org.apache.hadoop.hive.ql.io.orc.OrcInputFormat
  - outputFormat: org.apache.hadoop.hive.ql.io.orc.OrcOutputFormat
  - SerDeInfo.serializationLib: org.apache.hadoop.hive.ql.io.orc.OrcSerde
  - parameters.classification: orc

Formato OpenTelemetry 1.0.0

> **Note**
>
> Con el formato OpenTelemetry 1.0.0, los atributos de las métricas se codifican como una lista de objetos `KeyValue` en lugar del tipo `StringKeyValue` utilizado en el formato 0.7.0. Como consumidor, este es el único cambio importante entre los formatos 0.7.0 y 1.0.0.

> Un analizador generado a partir de los archivos proto de la versión 0.7.0 no analizará los atributos de las métricas codificados en el formato 1.0.0. Lo mismo ocurre a la inversa: un analizador generado a partir de los archivos proto de la versión 1.0.0 no analizará los atributos de las métricas codificados en el formato 0.7.0.

OpenTelemetry es una colección de herramientas, API y SDK. Puede utilizarlo para instrumentar, generar, recopilar y exportar datos telemétricos (métricas, registros y seguimientos) para analizarlos. OpenTelemetry es parte de Cloud Native Computing Foundation. Para obtener más información, consulte OpenElemetry.

Para obtener información acerca de la especificación completa de OpenTelemetry 1.0.0, consulte Versión de lanzamiento 1.0.0.

Un registro de Kinesis puede contener una o más estructuras de datos OpenTelemetry `ExportMetricsServiceRequest`. Cada estructura de datos comienza con una cabecera con un `UnsignedVarInt32` que indica la longitud de registro en bytes. Cada `ExportMetricsServiceRequest` puede contener datos de varias métricas a la vez.

A continuación, se muestra una cadena de representación del mensaje de la estructura de datos OpenTelemetry `ExportMetricsServiceRequest`. OpenTelemetry serializa el protocolo binario Google Protocol Buffers y este no es legible para el ser humano.

```
resource_metrics {
  resource {
    attributes {
      key: "cloud.provider"
      value {
        string_value: "aws"
      }
    }
    attributes {
      key: "cloud.account.id"
      value {
        string_value: "123456789012"
      }
    }
    attributes {
      key: "cloud.region"
      value {
        string_value: "us-east-1"
```

```
      }
    }
    attributes {
      key: "aws.exporter.arn"
      value {
        string_value: "arn:aws:cloudwatch:us-east-1:123456789012:metric-stream/
MyMetricStream"
      }
    }
  }
  scope_metrics {
    metrics {
      name: "amazonaws.com/AWS/DynamoDB/ConsumedReadCapacityUnits"
      unit: "NoneTranslated"
      summary {
        data_points {
          start_time_unix_nano: 60000000000
          time_unix_nano: 120000000000
          count: 1
          sum: 1.0
          quantile_values {
            value: 1.0
          }
          quantile_values {
            quantile: 0.95
            value: 1.0
          }
          quantile_values {
            quantile: 0.99
            value: 1.0
          }
          quantile_values {
            quantile: 1.0
            value: 1.0
          }
          attributes {
            key: "Namespace"
            value {
              string_value: "AWS/DynamoDB"
            }
          }
          attributes {
            key: "MetricName"
            value {
```

```
            string_value: "ConsumedReadCapacityUnits"
          }
        }
        attributes {
          key: "Dimensions"
          value {
            kvlist_value {
              values {
                key: "TableName"
                value {
                  string_value: "MyTable"
                }
              }
            }
          }
        }
      }
      data_points {
        start_time_unix_nano: 70000000000
        time_unix_nano: 130000000000
        count: 2
        sum: 5.0
        quantile_values {
          value: 2.0
        }
        quantile_values {
          quantile: 1.0
          value: 3.0
        }
        attributes {
          key: "Namespace"
          value {
            string_value: "AWS/DynamoDB"
          }
        }
        attributes {
          key: "MetricName"
          value {
            string_value: "ConsumedReadCapacityUnits"
          }
        }
        attributes {
          key: "Dimensions"
          value {
```

```
              kvlist_value {
                values {
                  key: "TableName"
                  value {
                    string_value: "MyTable"
                  }
                }
              }
            }
          }
        }
      }
    }
  }
}
```

Objeto de nivel superior para serializar datos de métrica de OpenTelemetry

`ExportMetricsServiceRequest` es el envoltorio de nivel superior para serializar una carga del exportador de OpenTelemetry. Contiene uno o más `ResourceMetrics`.

```
message ExportMetricsServiceRequest {
  // An array of ResourceMetrics.
  // For data coming from a single resource this array will typically contain one
  // element. Intermediary nodes (such as OpenTelemetry Collector) that receive
  // data from multiple origins typically batch the data before forwarding further and
  // in that case this array will contain multiple elements.
  repeated opentelemetry.proto.metrics.v1.ResourceMetrics resource_metrics = 1;
}
```

`ResourceMetrics` es el objeto de nivel superior para representar objetos MetricData.

```
// A collection of ScopeMetrics from a Resource.
message ResourceMetrics {
  reserved 1000;

  // The resource for the metrics in this message.
  // If this field is not set then no resource info is known.
  opentelemetry.proto.resource.v1.Resource resource = 1;

  // A list of metrics that originate from a resource.
  repeated ScopeMetrics scope_metrics = 2;
```

```
  // This schema_url applies to the data in the "resource" field. It does not apply
  // to the data in the "scope_metrics" field which have their own schema_url field.
  string schema_url = 3;
}
```

El objeto recurso

Un objeto `Resource` es un objeto de par de valores que contiene información sobre el recurso que generó las métricas. Para las métricas que AWS crea, la estructura de datos contiene el Nombre de recurso de Amazon (ARN) del recurso relacionado con la métrica, como una instancia EC2 o un bucket de S3.

El objeto `Resource` contiene un atributo llamado `attributes`, que almacena una lista de pares clave-valor.

- `cloud.account.id` contiene el ID de la cuenta
- `cloud.region` contiene la Región
- `aws.exporter.arn` contiene el ARN del flujo métrico
- `cloud.provider` es siempre `aws`.

```
// Resource information.
message Resource {
  // Set of attributes that describe the resource.
  // Attribute keys MUST be unique (it is not allowed to have more than one
  // attribute with the same key).
  repeated opentelemetry.proto.common.v1.KeyValue attributes = 1;

  // dropped_attributes_count is the number of dropped attributes. If the value is 0,
 then
  // no attributes were dropped.
  uint32 dropped_attributes_count = 2;
}
```

El objeto ScopeMetrics

El campo `scope` no se rellenará. Se rellena solo el campo de métricas que se está exportando.

```
// A collection of Metrics produced by an Scope.
message ScopeMetrics {
  // The instrumentation scope information for the metrics in this message.
```

```
  // Semantically when InstrumentationScope isn't set, it is equivalent with
  // an empty instrumentation scope name (unknown).
  opentelemetry.proto.common.v1.InstrumentationScope scope = 1;

  // A list of metrics that originate from an instrumentation library.
  repeated Metric metrics = 2;

  // This schema_url applies to all metrics in the "metrics" field.
  string schema_url = 3;
}
```

El objeto métrica

El objeto métrica contiene algunos metadatos y un campo de datos `Summary` que contiene una lista de `SummaryDataPoint`.

En el caso de los flujos métricos, los metadatos son los siguientes:

- `name` será `amazonaws.com/`*`metric_namespace`*`/`*`metric_name`*
- `description` estará en blanco
- `unit` se rellenará al mapear la unidad del dato métrico a la variante de reconocimiento de mayúsculas y minúsculas del código unificado para unidades de medida. Para obtener más información, consulte Conversiones con formato OpenTelemetry 1.0.0 y The Unified Code For Units of Measure (Código unificado para las unidades de medida).
- `type` será `SUMMARY`

```
message Metric {
  reserved 4, 6, 8;

  // name of the metric, including its DNS name prefix. It must be unique.
  string name = 1;

  // description of the metric, which can be used in documentation.
  string description = 2;

  // unit in which the metric value is reported. Follows the format
  // described by http://unitsofmeasure.org/ucum.html.
  string unit = 3;

  // Data determines the aggregation type (if any) of the metric, what is the
```

```
  // reported value type for the data points, as well as the relatationship to
  // the time interval over which they are reported.
  oneof data {
    Gauge gauge = 5;
    Sum sum = 7;
    Histogram histogram = 9;
    ExponentialHistogram exponential_histogram = 10;
    Summary summary = 11;
  }
}

message Summary {
  repeated SummaryDataPoint data_points = 1;
}
```

El objeto SummaryDataPoint

El objeto SummaryDataPoint contiene el valor de un único punto de datos de una serie temporal en una métrica DoubleSummary.

```
// SummaryDataPoint is a single data point in a timeseries that describes the
// time-varying values of a Summary metric.
message SummaryDataPoint {
  reserved 1;

  // The set of key/value pairs that uniquely identify the timeseries from
  // where this point belongs. The list may be empty (may contain 0 elements).
  // Attribute keys MUST be unique (it is not allowed to have more than one
  // attribute with the same key).
  repeated opentelemetry.proto.common.v1.KeyValue attributes = 7;

  // StartTimeUnixNano is optional but strongly encouraged, see the
  // the detailed comments above Metric.
  //
  // Value is UNIX Epoch time in nanoseconds since 00:00:00 UTC on 1 January
  // 1970.
  fixed64 start_time_unix_nano = 2;

  // TimeUnixNano is required, see the detailed comments above Metric.
  //
  // Value is UNIX Epoch time in nanoseconds since 00:00:00 UTC on 1 January
  // 1970.
  fixed64 time_unix_nano = 3;
```

```
  // count is the number of values in the population. Must be non-negative.
  fixed64 count = 4;

  // sum of the values in the population. If count is zero then this field
  // must be zero.
  //
  // Note: Sum should only be filled out when measuring non-negative discrete
  // events, and is assumed to be monotonic over the values of these events.
  // Negative events *can* be recorded, but sum should not be filled out when
  // doing so.  This is specifically to enforce compatibility w/ OpenMetrics,
  // see: https://github.com/OpenObservability/OpenMetrics/blob/main/specification/
OpenMetrics.md#summary
  double sum = 5;

  // Represents the value at a given quantile of a distribution.
  //
  // To record Min and Max values following conventions are used:
  // - The 1.0 quantile is equivalent to the maximum value observed.
  // - The 0.0 quantile is equivalent to the minimum value observed.
  //
  // See the following issue for more context:
  // https://github.com/open-telemetry/opentelemetry-proto/issues/125
  message ValueAtQuantile {
    // The quantile of a distribution. Must be in the interval
    // [0.0, 1.0].
    double quantile = 1;

    // The value at the given quantile of a distribution.
    //
    // Quantile values must NOT be negative.
    double value = 2;
  }

  // (Optional) list of values at different quantiles of the distribution calculated
  // from the current snapshot. The quantiles must be strictly increasing.
  repeated ValueAtQuantile quantile_values = 6;

  // Flags that apply to this specific data point.  See DataPointFlags
  // for the available flags and their meaning.
  uint32 flags = 8;
}
```

Para obtener más información, consulte Conversiones con formato OpenTelemetry 1.0.0.

Conversiones con formato OpenTelemetry 1.0.0

CloudWatch realiza algunas transformaciones para poner los datos de CloudWatch en formato OpenTelemetry.

Conversión de espacio de nombres, nombre de métrica y dimensiones

Estos atributos son pares clave-valor codificados en el mapeo.

- Un atributo tiene la clave `Namespace` y su valor es el espacio de nombres de la métrica
- Un atributo tiene la clave `MetricName` y su valor es el nombre de la métrica
- Un par cuenta con la clave `Dimensions` y su valor es una lista anidada de pares clave-valor. Cada par de esta lista se asigna a una dimensión métrica de CloudWatch, donde la clave del par es el nombre de la dimensión y su valor es el valor de la dimensión.

Conversión de promedio, suma, recuento de muestra, mínima y máxima

El punto de datos de resumen permite que CloudWatch exporte todas estas estadísticas mediante un punto de datos.

- `startTimeUnixNano` contiene el `startTime` de CloudWatch
- `timeUnixNano` contiene el `endTime` de CloudWatch
- `sum` contiene la estadística de la suma.
- `count` contiene la estadística del recuento de muestra.
- `quantile_values` contiene dos objetos `valueAtQuantile.value`:
  - `valueAtQuantile.quantile = 0.0` por `valueAtQuantile.value = Min value`
  - `valueAtQuantile.quantile = 0.99` por `valueAtQuantile.value = p99 value`
  - `valueAtQuantile.quantile = 0.999` por `valueAtQuantile.value = p99.9 value`
  - `valueAtQuantile.quantile = 1.0` por `valueAtQuantile.value = Max value`

Los recursos que consumen el flujo métrico pueden calcular la estadística promedio como Sum/SampleCount (Suma/recuento de muestra).

Conversión de unidades

Las unidades de CloudWatch se mapean a la variante de reconocimiento de mayúsculas y minúsculas del código unificado para unidades de medida, como se muestra en la siguiente tabla. Para obtener más información, consulte The Unified Code For Units of Measure (Código unificado para las unidades de medida).

CloudWatch	OpenTelemetry
Segundo	s
Segundo o segundos	s
Microsegundos	EE. UU.
Milisegundos	ms
Bytes	B
Kilobytes	KB
Megabytes	MB
Gigabytes	GB
Terabytes	TB
Bits	bit
Kilobits	Kbit
Megabits	Mbit
Gigabits	Gbit
Terabits	Tbit
Porcentaje	%
Recuento	{Count}
Ninguna	1

Las unidades que se combinan con una barra diagonal se asignan mediante la conversión OpenTelemetry de ambas unidades. Por ejemplo, Bytes/segundo se asigna a B/s.

Descubra cómo se analizan los mensajes de OpenTelemetry 1.0.0

En esta sección se proporciona información que lo ayudará a comenzar a analizar OpenTelemetry 1.0.0.

En primer lugar, debe obtener enlaces específicos de idioma, que le permiten analizar mensajes de OpenTelemetry 1.0.0 en el idioma de su preferencia.

Para obtener enlaces específicos de idioma

- Los pasos dependen del idioma que prefiera.
  - Para utilizar Java, agregue la siguiente dependencia de Maven a su proyecto de Java: OpenTelemetry Java >> 0.14.1.
  - Para utilizar cualquier otro idioma, siga estos pasos:

    a. Asegúrese de que su idioma es compatible; puede verificarlo en la lista en Generating Your Classes (Generación de clases).

    b. Instale el compilador Protobuf con los pasos que se indican en Download Protocol Buffers (Descargar búferes de protocolo).

    c. Descargue las definiciones de OpenTelemetry 0.7.0 ProtoBuf en Versión de lanzamiento 1.0.0.

    d. Confirme que se encuentra en la carpeta raíz de las definiciones OpenTelemetry 0.7.0 ProtoBuf descargadas. Después, cree una carpeta de `src` y, a continuación, ejecute el comando para generar enlaces específicos de idioma. Para obtener más información, consulte Generating Your Classes (Generación de clases).

       A continuación se muestra un ejemplo de cómo se generan los enlaces de Javascript.

```
protoc --proto_path=./ --js_out=import_style=commonjs,binary:src \
opentelemetry/proto/common/v1/common.proto \
opentelemetry/proto/resource/v1/resource.proto \
opentelemetry/proto/metrics/v1/metrics.proto \
opentelemetry/proto/collector/metrics/v1/metrics_service.proto
```

En la siguiente sección se incluyen ejemplos de uso de enlaces específicos de idioma que se pueden crear mediante las instrucciones anteriores.

Java

```
package com.example;

import io.opentelemetry.proto.collector.metrics.v1.ExportMetricsServiceRequest;

import java.io.IOException;
import java.io.InputStream;
import java.util.ArrayList;
import java.util.List;

public class MyOpenTelemetryParser {

    public List<ExportMetricsServiceRequest> parse(InputStream inputStream) throws
 IOException {
        List<ExportMetricsServiceRequest> result = new ArrayList<>();

        ExportMetricsServiceRequest request;
        /* A Kinesis record can contain multiple `ExportMetricsServiceRequest`
           records, each of them starting with a header with an
           UnsignedVarInt32 indicating the record length in bytes:
            ------ -------------------------- ------ ----------------------
           |UINT32|ExportMetricsServiceRequest|UINT32|ExportMetricsService...
            ------ -------------------------- ------ ----------------------
         */
        while ((request =
 ExportMetricsServiceRequest.parseDelimitedFrom(inputStream)) != null) {
            // Do whatever we want with the parsed message
            result.add(request);
        }

        return result;
    }
}
```

Javascript

En este ejemplo se asume que la carpeta raíz con los enlaces generados es `./`

El argumento de datos de la función `parseRecord` puede tratarse de uno de los siguientes tipos:

- `Uint8Array` es opcional.
- `Buffer` óptimo debajo del nodo
- `Array.number` enteros de 8 bits

```
const pb = require('google-protobuf')
const pbMetrics =
    require('./opentelemetry/proto/collector/metrics/v1/metrics_service_pb')

function parseRecord(data) {
    const result = []

    // Loop until we've read all the data from the buffer
    while (data.length) {
        /* A Kinesis record can contain multiple `ExportMetricsServiceRequest`
           records, each of them starting with a header with an
           UnsignedVarInt32 indicating the record length in bytes:
            ------ -------------------------- ------ ----------------------
           |UINT32|ExportMetricsServiceRequest|UINT32|ExportMetricsService...
            ------ -------------------------- ------ ----------------------
         */
        const reader = new pb.BinaryReader(data)
        const messageLength = reader.decoder_.readUnsignedVarint32()
        const messageFrom = reader.decoder_.cursor_
        const messageTo = messageFrom + messageLength

        // Extract the current `ExportMetricsServiceRequest` message to parse
        const message = data.subarray(messageFrom, messageTo)

        // Parse the current message using the ProtoBuf library
        const parsed =
            pbMetrics.ExportMetricsServiceRequest.deserializeBinary(message)

        // Do whatever we want with the parsed message
        result.push(parsed.toObject())

        // Shrink the remaining buffer, removing the already parsed data
        data = data.subarray(messageTo)
    }

    return result
}
```

Python

Debe leer los delimitadores `var-int` usted mismo o utilizar los métodos internos `_VarintBytes(size)` y `_DecodeVarint32(buffer, position)`. Estos devuelven la posición en el búfer justo después de los bytes. El lado de lectura construye un nuevo búfer que se limita a leer sólo los bytes del mensaje.

```
size = my_metric.ByteSize()
f.write(_VarintBytes(size))
f.write(my_metric.SerializeToString())
msg_len, new_pos = _DecodeVarint32(buf, 0)
msg_buf = buf[new_pos:new_pos+msg_len]
request = metrics_service_pb.ExportMetricsServiceRequest()
request.ParseFromString(msg_buf)
```

Go

Utilice `Buffer.DecodeMessage()`.

C#

Utilice `CodedInputStream`. Esta clase puede leer mensajes delimitados por tamaño.

C++

Las funciones descritas en `google/protobuf/util/delimited_message_util.h` pueden leer mensajes delimitados por tamaño.

Otros idiomas

Para ver otros idiomas, consulte Download Protocol Buffers (Descargar búferes de protocolo).

Al implementar el analizador, tenga en cuenta que un registro Kinesis puede contener varios mensajes de búferes de protocolo `ExportMetricsServiceRequest`, cada uno de ellos con un encabezado con un `UnsignedVarInt32` que indica la longitud de registro en bytes.

Formato OpenTelemetry 0.7.0

OpenTelemetry es una colección de herramientas, API y SDK. Puede utilizarlo para instrumentar, generar, recopilar y exportar datos telemétricos (métricas, registros y seguimientos) para analizarlos. OpenTelemetry es parte de Cloud Native Computing Foundation. Para obtener más información, consulte OpenElemetry.

Para obtener información acerca de la especificación completa de OpenTelemetry 0.7.0, consulte v0.7.0 release (Versión v0.7.0).

Un registro de Kinesis puede contener una o más estructuras de datos OpenTelemetry `ExportMetricsServiceRequest`. Cada estructura de datos comienza con una cabecera con un `UnsignedVarInt32` que indica la longitud de registro en bytes. Cada `ExportMetricsServiceRequest` puede contener datos de varias métricas a la vez.

A continuación, se muestra una cadena de representación del mensaje de la estructura de datos OpenTelemetry `ExportMetricsServiceRequest`. OpenTelemetry serializa el protocolo binario Google Protocol Buffers y este no es legible para el ser humano.

```
resource_metrics {
  resource {
    attributes {
      key: "cloud.provider"
      value {
        string_value: "aws"
      }
    }
    attributes {
      key: "cloud.account.id"
      value {
        string_value: "2345678901"
      }
    }
    attributes {
      key: "cloud.region"
      value {
        string_value: "us-east-1"
      }
    }
    attributes {
      key: "aws.exporter.arn"
      value {
        string_value: "arn:aws:cloudwatch:us-east-1:123456789012:metric-stream/
MyMetricStream"
      }
    }
  }
  instrumentation_library_metrics {
    metrics {
      name: "amazonaws.com/AWS/DynamoDB/ConsumedReadCapacityUnits"
```

```
      unit: "1"
      double_summary {
        data_points {
          labels {
            key: "Namespace"
            value: "AWS/DynamoDB"
          }
          labels {
            key: "MetricName"
            value: "ConsumedReadCapacityUnits"
          }
          labels {
            key: "TableName"
            value: "MyTable"
          }
          start_time_unix_nano: 1604948400000000000
          time_unix_nano: 1604948460000000000
          count: 1
          sum: 1.0
          quantile_values {
            quantile: 0.0
            value: 1.0
          }
          quantile_values {
            quantile: 0.95
            value: 1.0
          }
          quantile_values {
            quantile: 0.99
            value: 1.0
          }
          quantile_values {
            quantile: 1.0
            value: 1.0
          }
        }
        data_points {
          labels {
            key: "Namespace"
            value: "AWS/DynamoDB"
          }
          labels {
            key: "MetricName"
            value: "ConsumedReadCapacityUnits"
```

```
          }
          labels {
            key: "TableName"
            value: "MyTable"
          }
          start_time_unix_nano: 1604948460000000000
          time_unix_nano: 1604948520000000000
          count: 2
          sum: 5.0
          quantile_values {
            quantile: 0.0
            value: 2.0
          }
          quantile_values {
            quantile: 1.0
            value: 3.0
          }
        }
      }
    }
  }
}
```

Objeto de nivel superior para serializar datos de métrica de OpenTelemetry

`ExportMetricsServiceRequest` es el envoltorio de nivel superior para serializar una carga del exportador de OpenTelemetry. Contiene uno o más `ResourceMetrics`.

```
message ExportMetricsServiceRequest {
  // An array of ResourceMetrics.
  // For data coming from a single resource this array will typically contain one
  // element. Intermediary nodes (such as OpenTelemetry Collector) that receive
  // data from multiple origins typically batch the data before forwarding further and
  // in that case this array will contain multiple elements.
  repeated opentelemetry.proto.metrics.v1.ResourceMetrics resource_metrics = 1;
}
```

`ResourceMetrics` es el objeto de nivel superior para representar objetos MetricData.

```
// A collection of InstrumentationLibraryMetrics from a Resource.
message ResourceMetrics {
  // The resource for the metrics in this message.
  // If this field is not set then no resource info is known.
```

```
  opentelemetry.proto.resource.v1.Resource resource = 1;

  // A list of metrics that originate from a resource.
  repeated InstrumentationLibraryMetrics instrumentation_library_metrics = 2;
}
```

El objeto recurso

Un objeto `Resource` es un objeto de par de valores que contiene información sobre el recurso que generó las métricas. Para las métricas que AWS crea, la estructura de datos contiene el Nombre de recurso de Amazon (ARN) del recurso relacionado con la métrica, como una instancia EC2 o un bucket de S3.

El objeto `Resource` contiene un atributo llamado `attributes`, que almacena una lista de pares clave-valor.

- `cloud.account.id` contiene el ID de la cuenta
- `cloud.region` contiene la Región
- `aws.exporter.arn` contiene el ARN del flujo métrico
- `cloud.provider` es siempre `aws`.

```
// Resource information.
message Resource {
  // Set of labels that describe the resource.
  repeated opentelemetry.proto.common.v1.KeyValue attributes = 1;

  // dropped_attributes_count is the number of dropped attributes. If the value is 0,
  // no attributes were dropped.
  uint32 dropped_attributes_count = 2;
}
```

El objeto InstrumentationLibraryMetrics

El campo instrumentation_library no se rellenará. Se rellenará sólo el campo de métricas que se está exportando.

```
// A collection of Metrics produced by an InstrumentationLibrary.
message InstrumentationLibraryMetrics {
  // The instrumentation library information for the metrics in this message.
```

```
  // If this field is not set then no library info is known.
  opentelemetry.proto.common.v1.InstrumentationLibrary instrumentation_library = 1;
  // A list of metrics that originate from an instrumentation library.
  repeated Metric metrics = 2;
}
```

El objeto métrica

El objeto métrica contiene un campo de datos `DoubleSummary` que contiene una lista de `DoubleSummaryDataPoint`.

```
message Metric {
  // name of the metric, including its DNS name prefix. It must be unique.
  string name = 1;

  // description of the metric, which can be used in documentation.
  string description = 2;

  // unit in which the metric value is reported. Follows the format
  // described by http://unitsofmeasure.org/ucum.html.
  string unit = 3;

  oneof data {
    IntGauge int_gauge = 4;
    DoubleGauge double_gauge = 5;
    IntSum int_sum = 6;
    DoubleSum double_sum = 7;
    IntHistogram int_histogram = 8;
    DoubleHistogram double_histogram = 9;
    DoubleSummary double_summary = 11;
  }
}

message DoubleSummary {
  repeated DoubleSummaryDataPoint data_points = 1;
}
```

El objeto MetricDescriptor

El objeto MetricDescriptor contiene metadatos. Para obtener más información, consulte metrics.proto en GitHub.

Para flujos métricos, MetricDescriptor cuenta con el siguiente contenido:

- `name` será `amazonaws.com/`*`metric_namespace`*`/`*`metric_name`*
- `description` estará en blanco.
- `unit` se rellenará al mapear la unidad del dato métrico a la variante de reconocimiento de mayúsculas y minúsculas del código unificado para unidades de medida. Para obtener más información, consulte Conversiones con formato OpenTelemetry 0.7.0 y The Unified Code For Units of Measure (Código unificado para las unidades de medida).
- `type` será `SUMMARY`.

El objeto DoubleSummaryDataPoint

El objeto DoubleSummaryDataPoint contiene el valor de un único punto de datos de una serie temporal en una métrica DoubleSummary.

```
// DoubleSummaryDataPoint is a single data point in a timeseries that describes the
// time-varying values of a Summary metric.
message DoubleSummaryDataPoint {
  // The set of labels that uniquely identify this timeseries.
  repeated opentelemetry.proto.common.v1.StringKeyValue labels = 1;

  // start_time_unix_nano is the last time when the aggregation value was reset
  // to "zero". For some metric types this is ignored, see data types for more
  // details.
  //
  // The aggregation value is over the time interval (start_time_unix_nano,
  // time_unix_nano].
  //
  // Value is UNIX Epoch time in nanoseconds since 00:00:00 UTC on 1 January
  // 1970.
  //
  // Value of 0 indicates that the timestamp is unspecified. In that case the
  // timestamp may be decided by the backend.
  fixed64 start_time_unix_nano = 2;

  // time_unix_nano is the moment when this aggregation value was reported.
  //
  // Value is UNIX Epoch time in nanoseconds since 00:00:00 UTC on 1 January
  // 1970.
  fixed64 time_unix_nano = 3;

  // count is the number of values in the population. Must be non-negative.
  fixed64 count = 4;
```

```
  // sum of the values in the population. If count is zero then this field
  // must be zero.
  double sum = 5;

  // Represents the value at a given quantile of a distribution.
  //
  // To record Min and Max values following conventions are used:
  // - The 1.0 quantile is equivalent to the maximum value observed.
  // - The 0.0 quantile is equivalent to the minimum value observed.
  message ValueAtQuantile {
    // The quantile of a distribution. Must be in the interval
    // [0.0, 1.0].
    double quantile = 1;

    // The value at the given quantile of a distribution.
    double value = 2;
  }

  // (Optional) list of values at different quantiles of the distribution calculated
  // from the current snapshot. The quantiles must be strictly increasing.
  repeated ValueAtQuantile quantile_values = 6;
}
```

Para obtener más información, consulte Conversiones con formato OpenTelemetry 0.7.0.

Conversiones con formato OpenTelemetry 0.7.0

CloudWatch realiza algunas transformaciones para poner los datos de CloudWatch en formato OpenTelemetry.

Conversión de espacio de nombres, nombre de métrica y dimensiones

Estos atributos son pares clave-valor codificados en el mapeo.

- Un par contiene el espacio de nombres de la métrica
- Un par contiene el nombre de la métrica
- Para cada dimensión, CloudWatch almacena el siguiente par: `metricDatum.Dimensions[i].Name, metricDatum.Dimensions[i].Value`

Conversión de promedio, suma, recuento de muestra, mínima y máxima

El punto de datos de resumen permite que CloudWatch exporte todas estas estadísticas mediante un punto de datos.

- `startTimeUnixNano` contiene el `startTime` de CloudWatch
- `timeUnixNano` contiene el `endTime` de CloudWatch
- `sum` contiene la estadística de la suma.
- `count` contiene la estadística del recuento de muestra.
- `quantile_values` contiene dos objetos `valueAtQuantile.value`:
  - `valueAtQuantile.quantile = 0.0` por `valueAtQuantile.value = Min value`
  - `valueAtQuantile.quantile = 0.99` por `valueAtQuantile.value = p99 value`
  - `valueAtQuantile.quantile = 0.999` por `valueAtQuantile.value = p99.9 value`
  - `valueAtQuantile.quantile = 1.0` por `valueAtQuantile.value = Max value`

Los recursos que consumen el flujo métrico pueden calcular la estadística promedio como Sum/SampleCount (Suma/recuento de muestra).

Conversión de unidades

Las unidades de CloudWatch se mapean a la variante de reconocimiento de mayúsculas y minúsculas del código unificado para unidades de medida, como se muestra en la siguiente tabla. Para obtener más información, consulte The Unified Code For Units of Measure (Código unificado para las unidades de medida).

CloudWatch	OpenTelemetry
Segundo	s
Segundo o segundos	s
Microsegundo	EE. UU.
Milisegundos	ms
Bytes	B
Kilobytes	KB

CloudWatch	OpenTelemetry
Megabytes	MB
Gigabytes	GB
Terabytes	TB
Bits	bit
Kilobits	Kbit
Megabits	Mbit
Gigabits	Gbit
Terabits	Tbit
Porcentaje	%
Recuento	{Count}
Ninguna	1

Las unidades que se combinan con una barra diagonal se asignan mediante la conversión OpenTelemetry de ambas unidades. Por ejemplo, Bytes/segundo se asigna a B/s.

Descubra cómo se analizan los mensajes de OpenTelemetry 0.7.0

En esta sección se proporciona información que le ayudará a comenzar a analizar OpenTelemetry 0.7.0.

En primer lugar, debe obtener enlaces específicos de idioma, que le permiten analizar mensajes OpenTelemetry 0.7.0 en el idioma de su preferencia.

Para obtener enlaces específicos de idioma

- Los pasos dependen del idioma que prefiera.
  - Para utilizar Java, agregue la siguiente dependencia de Maven a su proyecto de Java: OpenTelemetry Java >> 0.14.1.

- Para utilizar cualquier otro idioma, siga estos pasos:

  a. Asegúrese de que su idioma es compatible; puede verificarlo en la lista en Generating Your Classes (Generación de clases).

  b. Instale el compilador Protobuf con los pasos que se indican en Download Protocol Buffers (Descargar búferes de protocolo).

  c. Descargue las definiciones de OpenTelemetry 0.7.0 ProtoBuf en v0.7.0 release (Versión v0.7.0).

  d. Confirme que se encuentra en la carpeta raíz de las definiciones OpenTelemetry 0.7.0 ProtoBuf descargadas. Después, cree una carpeta de `src` y, a continuación, ejecute el comando para generar enlaces específicos de idioma. Para obtener más información, consulte Generating Your Classes (Generación de clases).

     A continuación se muestra un ejemplo de cómo se generan los enlaces de Javascript.

     ```
     protoc --proto_path=./ --js_out=import_style=commonjs,binary:src \
     opentelemetry/proto/common/v1/common.proto \
     opentelemetry/proto/resource/v1/resource.proto \
     opentelemetry/proto/metrics/v1/metrics.proto \
     opentelemetry/proto/collector/metrics/v1/metrics_service.proto
     ```

En la siguiente sección se incluyen ejemplos de uso de enlaces específicos de idioma que se pueden crear mediante las instrucciones anteriores.

Java

```
package com.example;

import io.opentelemetry.proto.collector.metrics.v1.ExportMetricsServiceRequest;

import java.io.IOException;
import java.io.InputStream;
import java.util.ArrayList;
import java.util.List;

public class MyOpenTelemetryParser {

    public List<ExportMetricsServiceRequest> parse(InputStream inputStream) throws
 IOException {
```

```
        List<ExportMetricsServiceRequest> result = new ArrayList<>();

        ExportMetricsServiceRequest request;
        /* A Kinesis record can contain multiple `ExportMetricsServiceRequest`
           records, each of them starting with a header with an
           UnsignedVarInt32 indicating the record length in bytes:
            ------ -------------------------- ------ ----------------------
           |UINT32|ExportMetricsServiceRequest|UINT32|ExportMetricsService...
            ------ -------------------------- ------ ----------------------
         */
        while ((request =
 ExportMetricsServiceRequest.parseDelimitedFrom(inputStream)) != null) {
            // Do whatever we want with the parsed message
            result.add(request);
        }

        return result;
    }
}
```

Javascript

En este ejemplo se asume que la carpeta raíz con los enlaces generados es `./`

El argumento de datos de la función `parseRecord` puede tratarse de uno de los siguientes tipos:

- `Uint8Array` es opcional.
- `Buffer` óptimo debajo del nodo
- `Array.`*`number`* enteros de 8 bits

```
const pb = require('google-protobuf')
const pbMetrics =
    require('./opentelemetry/proto/collector/metrics/v1/metrics_service_pb')

function parseRecord(data) {
    const result = []

    // Loop until we've read all the data from the buffer
    while (data.length) {
        /* A Kinesis record can contain multiple `ExportMetricsServiceRequest`
           records, each of them starting with a header with an
           UnsignedVarInt32 indicating the record length in bytes:
```

```
            ------ -------------------------- ------ ----------------------
           |UINT32|ExportMetricsServiceRequest|UINT32|ExportMetricsService...
            ------ -------------------------- ------ ----------------------
         */
        const reader = new pb.BinaryReader(data)
        const messageLength = reader.decoder_.readUnsignedVarint32()
        const messageFrom = reader.decoder_.cursor_
        const messageTo = messageFrom + messageLength

        // Extract the current `ExportMetricsServiceRequest` message to parse
        const message = data.subarray(messageFrom, messageTo)

        // Parse the current message using the ProtoBuf library
        const parsed =
            pbMetrics.ExportMetricsServiceRequest.deserializeBinary(message)

        // Do whatever we want with the parsed message
        result.push(parsed.toObject())

        // Shrink the remaining buffer, removing the already parsed data
        data = data.subarray(messageTo)
    }

    return result
}
```

Python

Debe leer los delimitadores `var-int` usted mismo o utilizar los métodos internos `_VarintBytes(size)` y `_DecodeVarint32(buffer, position)`. Estos devuelven la posición en el búfer justo después de los bytes. El lado de lectura construye un nuevo búfer que se limita a leer sólo los bytes del mensaje.

```
size = my_metric.ByteSize()
f.write(_VarintBytes(size))
f.write(my_metric.SerializeToString())
msg_len, new_pos = _DecodeVarint32(buf, 0)
msg_buf = buf[new_pos:new_pos+msg_len]
request = metrics_service_pb.ExportMetricsServiceRequest()
request.ParseFromString(msg_buf)
```

Go

Utilice `Buffer.DecodeMessage()`.

C#

Utilice `CodedInputStream`. Esta clase puede leer mensajes delimitados por tamaño.

C++

Las funciones descritas en `google/protobuf/util/delimited_message_util.h` pueden leer mensajes delimitados por tamaño.

Otros idiomas

Para ver otros idiomas, consulte Download Protocol Buffers (Descargar búferes de protocolo).

Al implementar el analizador, tenga en cuenta que un registro Kinesis puede contener varios mensajes de búferes de protocolo `ExportMetricsServiceRequest`, cada uno de ellos con un encabezado con un `UnsignedVarInt32` que indica la longitud de registro en bytes.

Solución de problemas

Si no ve datos de métricas en el destino final, verifique lo siguiente:

- Verifique que la secuencia métrica esté en el estado de ejecución. Para obtener pasos acerca de cómo utilizar la consola de CloudWatch para hacerlo, consulte Operación y mantenimiento del flujo métrico.
- Las métricas publicadas hace más de dos días no se transmiten. Para determinar si se transmitirá una métrica concreta, grafique la métrica en la consola de CloudWatch y compruebe qué antigüedad tiene el último punto de datos visible. Si han pasado más de dos días, los flujos métricos no lo recogerán.
- Verifique las métricas que el flujo métrico emite. En la consola de CloudWatch, bajo el título Metrics (Métricas), consulte el espacio de nombres AWS/CloudWatch/MetricStreams para las métricas MetricUpdate, TotalMetricUpdate y PublishErrorRate.
- Si la métrica PublishErrorRate es alta, confirme que existe el destino que utiliza el flujo de entrega de Firehose y que el rol de IAM especificado en la configuración del flujo métrico otorga los permisos principales del servicio `CloudWatch` para escribir en él. Para obtener más información, consulte Confianza entre CloudWatch y Firehose.
- Verifique que el flujo de entrega de Firehose tenga permiso para escribir en el destino final.
- En la consola de Firehose, vea el flujo de entrega de Firehose que se utiliza para el flujo métrico y verifique la pestaña Monitoreo para ver si el flujo de entrega de Firehose está recibiendo datos.

- Confirme que ha configurado el flujo de entrega de Firehose con los detalles correctos.
- Verifique los registros o métricas disponibles para el destino final en el que el flujo de entrega de Firehose escribe.
- Para obtener información más detallada, habilite el registro de errores de Registros de CloudWatch en el flujo de entrega de Firehose. Para obtener más información, consulte Supervisión de Amazon Data Firehose mediante Registros de CloudWatch.

Ver métricas disponibles

Las métricas se agrupan en primer lugar por el espacio de nombres y, a continuación, por las diversas combinaciones de dimensiones dentro de cada espacio de nombres. Por ejemplo, puede ver todas las métricas de EC2, las métricas de EC2 agrupadas por instancia o las métricas de EC2 agrupadas por grupo de Auto Scaling.

Solo los servicios de AWS que está utilizando envían métricas a Amazon CloudWatch.

Para obtener una lista de los servicios de AWS que envían métricas a CloudWatch, consulte Servicios de AWS que publican métricas de CloudWatch. En esta página, también pude ver las métricas y dimensiones publicadas por cada uno de estos servicios.

> **Note**
>
> Las métricas que no han tenido nuevos puntos de datos en las últimas dos semanas no aparecen en la consola. Tampoco aparecen al escribir su nombre de métrica o nombres de dimensión en el cuadro de búsqueda de la pestaña Todas las métricas de la consola y no se devuelven en los resultados de un comando list-metrics. La mejor manera de recuperar estas métricas es con los comandos get-metric-data o get-metric-statistics de la AWS CLI.
> Si la métrica antigua que desea consultar tiene una métrica actual con dimensiones similares, puede ver esa métrica actual similar y, a continuación, elegir la pestaña Origen y cambiar el nombre de la métrica y los campos de dimensión a los que desee, así como cambiar el intervalo de tiempo a una hora a la que se estaba notificando la métrica.

Los siguientes pasos le ayudan a navegar por los espacios de nombres de métricas para buscar y ver métricas. También puede buscar métricas mediante términos de búsqueda dirigidos. Para obtener más información, consulte Buscar métricas disponibles.

Si está explorando una cuenta configurada como cuenta de supervisión para la observabilidad entre cuentas de CloudWatch, puede ver las métricas de las cuentas de origen vinculadas a esta cuenta de supervisión. Cuando se muestran las métricas de las cuentas de origen, también se muestra el ID o la etiqueta de la cuenta de la que provengan. Para obtener más información, consulte Observabilidad entre cuentas de CloudWatch.

Para ver las métricas disponibles por espacio de nombres y dimensión mediante la consola

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, seleccione Métricas y, a continuación, Todas las métricas.
3. Seleccione un espacio de nombres de métrica (por ejemplo, EC2 o Lambda).
4. Seleccione una dimensión de métrica (por ejemplo, Per-Instance Metrics [Métricas por instancia] o By Function Name [Por nombre de función]).
5. La pestaña Browse (Explorar) muestra todas las métricas para dicha dimensión en el espacio de nombres. Junto al nombre de cada métrica hay un botón de información que puede elegir para ver una ventana emergente con la definición de la métrica.

   Si se trata de una cuenta de supervisión para la observabilidad entre cuentas de CloudWatch, también verá las métricas de las cuentas de origen vinculadas a esta cuenta de supervisión. Las columnas Account label (Etiqueta de cuenta) y Account id (ID de cuenta) de la tabla muestran de qué cuenta proviene cada métrica.

   Puede hacer lo siguiente:

   a. Para ordenar la tabla, utilice el encabezado de columna.
   b. Para representar gráficamente una métrica, active la casilla de verificación situada junto a ella. Para seleccionar todas las métricas, seleccione la casilla de verificación en la fila de encabezado de la tabla.
   c. Para filtrar por cuenta, seleccione la etiqueta de la cuenta o el id. de la cuenta y luego elija Add to search (Agregar a la búsqueda).
   d. Para filtrar por recurso, seleccione el ID de recurso y, a continuación, elija Add to search (Añadir a la búsqueda).
   e. Para filtrar por métrica, elija el nombre de la métrica y, a continuación, seleccione Add to search (Añadir a búsqueda).
6. (Opcional) Para agregar el gráfico a un panel de CloudWatch, elija Actions (Acciones) y después Add to dashboard (Añadir al panel).

Consulta de las métricas disponibles según el espacio de nombres, dimensión o métrica de cuenta mediante AWS CLI

Utilice el comando list-metrics para mostrar un listado de métricas de CloudWatch. Para obtener una lista de los espacios de nombres, métricas y dimensiones de todos los servicios que publican métricas, consulte Servicios de AWS que publican métricas de CloudWatch.

El siguiente comando de ejemplo muestra todas las métricas de Amazon EC2.

```
aws cloudwatch list-metrics --namespace AWS/EC2
```

A continuación, se muestra un ejemplo del resultado.

```
{
  "Metrics" : [
    ...
    {
        "Namespace": "AWS/EC2",
        "Dimensions": [
            {
                "Name": "InstanceId",
                "Value": "i-1234567890abcdef0"
            }
        ],
        "MetricName": "NetworkOut"
    },
    {
        "Namespace": "AWS/EC2",
        "Dimensions": [
            {
                "Name": "InstanceId",
                "Value": "i-1234567890abcdef0"
            }
        ],
        "MetricName": "CPUUtilization"
    },
    {
        "Namespace": "AWS/EC2",
        "Dimensions": [
            {
                "Name": "InstanceId",
                "Value": "i-1234567890abcdef0"
            }
```

```
        ],
        "MetricName": "NetworkIn"
    },
    ...
  ]
}
```

Para mostrar un listado de todas las métricas disponibles para un recurso determinado

En el siguiente ejemplo, se especifica el espacio de nombres de `AWS/EC2` y la dimensión `InstanceId` para ver los resultados únicamente de la instancia especificada.

```
aws cloudwatch list-metrics --namespace AWS/EC2 --dimensions
 Name=InstanceId,Value=i-1234567890abcdef0
```

Para mostrar un listado de métricas para todos los recursos

En el siguiente ejemplo, se especifica el espacio de nombres de `AWS/EC2` y un nombre de métrica para ver los resultados únicamente de la métrica especificada.

```
aws cloudwatch list-metrics --namespace AWS/EC2 --metric-name CPUUtilization
```

Recuperación de las métricas de las cuentas de origen vinculadas en la observabilidad entre cuentas de CloudWatch

En el siguiente ejemplo se ejecuta una cuenta de supervisión para recuperar las métricas tanto de la cuenta de supervisión como de todas las cuentas de origen vinculadas. Si no agrega `--include-linked-accounts`, el comando solo devolverá las métricas de la cuenta de supervisión.

```
aws cloudwatch list-metrics --include-linked-accounts
```

Recuperación de las métricas de una cuenta de origen vinculada en la observabilidad entre cuentas de CloudWatch

En el siguiente ejemplo se ejecuta una cuenta de supervisión para recuperar las métricas de la cuenta de origen con el ID 111122223333.

```
aws cloudwatch list-metrics --include-linked-accounts --owning-account "111122223333"
```

Buscar métricas disponibles

Puede buscar en todas las métricas en su cuenta utilizando términos de búsqueda dirigidos. Se devuelven métricas que tienen resultados coincidentes dentro de su espacio de nombres, nombre de métrica o dimensiones.

Si se trata de una cuenta de supervisión para la observabilidad entre cuentas de CloudWatch, también buscará las métricas de las cuentas de origen vinculadas a esta cuenta de supervisión.

> **Note**
>
> Las métricas que no han tenido nuevos puntos de datos en las últimas dos semanas no aparecen en la consola. Tampoco aparecen al escribir su nombre de métrica o nombres de dimensión en el cuadro de búsqueda de la pestaña Todas las métricas de la consola y no se devuelven en los resultados de un comando list-metrics. La mejor manera de recuperar estas métricas es con los comandos get-metric-data o get-metric-statistics de la AWS CLI.

Para buscar métricas disponibles en CloudWatch

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, seleccione Métricas.
3. En el campo de búsqueda de la pestaña All metrics (Todas las métricas), escriba un término de búsqueda, como un nombre de métrica, espacio de nombres, nombre o valor de dimensión o nombre de recurso. Se mostrarán todos los espacios de nombres con métricas con este término de búsqueda.

   Por ejemplo, si busca **`volume`**, esto muestra los espacios de nombres que contienen métricas con este término en su nombre.

   Para obtener más información acerca de la búsqueda, consulte Usar expresiones de búsqueda en gráficos.
4. Para representar gráficamente todos los resultados de búsqueda, elija Graph search (Diagramar búsqueda)

   o

   Seleccione un espacio de nombres para ver las métricas de ese espacio de nombres. A continuación puede hacer lo siguiente:

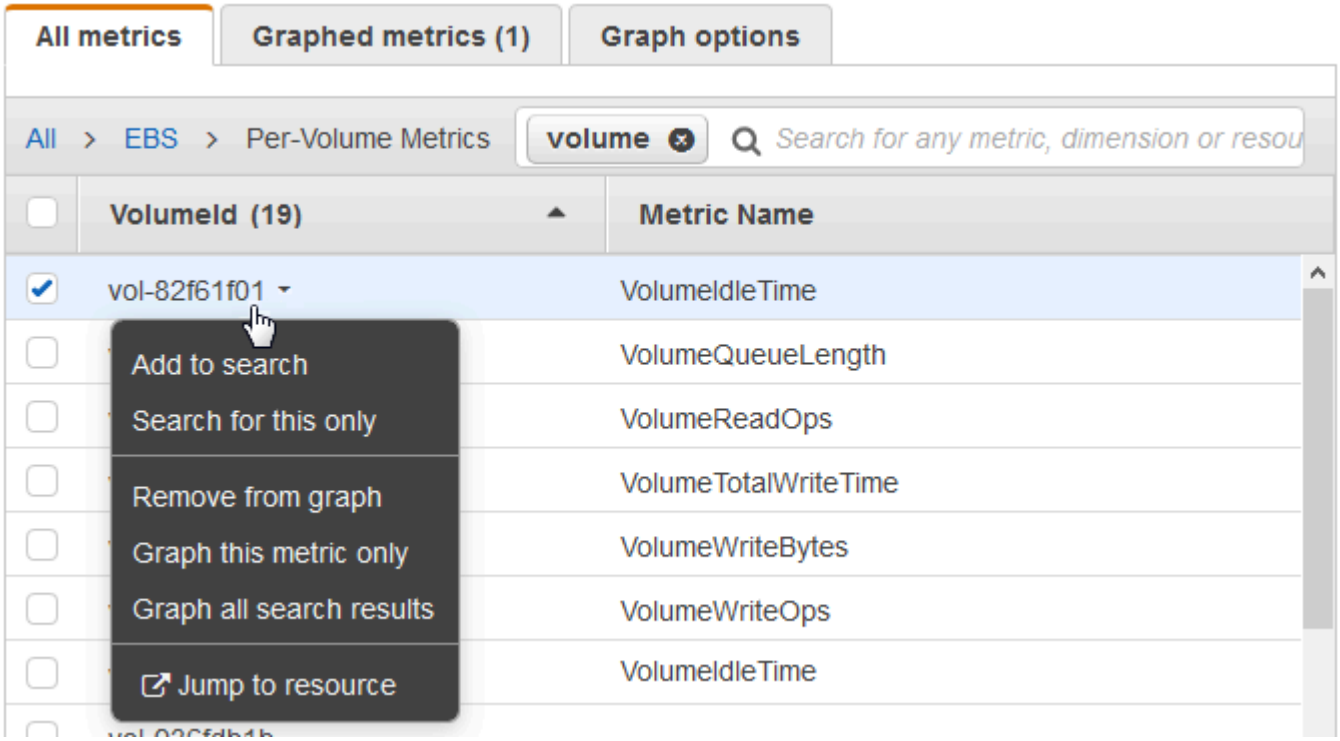
a. Para representar gráficamente una o varias métricas, seleccione la casilla de verificación junto a cada métrica. Para seleccionar todas las métricas, seleccione la casilla de verificación en la fila de encabezado de la tabla.

b. Para acotar la búsqueda, coloque el cursor sobre un nombre de métrica y elija Add to search (Añadir a búsqueda) o Search for this only (Buscar solo esto).

c. Para ver uno de los recursos en su consola, elija el ID de recurso y, a continuación, seleccione Jump to resource (Ir a recurso).

d. Para ver la ayuda de una métrica, elija el nombre de la métrica y, a continuación, seleccione What is this?.

La métricas seleccionadas aparecen en el gráfico.



5. (Opcional) Seleccione uno de los botones de la barra de búsqueda para editar esa parte del término de búsqueda.

Representación gráfica de las métricas

Utilice la consola de CloudWatch para representar gráficamente los datos de métricas generados por otros servicios de AWS. De esta forma, podrá ver más fácilmente la actividad de las métricas en sus

servicios. En los siguientes procedimientos se describe cómo se crean los gráficos de métricas en CloudWatch.

Contenido

Representar gráficamente una métrica

Puede seleccionar métricas y crear gráficos de los datos de las métricas mediante la consola de CloudWatch.

CloudWatch es compatible con las siguientes estadísticas sobre métricas: `Average`, `Minimum`, `Maximum`, `Sum` y `SampleCount`. Para obtener más información, consulte Statistics.

Puede ver los datos con diferentes niveles de detalle. Por ejemplo, puede elegir una vista de un minuto, que puede ser útil a la hora de solucionar problemas. O bien puede elegir una vista menos detallada de una hora. Esto puede ser útil cuando desee ver un intervalo de tiempo mayor (por ejemplo, tres días) para poder identificar las tendencias a lo largo del tiempo. Para obtener más información, consulte Periodos.

Si usa una cuenta configurada como cuenta de supervisión para la observabilidad entre cuentas de CloudWatch, puede ver en un gráfico las métricas de las cuentas de origen vinculadas a esta cuenta de supervisión. Para obtener más información, consulte Observabilidad entre cuentas de CloudWatch.

Creación de un gráfico

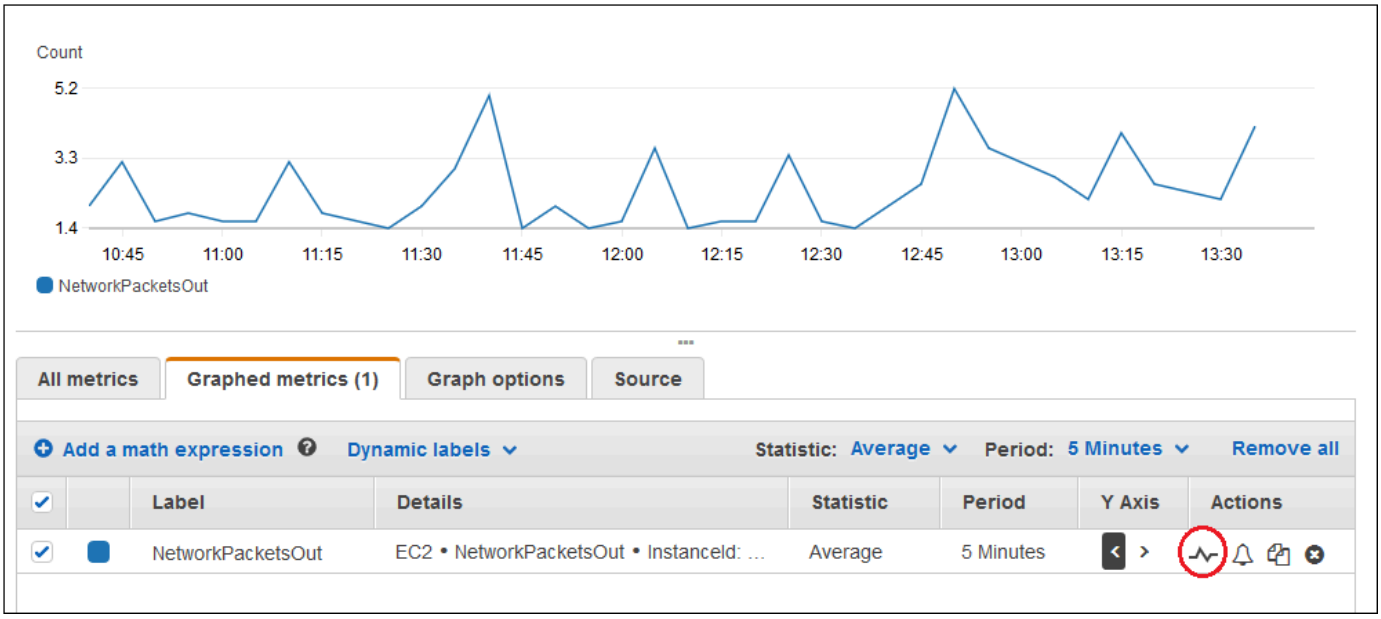
Para representar gráficamente una métrica

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.

2. En el panel de navegación, seleccione Métricas y, a continuación, Todas las métricas.
3. En la pestaña Explorar, introduzca un término de búsqueda en el campo de búsqueda, como un nombre de métrica, un ID de cuenta o un nombre de recurso.

   Por ejemplo, si busca la métrica `CPUUtilization`, verá los espacios de nombres y dimensiones con esta métrica.
4. Seleccione uno de los resultados de la búsqueda para ver las métricas.
5. Para representar gráficamente una o varias métricas, seleccione la casilla de verificación junto a cada métrica. Para seleccionar todas las métricas, seleccione la casilla de verificación en la fila de encabezado de la tabla.
6. (Opcional) Para cambiar el tipo de gráfico, elija la pestaña Opciones. A continuación, puede elegir entre un gráfico de líneas, de áreas apiladas, una visualización numérica, un indicador, un gráfico de barras o un gráfico circular.
7. Elija la pestaña Métricas diagramadas.
8. (Opcional) Para cambiar la estadística utilizada en el gráfico, elija la nueva estadística en la columna Statistic (Estadística) junto al nombre de métrica.

   Para obtener más información acerca de las métricas de CloudWatch, consulte Definiciones de estadísticas de CloudWatch. Para obtener más información acerca de las pxx percentile statistics, see Percentiles. (estadísticas del percentil pxx, consulte .)
9. (Opcional) Para añadir una banda de detección de anomalías que muestre los valores esperados para la métrica, elija el icono de detección de anomalías en Actions (Acciones) junto a la métrica.



CloudWatch utiliza hasta dos semanas de datos históricos recientes de la métrica para calcular un modelo para los valores esperados. A continuación, muestra el rango de valores esperados como una franja en el gráfico. CloudWatch agrega una nueva fila debajo de la métrica para mostrar la expresión matemática de la franja de detección de anomalías, denominada ANOMALY_DETECTION_BAND. Si existen datos históricos recientes, podrá ver inmediatamente una vista previa de la banda de detección de anomalías, que es una aproximación de la banda de detección de anomalías generada por el modelo. La banda de detección de anomalías real tarda hasta 15 minutos en aparecer.

De forma predeterminada, CloudWatch crea los límites superiores e inferiores de la franja de valores esperados con un valor predeterminado de 2 para el umbral de la banda. Para cambiar este número, cambie el valor al final de la fórmula bajo Detailes (Detalles) para la banda.

- (Opcional) Elija Edit model (Editar modelo) para cambiar la forma en que se calcula el modelo de detección de anomalías. Puede excluir periodos de tiempo pasados y futuros del entrenamiento para calcular el modelo. Es fundamental que excluya eventos inusuales del sistema, como interrupciones del sistema, implementaciones y días festivos, de los datos de entrenamiento. También puede especificar la zona horaria que se utilizará para el modelo para los cambios de horario de verano.

Para obtener más información, consulte Uso de la detección de anomalías de CloudWatch.

Para ocultar el modelo del gráfico, elimine la marca de verificación de la línea con la función `ANOMALY_DETECTION_BAND` o elija el icono `X`. Para suprimir el modelo por completo, elija Edit model (Editar modelo), y Delete model (Eliminar modelo).

10. (Opcional) Cuando elija métricas para representarlas gráficamente, especifique una etiqueta dinámica para que aparezca en la leyenda del gráfico de cada métrica. Las etiquetas dinámicas muestran una estadística acerca de la métrica y se actualizan automáticamente cuando se actualiza el panel o gráfico. Para añadir una etiqueta dinámica, elija Métricas representadas gráficamente, Etiquetas dinámicas.

    De forma predeterminada, los valores dinámicos que añade a la etiqueta aparecen al principio de la etiqueta. A continuación, puede elegir el valor de Label (Etiqueta) de la métrica para editar la etiqueta. Para obtener más información, consulte Uso de etiquetas dinámicas.
11. Si desea ver más información acerca de la métrica que se representa gráficamente, sitúe el cursor sobre la leyenda.

12. Las anotaciones horizontales pueden ayudar a los usuarios del gráfico a saber rápidamente cuándo una métrica ha aumentado hasta un determinado nivel o si la métrica está dentro de un intervalo predefinido. Para añadir una anotación horizontal, elija Opciones de gráfico y después Añadir anotación horizontal:

    a. En Label (Etiqueta), escriba la etiqueta de la anotación.
    b. En Value (Valor), escriba el valor de métrica en el que aparece la anotación horizontal.
    c. En Fill, especifique si se usará sombreado de relleno con esta anotación. Por ejemplo, elija `Above` o `Below` para el área correspondiente que se rellenará. Si especifica `Between`, aparece otro campo `Value` y se rellena el área del gráfico entre los dos valores.
    d. En Axis, especifique si los números de `Value` hacen referencia a la métrica asociada con el eje Y izquierdo o con el eje Y derecho, en caso de que el gráfico incluya varias métricas.

       Puede cambiar el color de relleno de una anotación eligiendo el cuadrado de color en la columna izquierda de la anotación.

    Repita estos pasos para agregar varias anotaciones horizontales al mismo gráfico.

    Para ocultar una anotación, quite la marca de la casilla en la columna izquierda de dicha anotación.

    Para eliminar una anotación, elija x en la columna Actions.
13. Para obtener una URL para el gráfico, elija Actions, Share. Copie la URL para guardarla o compartirla.
14. Para agregar su gráfico a un panel, elija Actions, Add to dashboard.

Creación de un gráfico de métricas a partir de otro origen de datos

Puede crear un gráfico que muestre los recursos de orígenes de datos distintos de CloudWatch. Para obtener más información acerca de la creación de conexiones con los otros orígenes de datos, consulte Consulta de métricas de otros orígenes de datos.

Cómo crear una métrica a partir de otro origen de datos

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, seleccione Métricas y, a continuación, Todas las métricas.
3. Seleccione la pestaña Consulta de múltiples orígenes.

4. En Origen de datos, seleccione el origen de datos que desee utilizar.

   Si aún no ha creado una conexión al origen de datos que desea, seleccione Crear y administrar orígenes de datos y, a continuación, elija Crear y administrar orígenes de datos. Para obtener información sobre el resto del proceso de creación del origen de datos, consulte Conéctese a un origen de datos prediseñado con un asistente.

5. El asistente o el editor de peticiones le solicitarán la información necesaria para la consulta. El flujo de trabajo es diferente para cada origen de datos y se adapta a cada origen de datos. Por ejemplo, para Amazon Managed Service para Prometheus y los orígenes de datos de Prometheus, aparece un cuadro del editor de consultas de ProMQL con un asistente de consultas.

6. Cuando haya terminado de crear la consulta, elija Consulta de gráficos.

   El gráfico se rellena con las métricas de la consulta.

7. (Opcional) Las anotaciones horizontales pueden ayudar a los usuarios del gráfico a saber rápidamente cuándo una métrica ha aumentado hasta un determinado nivel o si la métrica está dentro de un intervalo predefinido. Para añadir una anotación horizontal, elija Opciones de gráfico y después Añadir anotación horizontal:

   a. En Label (Etiqueta), escriba la etiqueta de la anotación.

   b. En Value (Valor), escriba el valor de métrica en el que aparece la anotación horizontal.

   c. En Fill, especifique si se usará sombreado de relleno con esta anotación. Por ejemplo, elija `Above` o `Below` para el área correspondiente que se rellenará. Si especifica `Between`, aparece otro campo `Value` y se rellena el área del gráfico entre los dos valores.

   d. En Axis, especifique si los números de `Value` hacen referencia a la métrica asociada con el eje Y izquierdo o con el eje Y derecho, en caso de que el gráfico incluya varias métricas.

      Puede cambiar el color de relleno de una anotación eligiendo el cuadrado de color en la columna izquierda de la anotación.

   Repita estos pasos para agregar varias anotaciones horizontales al mismo gráfico.

   Para ocultar una anotación, quite la marca de la casilla en la columna izquierda de dicha anotación.

   Para eliminar una anotación, elija x en la columna Actions.

8. (Opcional) Para agregar este gráfico a un panel, elija Acciones, Añadir al panel.

Actualización de un gráfico

Para actualizar el gráfico

1. Para cambiar el nombre del gráfico, seleccione el icono de lápiz.
2. Para cambiar el intervalo de tiempo, seleccione uno de los valores predefinidos o elija custom (personalizado). Para obtener más información, consulte Modificar el intervalo de tiempo o el formato de zona horaria de un gráfico.
3. Para cambiar la estadística, elija la pestaña Graphed metrics. Elija el encabezado de columna o un valor individual y, a continuación, elija una de las estadísticas o percentiles predefinidos o especifique un percentil personalizado (por ejemplo, **`p95.45`**).
4. Para cambiar el periodo, elija la pestaña Graphed metrics. Elija el encabezado de columna o un valor individual y, a continuación, elija un valor diferente.
5. Para añadir una anotación horizontal, elija Graph options (Opciones de gráfico) y después Add horizontal annotation (Añadir anotación horizontal):

   a. En Label (Etiqueta), escriba la etiqueta de la anotación.

   b. En Value (Valor), escriba el valor de métrica en el que aparece la anotación horizontal.

   c. En Fill, especifique si se usará sombreado de relleno con esta anotación. Por ejemplo, elija `Above` o `Below` para el área correspondiente que se rellenará. Si especifica `Between`, aparece otro campo `Value` y se rellena el área del gráfico entre los dos valores.

   d. En Axis (Eje), especifique si los números de `Value` hacen referencia a la métrica asociada con el eje Y izquierdo o con el eje Y derecho, en caso de que el gráfico incluya varias métricas.

      Puede cambiar el color de relleno de una anotación eligiendo el cuadrado de color en la columna izquierda de la anotación.

   Repita estos pasos para agregar varias anotaciones horizontales al mismo gráfico.

   Para ocultar una anotación, quite la marca de la casilla en la columna izquierda de dicha anotación.

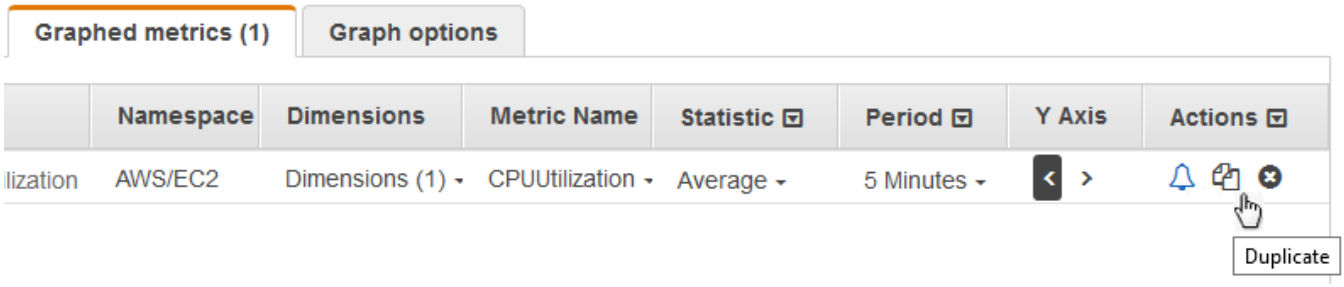
   Para eliminar una anotación, elija x en la columna Actions.

6. Para cambiar el intervalo de actualización, elija Refresh options (Actualizar opciones) y, a continuación, seleccione Auto refresh (Actualizar automáticamente) o elija 1 Minute (1 minuto), 2 Minutes (2 minutos), 5 Minutes (5 minutos) o 15 Minutes (15 minutos).

Duplicación de una métrica

Para duplicar una métrica

1. Elija la pestaña Métricas diagramadas.
2. En Actions, seleccione el icono Duplicate.



3. Actualice la métrica duplicada según sea necesario.

Combinar dos gráficos en uno solo

Puede combinar dos gráficos diferentes en uno y, a continuación, el gráfico resultante mostrará ambas métricas. Esto puede resultar útil si ya se muestran diferentes métricas en distintos gráficos y quiere combinarlas, o si desea crear fácilmente un único gráfico con métricas de distintas regiones.

Para combinar un gráfico con otro, utilice la URL o la fuente JSON del gráfico que desee combinar.

Combinar dos gráficos en uno

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. Abra el gráfico que desee combinar con otro gráfico. Para ello, puede elegir Métricas, Todas las métricas y, a continuación, elegir una métrica para graficar. O bien, puede abrir un panel y, a continuación, abrir uno de los gráficos del panel seleccionando el gráfico y eligiendo Abrir en métricas el menú de la parte superior derecha del gráfico.
3. Una vez que tenga abierto un gráfico, realice una de las siguientes acciones:
   - Copie la URL de la barra del navegador.
   - Elija la pestaña Origen y, a continuación, seleccione Copiar.

4. Abra el gráfico con el que desea combinar el gráfico anterior.
5. Cuando tenga el segundo gráfico abierto en la vista de Métricas, seleccione Acciones y Combinar gráfico.
6. Introduzca la URL o el JSON que copió anteriormente y seleccione Combinar.
7. Aparecen los gráficos combinados. El eje Y de la izquierda corresponde al gráfico original y el eje Y de la derecha corresponde al gráfico que ha fusionado en él.

> **Note**
>
> Si el gráfico en el que se ha fusionado utiliza la función METRICS (), las métricas del gráfico que se ha fusionado no se incluyen en el cálculo de METRICS () del gráfico combinado.

8. Para agregar su gráfico a un panel, elija Acciones, Agregar al panel.

Uso de etiquetas dinámicas

Puede utilizar etiquetas dinámicas con sus gráficos. Las etiquetas dinámicas añaden un valor actualizado de forma dinámica a la etiqueta para la métrica seleccionada. Puede agregar un amplio intervalo de valores a las etiquetas, tal y como se muestra en las siguientes tablas.

El valor dinámico mostrado en la etiqueta se obtiene del intervalo de tiempo que se muestra actualmente en el gráfico. La parte dinámica de la etiqueta se actualiza automáticamente cuando se actualiza el panel o el gráfico.

Si utiliza una etiqueta dinámica con una expresión de búsqueda, la etiqueta dinámica se aplica a cada métrica devuelta por la búsqueda.

Puede utilizar la consola de CloudWatch para añadir un valor dinámico a una etiqueta, editar la etiqueta, cambiar la posición del valor dinámico dentro de la columna de la etiqueta y realizar otras opciones de personalización.

Etiquetas dinámicas

Dentro de una etiqueta dinámica, puede utilizar los siguientes valores relacionados con las propiedades de la métrica:

Valor activo de la etiqueta dinámica	Descripción
${AVG}	El promedio de los valores en el intervalo de tiempo mostrado actualmente en el gráfico.
${DATAPOINT_COUNT}	El número de puntos de datos en el intervalo de tiempo que se muestra actualmente en el gráfico.
${FIRST}	El más antiguo de los valores de la métrica en el rango de tiempo que se muestra actualmente en el gráfico.
${FIRST_LAST_RANGE}	La diferencia entre los valores de la métrica de los puntos de datos más antiguos y más recientes que se muestran actualmen te en el gráfico.
${FIRST_LAST_TIME_ RANGE}	El intervalo de tiempo absoluto entre los puntos de datos más antiguos y más recientes que se muestran actualmente en el gráfico.
${FIRST_TIME}	La marca de tiempo del punto de datos más antiguo del intervalo de tiempo que se muestra actualmente en el gráfico.
${FIRST_TIME_RELATIVE}	La diferencia de tiempo absoluta entre ahora y la marca de tiempo del punto de datos más antiguo en el intervalo de tiempo que se muestra actualmente en el gráfico.
${LABEL}	Representación de la etiqueta predeterminada para una métrica.
${LAST}	El más reciente de los valores en el intervalo de tiempo que se muestra actualmente en el gráfico.
${LAST_TIME}	La marca de tiempo del punto de datos más reciente en el intervalo de tiempo que se muestra actualmente en el gráfico.
${LAST_TIME_RELATIVE}	La diferencia de tiempo absoluta entre el momento actual y la marca de tiempo del punto de datos más reciente en el intervalo de tiempo que se muestra actualmente en el gráfico.

Valor activo de la etiqueta dinámica	Descripción
${MAX}	El máximo de los valores en el intervalo de tiempo mostrado actualmente en el gráfico.
${MAX_TIME}	La marca de tiempo del punto de datos que tiene el valor de métrica más alto, de los puntos de datos que se muestran actualmente en el gráfico.
${MAX_TIME_RELATIVE}	La diferencia de tiempo absoluta entre el momento actual y la marca de tiempo del punto de datos con el valor más alto, de aquellos puntos de datos que se muestran actualmente en el gráfico.
${MIN}	El mínimo de los valores en el intervalo de tiempo mostrado actualmente en el gráfico.
${MIN_MAX_RANGE}	La diferencia en los valores de métrica entre los puntos de datos con los valores de métrica más altos y más bajos, de los puntos de datos que se muestran actualmente en el gráfico.
${MIN_MAX_TIME_RANGE}	La diferencia de tiempo absoluta entre los puntos de datos con los valores métricos más altos y más bajos, de los puntos de datos que se muestran actualmente en el gráfico.
${MIN_TIME}	La marca de tiempo del punto de datos que tiene el valor de métrica más bajo, de los puntos de datos que se muestran actualmente en el gráfico.
${MIN_TIME_RELATIVE}	La diferencia de tiempo absoluta entre el momento actual y la marca de tiempo del punto de datos con el valor más bajo, de aquellos puntos de datos que se muestran actualmente en el gráfico.
${PROP('AccountId')}	La cuenta de la ID de AWS de la métrica.
${PROP('AccountLabel')}	La etiqueta especificada para la cuenta de origen que posee esta métrica, en la observabilidad entre cuentas de CloudWatch.

Valor activo de la etiqueta dinámica	Descripción
${PROP('Dim.*dimension_name*')}	El valor de la dimensión especificada. Sustituya *dimension_name* por el nombre de la dimensión, con distinción entre mayúsculas y minúsculas.
${PROP('MetricName')}	El nombre de la métrica.
${PROP('Namespace')}	El nombre del espacio de la métrica.
${PROP('Period')}	Período de la métrica en segundos.
${PROP('Region')}	La Región de AWS en la que se publica la métrica.
${PROP('Stat')}	La estadística de la métrica que se está graficando.
${SUM}	La suma de los valores en el intervalo de tiempo mostrado actualmente en el gráfico.

Suponga, por ejemplo, que tiene una expresión de búsqueda **`SEARCH(' {AWS/Lambda, FunctionName} Errors ', 'Sum')`**, que busca `Errors` para cada una de las funciones de Lambda. Si establece la etiqueta en `[max: ${MAX} Errors for Function Name ${LABEL}]`, la etiqueta de cada métrica es [max: ***número*** Errors for Function Name (Errores del nombre de función) ***Nombre***].

Puede agregar hasta seis valores dinámicos a una etiqueta. Puede utilizar el marcador de posición `${LABEL}` solo una vez dentro de cada etiqueta.

Modificar el intervalo de tiempo o el formato de zona horaria de un gráfico

En esta sección, se describe cómo modificar el formato de fecha, hora y zona horaria en un gráfico de métricas de CloudWatch. También se describe cómo agrandar un gráfico para aplicar un intervalo de tiempo específico. Para obtener información acerca de cómo crear un gráfico, consulte Representar gráficamente una métrica.

> **Note**
>
> Si el intervalo de tiempo de un panel es más corto que el periodo utilizado para un gráfico en el panel, ocurre lo siguiente:
>
> - El gráfico se modifica para mostrar la cantidad de datos correspondiente a un periodo completo para ese widget, aunque sea más largo que el intervalo de tiempo del panel. Esto garantiza que haya al menos un punto de datos en el gráfico.
> - La hora de inicio del periodo de este punto de datos se ajusta hacia atrás para garantizar que se pueda mostrar al menos un punto de datos.

Configurar un intervalo de tiempo relativo

New interface

Para especificar un intervalo de tiempo relativo para un gráfico

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Metrics (Métricas) y, a continuación, All metrics (Todas las métricas). En la esquina superior derecha de la pantalla, puede seleccionar uno de los intervalos de tiempo predefinidos, que van desde 1 hora hasta 1 semana (1 h, 3 h, 12 h, 1 día, 3 días o 1 sem.). También puede elegir Custom (Personalizado) para establecer su propio intervalo de tiempo.
3. Elija Custom (Personalizado) y, a continuación, seleccione la pestaña Relative (Relativo) en la esquina superior izquierda del cuadro. Puede especificar un intervalo de tiempo en minutos, horas, días, semanas o meses.
4. Después de especificar un intervalo de tiempo, seleccione Apply (Aplicar).

Original interface

Para especificar un intervalo de tiempo relativo para un gráfico

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Metrics (Métricas) y, a continuación, All metrics (Todas las métricas). En la esquina superior derecha de la pantalla, puede seleccionar uno de los intervalos de tiempo predefinidos, que van desde 1 hora hasta 1 semana (1 h, 3 h, 12 h,

1 día, 3 días o 1 sem.). También puede elegir Custom (Personalizado) para establecer su propio intervalo de tiempo.

3. Elija Custom (Personalizado) y, a continuación, Relative (Relativo) en la esquina superior izquierda del cuadro. Puede especificar un intervalo de tiempo en minutos, horas, días, semanas o meses.

Configurar un intervalo de tiempo absoluto

New interface

Para especificar un intervalo de tiempo absoluto para un gráfico

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Metrics (Métricas) y, a continuación, All metrics (Todas las métricas). En la esquina superior derecha de la pantalla, puede seleccionar uno de los intervalos de tiempo predefinidos, que van desde 1 hora hasta 1 semana (1 h, 3 h, 12 h, 1 día, 3 días o 1 sem.). También puede elegir Custom (Personalizado) para establecer su propio intervalo de tiempo.
3. Elija Custom (Personalizado) y, a continuación, seleccione la pestaña Absolute (Absoluto) en la esquina superior izquierda del cuadro. Utilice el selector de calendario o los cuadros de campos de texto para especificar el intervalo de tiempo.
4. Después de especificar un intervalo de tiempo, seleccione Apply (Aplicar).

Original interface

Para especificar un intervalo de tiempo absoluto para un gráfico

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Metrics (Métricas) y, a continuación, All metrics (Todas las métricas). En la esquina superior derecha de la pantalla, puede seleccionar uno de los intervalos de tiempo predefinidos, que van desde 1 hora hasta 1 semana (1 h, 3 h, 12 h, 1 día, 3 días o 1 sem.). También puede elegir Custom (Personalizado) para establecer su propio intervalo de tiempo.
3. Elija Custom (Personalizado) y, a continuación, Absolute (Absoluto) en la esquina superior izquierda del cuadro. Utilice el selector de calendario o los cuadros de campos de texto para especificar el intervalo de tiempo.

4. Después de especificar un intervalo de tiempo, seleccione Apply (Aplicar).

Configurar el formato de la zona horaria

New interface

Para especificar la zona horaria de un gráfico

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Metrics (Métricas) y, a continuación, All metrics (Todas las métricas). En la esquina superior derecha de la pantalla, puede seleccionar uno de los intervalos de tiempo predefinidos, que van desde 1 hora hasta 1 semana (1 h, 3 h, 12 h, 1 día, 3 días o 1 sem.). También puede elegir Custom (Personalizado) para establecer su propio intervalo de tiempo.
3. Elija Custom (Personalizado) y, a continuación, seleccione el menú desplegable ubicado en la esquina superior derecha del cuadro. Puede cambiar la zona horaria a UTC (UTC) o Local time zone (Zona horaria local).
4. Después de efectuar los cambios, seleccione Apply (Aplicar).

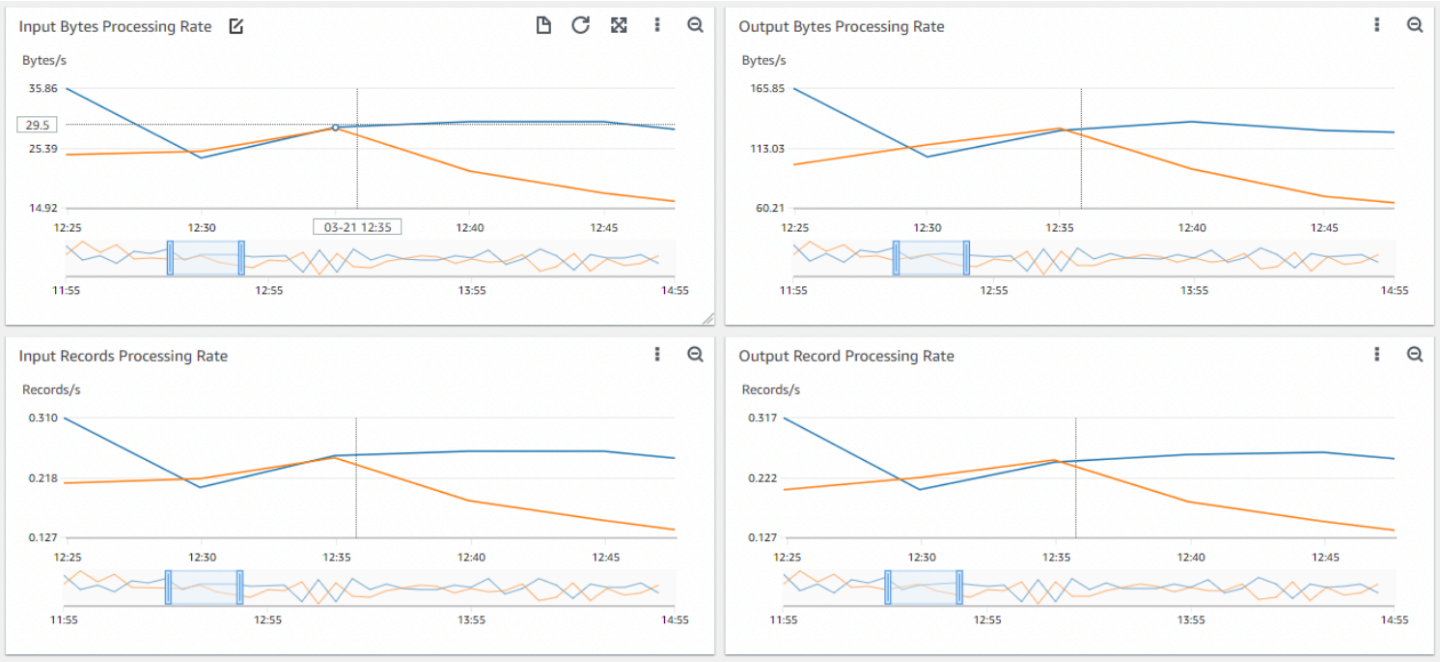
Original interface

Para especificar la zona horaria de un gráfico

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Metrics (Métricas) y, a continuación, All metrics (Todas las métricas). En la esquina superior derecha de la pantalla, puede seleccionar uno de los intervalos de tiempo predefinidos, que van desde 1 hora hasta 1 semana (1 h, 3 h, 12 h, 1 día, 3 días o 1 sem.). También puede elegir Custom (Personalizado) para establecer su propio intervalo de tiempo.
3. Elija Custom (Personalizado) y, a continuación, seleccione el menú desplegable ubicado en la esquina superior derecha del cuadro. Puede cambiar la zona horaria a UTC (UTC) o Local time zone (Zona horaria local).

Ampliar un gráfico de línea o un gráfico de área apilada

En la consola de CloudWatch, puede utilizar la característica de zoom de minimapa para centrarse en secciones de gráficos de línea y de área apilada sin cambiar entre vistas ampliadas y alejadas. Por ejemplo, puede utilizar la característica de zoom de minimapa para centrarse en un pico de un gráfico de línea, de modo que pueda comparar el pico con otras métricas del panel desde la misma línea de tiempo. En los procedimientos de esta sección, se describe cómo utilizar la característica de zoom.



En la imagen anterior, la característica de zoom se centra en un pico de un gráfico de línea relacionado con la velocidad de procesamiento de bytes de entrada, mientras que también muestra otros gráficos de línea en el panel que se centran en secciones de la misma línea de tiempo.

New interface

Para ampliar un gráfico

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Metrics (Métricas) y, a continuación, All metrics (Todas las métricas).
3. Elija Browse (Navegar) y, a continuación, seleccione una o más métricas para graficar.
4. Elija Options (Opciones) y, luego, seleccione Line (Línea) en Widget type (Tipo de widget).
5. Elija y arrastre el área del gráfico en la que desea centrarse y, a continuación, suéltela.

6. Para restablecer el tamaño del gráfico, elija el icono Reset zoom (Restablecer el zoom) que parece una lupa con un símbolo menos (-) en su interior.

Original interface

Para ampliar un gráfico

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Metrics (Métricas) y, a continuación, All metrics (Todas las métricas).
3. Elija All metrics (Todas las métricas) y, a continuación, seleccione una métrica para graficar.
4. Elija Graph options (Opciones de gráficos). En Widget type (Tipo de widget), seleccione Line (Línea).
5. Elija y arrastre el área del gráfico en la que desea centrarse y, a continuación, suéltela.
6. Para restablecer el tamaño del gráfico, elija el icono Reset zoom (Restablecer el zoom) que parece una lupa con un símbolo menos (-) en su interior.

> **Tip**
>
> Si ya creó un panel que contiene un gráfico de línea o de área apilada, puede ir al panel y empezar a utilizar la característica de zoom.

Modificar el eje Y de un gráfico

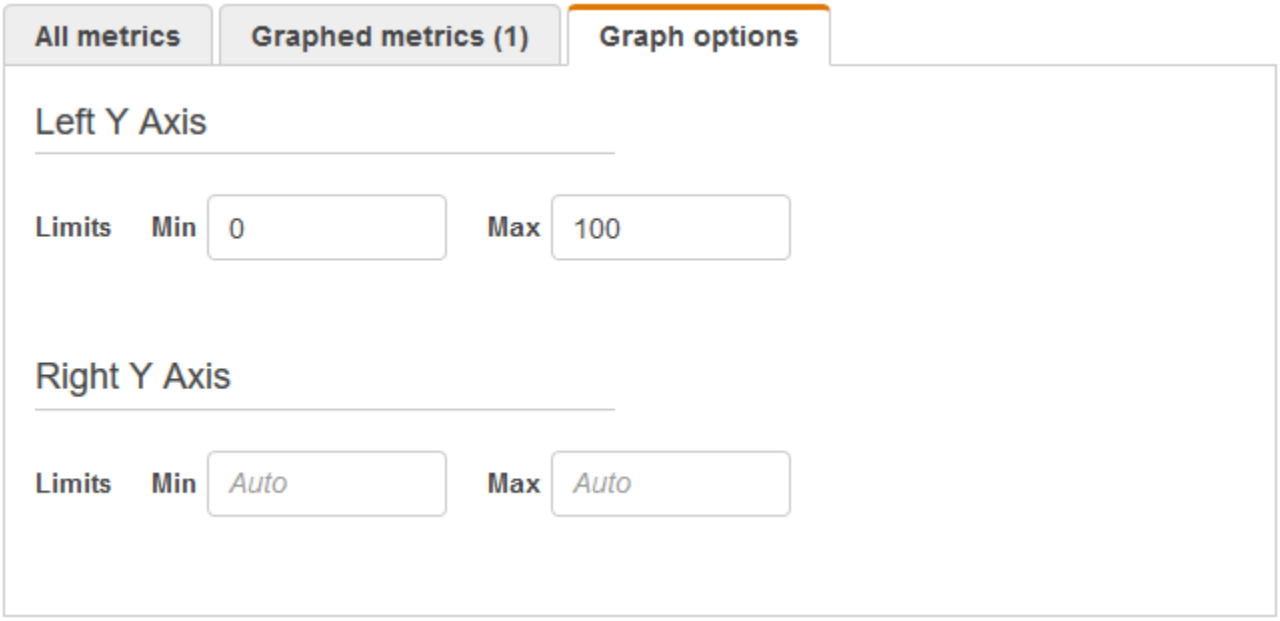
Puede definir límites personalizados para el eje Y en un gráfico para ayudarle a ver los datos con más claridad. Por ejemplo, puede cambiar los límites en un gráfico `CPUUtilization` al 100 % para que sea más fácil ver si el uso de CPU es bajo (la línea representada está cerca de la parte inferior del gráfico) o alto (la línea representada está cerca de la parte superior del gráfico).

Puede cambiar entre dos ejes Y distintos de su gráfico. Esto resulta útil si el gráfico contiene métricas que tengan distintas unidades o que difieran ampliamente en su intervalo de valores.

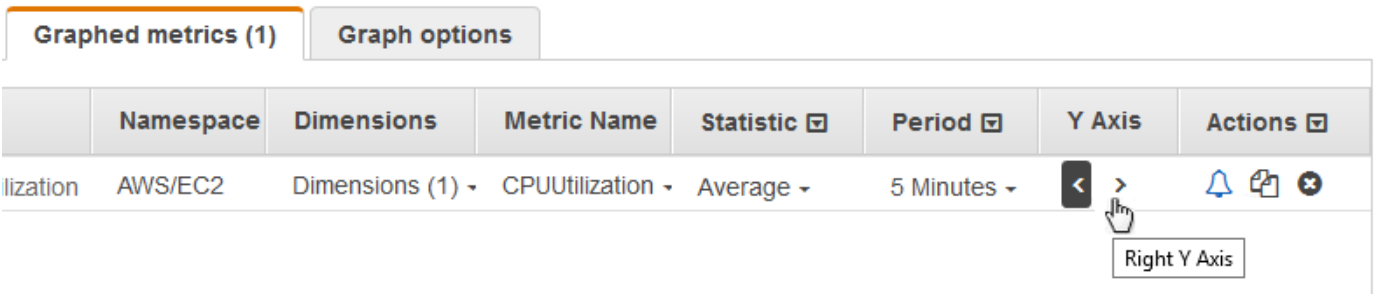
Para modificar el eje Y de un gráfico

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.

2. En el panel de navegación, seleccione Métricas.
3. Seleccione un espacio de nombres de métricas (por ejemplo, EC2) y, a continuación, una dimensión de métrica (por ejemplo, Per-Instance Metrics [Métricas por instancia]).
4. La pestaña All metrics muestra todas las métricas para dicha dimensión en ese espacio de nombres. Para representar gráficamente una métrica, active la casilla de verificación situada junto a ella.
5. En la pestaña Graph options, especifique los valores Min y Max para Left Y Axis. El valor de Min (Mínimo) no puede ser mayor que el valor de Max (Máximo).



6. Para crear un segundo eje Y, especifique los valores Min (Mínimo) y Max (Máximo) para Right Y Axis (Eje Y derecho).
7. Para alternar los dos ejes Y, elija la pestaña Graphed metrics (Métricas diagramadas). En Y Axis, elija Left Y Axis o Right Y Axis.

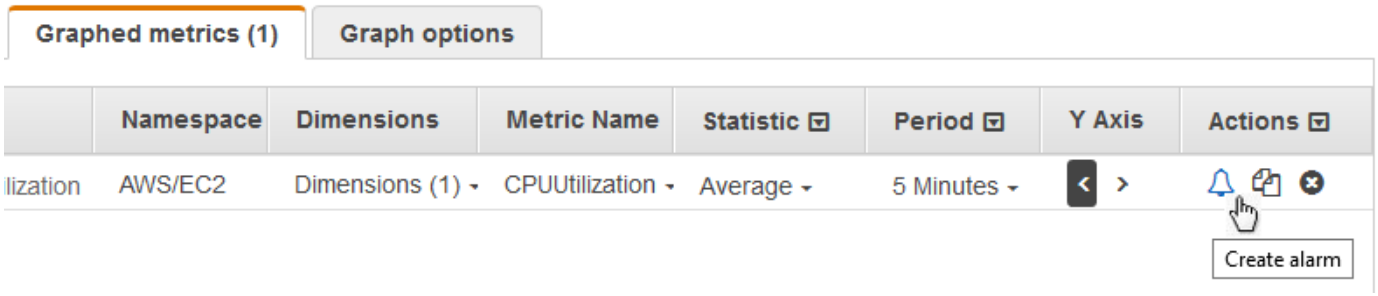


Crear una alarma desde una métrica en un gráfico

Puede representar gráficamente una métrica y, a continuación, crear una alarma desde la métrica en el gráfico, que tiene el beneficio de rellenar muchos de los campos de alarma.

Para crear una alarma desde una métrica en un gráfico

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, seleccione Métricas.
3. Seleccione un espacio de nombres de métricas (por ejemplo, EC2) y, a continuación, una dimensión de métrica (por ejemplo, Per-Instance Metrics [Métricas por instancia]).
4. La pestaña All metrics muestra todas las métricas para dicha dimensión en ese espacio de nombres. Para representar gráficamente una métrica, active la casilla de verificación situada junto a ella.
5. Para crear una alarma para la métrica, elija la pestaña Graphed metrics. En Actions, seleccione el icono de alarma.



6. En Conditions (Condiciones), seleccione Static (Estático) o Anomaly detection (Detección de anomalías) para especificar si desea utilizar un umbral estático o un modelo de detección de anomalías para la alarma.

   En función de la opción que elija, especifique el resto de los datos para las condiciones de la alarma.
7. Elija Configuración adicional. Para Puntos de datos para alarma, especifique el número de periodos de evaluación (puntos de datos) que deben tener el estado `ALARM` para que se active la alarma. Si estos dos valores coinciden, creará una alarma que pasará al estado `ALARM` si se infringen muchos periodos consecutivos.

   Para crear una alarma M de N, especifique un número menor para el primer valor que el especificado para el segundo valor. Para obtener más información, consulte Evaluación de una alarma.

8. En Missing data treatment (Tratamiento de datos que faltan), elija cómo debe comportarse la alarma cuando falten algunos puntos de datos. Para obtener más información, consulte Configuración de la forma en la que las alarmas de CloudWatch tratan los datos que faltan.
9. Elija Siguiente.
10. En Notification (Notificación), seleccione el tema de SNS al que desee enviar la notificación cuando la alarma tenga el estado `ALARM`, `OK` o `INSUFFICIENT_DATA`.

    Para que la alarma envíe varias notificaciones para el mismo estado de alarma o para estados de alarma diferentes, seleccione Add notificación (Añadir notificación).

    Para que la alarma no envíe notificaciones, elija Remove (Eliminar).
11. Para que la alarma realice el Auto Scaling o acciones de EC2, elija el botón correspondiente, y elija el estado de alarma y la acción que se debe realizar.
12. Cuando haya terminado, elija Next (Siguiente).
13. Escriba un nombre y la descripción de la alarma. El nombre solo debe contener caracteres ASCII. A continuación, elija Next.
14. En Preview and create (Obtener vista previa y crear), confirme que la información y las condiciones son las que desea y, a continuación, elija Create alarm (Crear alarma).

Uso de la detección de anomalías de CloudWatch

Cuando habilita la detección de anomalías para una métrica, CloudWatch aplica algoritmos estadísticos y del machine learning. Estos algoritmos analizan continuamente las métricas de sistemas y aplicaciones, determinan los valores de referencia normales y detectan anomalías con una intervención mínima del usuario.

Los algoritmos generan un modelo de detección de anomalías. El modelo genera un intervalo de valores esperados que representan el comportamiento normal de la métrica.

Puede habilitar la detección de anomalías utilizando la AWS Management Console, la AWS CLI, AWS CloudFormation o el SDK de AWS. Puede habilitar la detección de anomalías en las métricas ofrecidas por AWS y también en métricas personalizadas. En una cuenta configurada como cuenta de supervisión para la observabilidad entre cuentas de CloudWatch, puede crear detectores de anomalías en las métricas de las cuentas de origen, además de las métricas de la cuenta de supervisión.

Puede utilizar el modelo de valores esperados de dos formas:

- Puede crear alarmas de detección de anomalías basadas en el valor esperado de una métrica. Estos tipos de alarmas no tienen un umbral estático para determinar el estado de la alarma. En lugar de ello, comparan el valor de la métrica con el valor esperado en función del modelo de detección de anomalías.

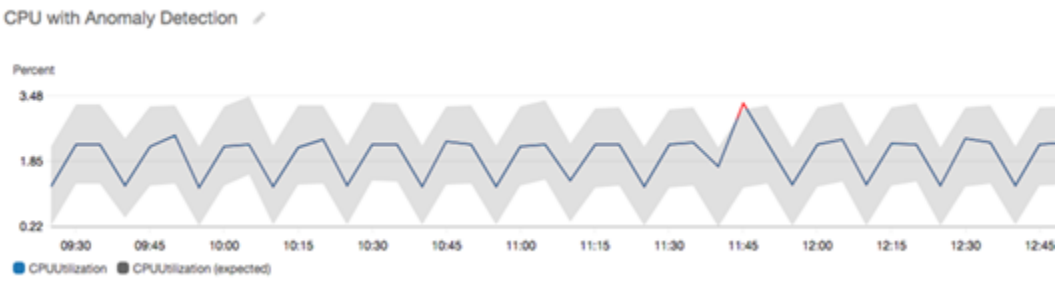
  Puede elegir si la alarma se activa cuando el valor de la métrica está por encima del intervalo de valores previstos, por debajo del intervalo, o bien por encima o por debajo del intervalo.

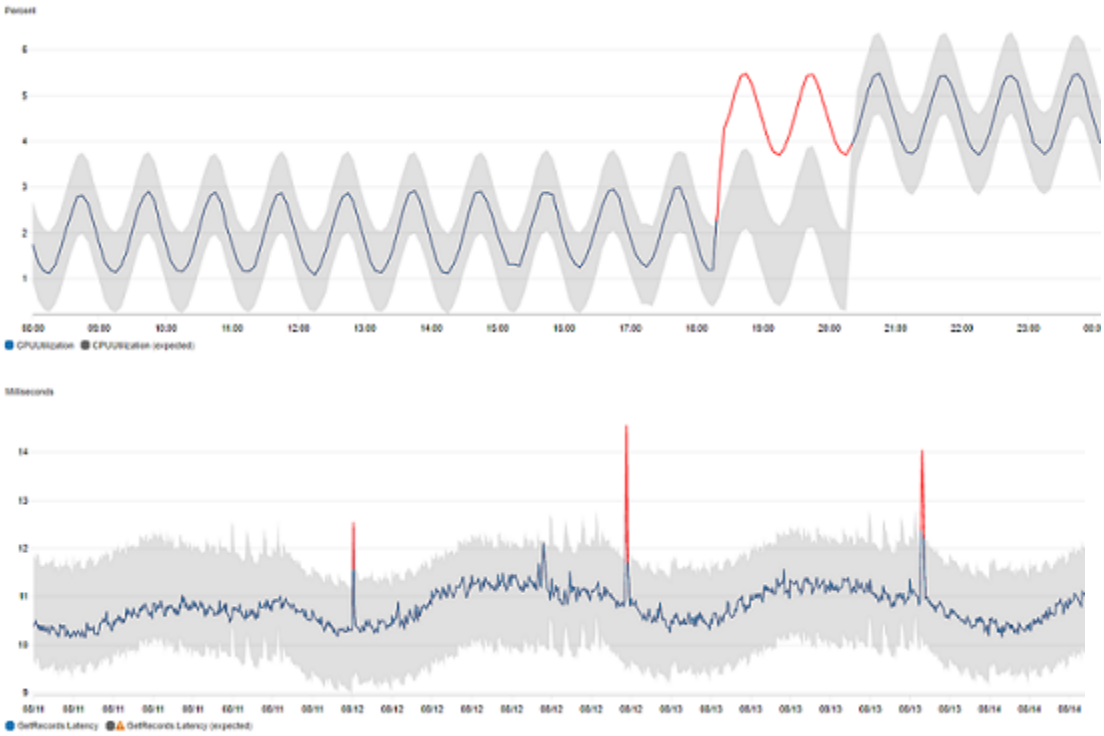
  Para obtener más información, consulte Crear una alarma de CloudWatch en función de la detección de anomalías.
- Cuando consulte un gráfico de datos de métricas, superponga los valores esperados en el gráfico en forma de banda. Esto permite ver claramente qué valores del gráfico están fuera del intervalo normal. Para obtener más información, consulte Creación de un gráfico.

  También puede recuperar los valores superior e inferior de la banda del modelo utilizando la solicitud de la API `GetMetricData` con la función matemática de la métrica `ANOMALY_DETECTION_BAND`. Para obtener más información, consulte GetMetricData.

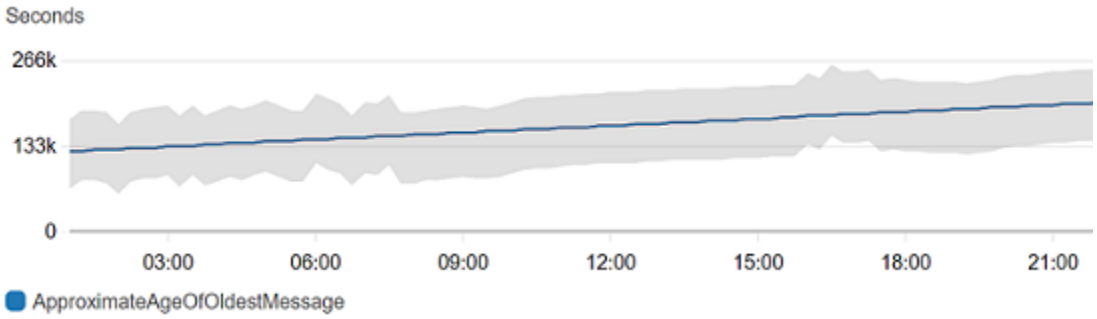
En un gráfico con detección de anomalías, el intervalo esperado de valores se muestra como una banda gris. Si el valor real de la métrica está fuera de esta banda, se muestra en rojo durante ese tiempo.

Los algoritmos de detección de anomalías dan cuenta de la estacionalidad y los cambios de tendencia de las métricas. Los cambios de estacionalidad pueden ser por hora, por día o por semana, como se muestra en los siguientes ejemplos.

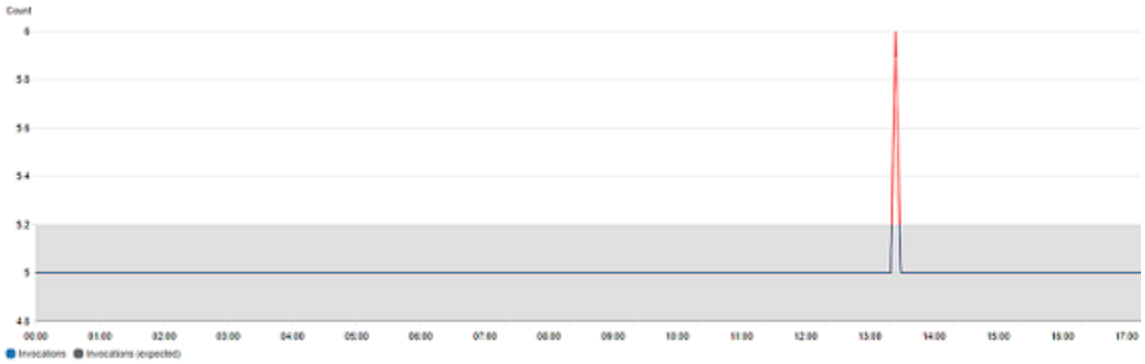




Las tendencias a más largo plazo podrían ser a la baja o al alza.



Las detecciones de anomalías también funcionan bien con métricas con patrones planos.



Descubra cómo funciona la detección de anomalías de CloudWatch

Al habilitar la detección de anomalías de una métrica, CloudWatch aplica algoritmos del machine learning a los datos anteriores de la métrica para crear un modelo de los valores esperados de la métrica. El modelo evalúa las tendencias y los patrones horarios, diarios y semanales de la métrica. El algoritmo se entrena con hasta dos semanas de datos de métricas, pero puede habilitar la detección de anomalías en una métrica aunque la métrica no tenga dos semanas completas de datos.

Especifique un valor para el umbral de detección de anomalías que CloudWatch utiliza junto con el modelo para determinar el intervalo ‘normal’ de los valores de la métrica. Un valor mayor del umbral de detección de anomalías produce un intervalo mayor de valores "normales".

El modelo de machine learning es específico de una métrica y una estadística. Por ejemplo, si habilita la detección de anomalías de una métrica utilizando la estadística `AVG`, el modelo es específica de la estadística `AVG`.

Cuando CloudWatch crea un modelo para muchas métricas comunes de AWS, asegura que la banda no se extienda fuera de los valores lógicos. Por ejemplo, la banda `MemoryUtilization` de una instancia de EC2 se mantendrá entre 0 y 100, y las bandas que rastrean `Requests` de CloudFront, que no pueden ser negativas, nunca se extenderán por debajo de cero.

Después de crear un modelo, la detección de anomalías de CloudWatch evalúa continuamente el modelo y realiza ajustes para garantizar que sea lo más preciso posible. Esto incluye volver a formar el modelo para ajustar si los valores de la métrica evolucionan con el tiempo o sufren cambios repentinos, y también incluye indicadores para mejorar los modelos de métricas estacionales, con picos o dispersas.

Después de habilitar la detección de anomalías en una métrica, tiene la opción de excluir periodos de tiempo específicos de la métrica para entrenar el modelo. De esta forma, puede excluir las implementaciones u otros eventos inusuales para la capacitación de modelos, garantizando la creación del modelo más preciso.

El uso de modelos de detección de anomalías para las alarmas implica cargos en su cuenta de AWS. Para obtener más información, consulte Precios de Amazon CloudWatch.

Detección de anomalías en matemáticas de métricas

La detección de anomalías en matemáticas de métricas es una característica que se puede utilizar para crear alarmas de detección de anomalías respecto al resultado de expresiones matemáticas

de métricas. Puede utilizar estas expresiones para crear gráficos en los que se visualicen bandas de detección de anomalías. La característica admite funciones aritméticas básicas, operadores lógicos y de comparación y la mayoría de las demás funciones. Para obtener información sobre las funciones que no son compatibles, consulte Uso de matemáticas de métricas en la Guía del usuario de Amazon CloudWatch.

Puede crear modelos de detección de anomalías según expresiones matemáticas de métricas similares a la forma en que ya se crean modelos de detección de anomalías. Desde la consola de CloudWatch, puede aplicar la detección de anomalías a expresiones matemáticas de métricas y seleccionar la detección de anomalías como tipo de umbral para estas expresiones.

> **Note**
>
> La detección de anomalías en matemáticas de métricas solo se puede habilitar y editar en la última versión de la interfaz de usuario de métricas. Cuando crea detectores de anomalías según expresiones matemáticas de métricas en la nueva versión de la interfaz, puede visualizarlos en la versión anterior, pero no editarlos.

Para obtener información sobre cómo crear alarmas y modelos para la detección de anomalías y matemáticas de métricas, consulte las siguientes secciones:

- Creación de una alarma de CloudWatch basada en la detección de anomalías
- Creación de una alarma de CloudWatch basada en una expresión matemática métrica

También puede crear, eliminar y descubrir modelos de detección de anomalías según expresiones matemáticas de métricas mediante la API de CloudWatch con `PutAnomalyDetector`, `DeleteAnomalyDetector` y `DescribeAnomalyDetectors`. Para obtener información sobre estas acciones de la API, consulte las siguientes secciones en la Referencia de la API de Amazon CloudWatch.

- PutAnomalyDetector
- DeleteAnomalyDetector
- DescribeAnomalyDetectors

Para obtener información sobre el precio de las alarmas de detección de anomalías, consulte Precios de Amazon CloudWatch.

Uso de la calculadora de métricas

Las matemáticas en las métricas le permiten consultar varias métricas de CloudWatch y usar expresiones matemáticas para crear series temporales nuevas basadas en estas métricas. Puede visualizar las series temporales resultantes en la consola de CloudWatch y agregarlas a los paneles. Puede usar las métricas de AWS Lambda como ejemplo para dividir la `Errors` métrica por la métrica `Invocations` y así obtener una tasa de error. A continuación, añada las series temporales resultantes a un gráfico del panel de CloudWatch.

También puede realizar cálculos de métricas mediante programación, con la operación de la API `GetMetricData`. Para obtener más información, consulte GetMetricData.

Añadir una expresión matemática a un gráfico de CloudWatch

Puede añadir una expresión matemática a un gráfico en el panel de CloudWatch. Cada gráfico está limitado a un máximo de 500 métricas y expresiones, por lo que puede agregar una expresión matemática solo si el gráfico tiene 499 métricas como máximo. Esto se aplica incluso si no todas las métricas se muestran en el gráfico.

Para añadir una expresión matemática a un gráfico

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. Cree o edite un gráfico. Debe haber al menos una métrica en el gráfico.
3. Elija la pestaña Métricas diagramadas.
4. Elija Math expression (Expresión matemática), Start with an empty expression (Comenzar con una expresión vacía). Aparece una nueva línea para la expresión.
5. En la nueva línea, en la columna Details (Detalles) escriba la expresión matemática. En las tablas de la sección de Sintaxis y funciones matemáticas de métricas se muestran las funciones que se pueden utilizar en la expresión.

   Para usar una métrica o el resultado de otra expresión como parte de la fórmula de esta expresión, utilice el valor que se muestra en la columna Id; por ejemplo, m1+m2 o e1-MIN(e1).

   Puede cambiar el valor de Id. Puede contener números, letras y guiones bajos y debe comenzar por una letra minúscula. El cambio del valor de Id por un nombre más descriptivo puede aumentar también la legibilidad de un gráfico; por ejemplo, cambiar m1 y m2 por errores y solicitudes.

> **Tip**
>
> Elija la flecha hacia abajo junto a Math Expression (expresión matemática) para ver una lista de funciones admitidas que puede utilizar al crear la expresión.

6. En la columna Label (Etiqueta) de la expresión, escriba un nombre que describa lo que calcula la expresión.

   Si el resultado de una expresión es una matriz de series temporales, cada una de esas series temporales se muestra en el gráfico con una línea independiente, con colores distintos. Inmediatamente debajo del gráfico hay una leyenda para cada línea del mismo. Para una expresión única que produce varias series temporales, las leyendas de dichas series temporales tienen el formato ***Etiqueta Expression Etiqueta-Metric (Etiqueta-Expresión Etiqueta-Métrica)***. Por ejemplo, en caso de que el gráfico incluya una métrica con una etiqueta de Errors (Errores) y una expresión FILL(METRICS(), 0) que tenga una etiqueta de Filled With 0: (Rellenado con 0:), una línea de la leyenda sería Filled With 0: Errors (Rellenado con 0: errores). Puede dejar *Etiqueta-Expression (Etiqueta-Expresión)* vacío para que la leyenda solo muestre las etiquetas de métricas originales.

   Cuando una expresión produce una matriz de series temporales en el gráfico, no puede cambiar los colores utilizados para cada una de esas series temporales.

7. Después de haber añadido las expresiones deseadas, puede simplificar el gráfico si oculta algunas de las métricas originales. Para ocultar una métrica o expresión, desactive la casilla de verificación situada a la izquierda del campo Id.

Sintaxis de matemáticas en las métricas y funciones

En las siguientes secciones se explican las funciones disponibles para el cálculo de métricas. Todas las funciones deben escribirse en letras mayúsculas (como AVG) y el campo Id de todas las métricas y expresiones matemáticas debe comenzar con una letra minúscula.

El resultado final de cualquier expresión matemática debe ser una serie temporal individual o una matriz de series temporales. Algunas funciones generan un número escalar. Puede utilizar estas funciones en una función más grande que, en última instancia, produzca una serie temporal. Por ejemplo, aplicar la función AVG a una serie temporal individual produce un número escalar, por lo que no puede ser el resultado de una expresión final. Pero, podría usarlo en la función m1-AVG(m1)

para mostrar una serie temporal de la diferencia entre cada punto de datos individual y el valor promedio de esa serie temporal.

Abreviaturas de tipos de datos

Algunas funciones son válidas únicamente para determinados tipos de datos. Las abreviaturas de la siguiente lista se utilizan en las tablas de funciones para representar los tipos de datos que admite cada función:

- S representa un número escalar, como 2, -5 o 50,25.
- TS es una serie temporal (una serie de valores para una métrica de CloudWatch única conforme avanza el tiempo): por ejemplo, la métrica `CPUUtilization` de la instancia `i-1234567890abcdef0` durante los últimos tres días.
- TS[] es una matriz de series temporales, como las series temporales de varias métricas.
- String [] es una matriz de cadenas.

Función METRICS()

La función METRICS() devuelve todas las métricas en la solicitud. Las expresiones matemáticas no se incluyen.

Puede utilizar METRICS() dentro de una expresión mayor que produce una serie temporal única o una matriz de series temporales. Por ejemplo, la expresión SUM(METRICS()) devuelve una serie temporal (TS) que es la suma de los valores de todas las métricas incluidas en el gráfico. METRICS()/100 devuelve una matriz de series temporales, cada una de las cuales es una serie temporal que muestra cada punto de datos de una de las métricas dividido por 100.

Puede utilizar la función METRICS() con una cadena para devolver solo las métricas incluidas en el gráfico que contienen esa cadena en su campo Id. Por ejemplo, la expresión SUM(METRICS(“errores”)) devuelve una serie temporal que es la suma de los valores de todas las métricas incluidas en el gráfico que tienen “errores” en su campo Id. También puede usar SUM([METRICS(“4xx”), METRICS(“5xx”)]) para establecer la coincidencia con varias cadenas.

Funciones aritméticas básicas

En la siguiente tabla se enumeran las funciones aritméticas básicas que se admiten. Los valores que faltan en una serie temporal se tratan como 0. Si el valor de un punto de datos provoca que una función intente dividir entre cero, se descarta el punto de datos.

Operación	Argumentos	Ejemplos
Operadores aritméticos: + - * / ^	S, S	PERIOD(m1)/60
	S, TS	5 * m1
	TS, TS	m1 - m2
	S, TS[]	SUM(100/[m1, m2])
	TS, TS[]	AVG(METRICS())
		METRICS()*100
Resta unaria -	S	-5*m1
	TS	-m1
	TS[]	SUM(-[m1, m2])

Operadores lógicos y de comparación

Puede utilizar operadores lógicos y de comparación con un par de series temporales o un par de valores escalares únicos. Cuando se utiliza un operador de comparación con un par de series temporales, devuelve una serie temporal en la que cada punto de datos es 0 (FALSE) o 1 (TRUE). Si utiliza un operador de comparación en un par de valores escalares, se devuelve un único valor escalar, 0 o 1.

Cuando se utilizan operadores de comparación entre dos series temporales, pero solo una de ellas tiene un valor para una marca temporal determinada, la función trata el valor que falta en la otra serie temporal como si fuera 0.

Puede utilizar operadores lógicos combinados con operadores de comparación para crear funciones más complejas.

En la siguiente tabla se enumeran los operadores admitidos.

Tipo de operador	Operadores admitidos
Operadores de comparación	==

Tipo de operador	Operadores admitidos
	!= <= >= < >
Logical operators (Operadores lógicos)	AND o && OR o ||

Para ver cómo se utilizan estos operadores, suponga que tiene dos series temporales: metric1 contiene los valores de `[30, 20, 0, 0]` y metric2 contiene los valores de `[20, -, 20, -]`, donde `-` indica que no hay ningún valor para esa marca de tiempo.

Expression	Salida
(metric1 < metric2)	0, 0, 1, 0
(metric1 >= 30)	1, 0, 0, 0
(metric1 > 15 Y metric2 > 15)	1, 0, 0, 0

Funciones admitidas para las métricas matemáticas

En la siguiente tabla se describen las funciones que puede usar en expresiones matemáticas. Escriba todas las funciones en letras mayúsculas.

El resultado final de cualquier expresión matemática debe ser una serie temporal individual o una matriz de series temporales. Algunas funciones en tablas en las siguientes secciones generan un número escalar. Puede utilizar estas funciones en una función más grande que, en última instancia, produzca una serie temporal. Por ejemplo, aplicar la función AVG a una serie temporal individual produce un número escalar, por lo que no puede ser el resultado de una expresión final. Pero, podría

usarlo en la función m1-AVG(m1) para mostrar una serie temporal de la diferencia entre cada punto de datos individual y el valor promedio de ese punto de datos.

En la siguiente tabla, cada ejemplo de la columna Examples (Ejemplos) es una expresión que da como resultado una sola serie temporal o una matriz de series temporales. En estos ejemplos se muestra cómo se pueden utilizar las funciones que devuelven números escalares como parte de una expresión válida que genera una sola serie temporal.

Función	Argumentos	Tipo de retorno *	Descripción	Ejemplos	¿Compatible con la modalidad entre cuentas?
ABS	TS TS[]	TS TS[]	Devuelve el valor absoluto de cada punto de datos.	ABS(m1-m2) MIN(ABS([m1, m2])) ABS(METRICS())	✓
ANOMALY_DETECTION_BAND	TS TS, S	TS[]	Devuelve una banda de detección de anomalías para la métrica especificada. La banda se compone de dos series, una que representa el límite superior del valor previsto “normal” de la métrica y el otro que representa el límite inferior. La función puede tomar dos argumentos. El primero es el ID de la métrica para la que crear la banda. El segundo argumento es	ANOMALY_DETECTION_BAND(m1) ANOMALY_DETECTION_BAND(m1,4)	

Función	Argumentos	Tipo de retorno *	Descripción	Ejemplos	¿Compatible con la modalidad entre cuentas?
			el número de desvíos estándar que desea utilizar para la conexión. Si no se especifica este argumento, se utiliza el valor predeterminado de 2. Para obtener más información, consulte Uso de la detección de anomalías de CloudWatch.		

Función	Argumentos	Tipo de retorno *	Descripción	Ejemplos	¿Compatible con la modalidad entre cuentas?
AVG	TS TS[]	S TS	La función AVG de una serie temporal individual devuelve un valor escalar que representa el promedio de todos los puntos de datos de la métrica. La función AVG de una matriz de series temporales devuelve una sola serie temporal. Los valores que faltan se tratan como 0. **Note** Se recomienda a que no utilice esta función en alarmas de CloudWatch si quiere que la función devuelva un valor escalar. Por ejemplo, `AVG(m2)`. Cada vez que una alarma evalúa si se debe cambiar de estado,	SUM([m1,m2])/AVG(m2) AVG(METRICS())	✓

Función	Argumentos	Tipo de retorno *	Descripción	Ejemplos	¿Compatible con la modalidad entre cuentas?
			CloudWatch intenta recuperar un número de puntos de datos más elevado que el número especificado como Evaluation Periods (Períodos de evaluación). Esta función actúa de manera diferente cuando se solicitan datos adicionales. Para usar esta función con las alarmas, especialmente las que tienen acciones de escalado automático, le recomendamos que configure la alarma para que		

Función	Argumentos	Tipo de retorno *	Descripción	Ejemplos	¿Compatible con la modalidad entre cuentas?
			utilice M de N puntos de datos, donde M < N.		
CEIL	TS TS[]	TS TS[]	Devuelve el valor techo de cada métrica. El valor techo es el menor entero que es mayor o igual que cada valor.	CEIL(m1) CEIL(METRICS()) SUM(CEIL(METRICS()))	✓

Función	Argumentos	Tipo de retorno *	Descripción	Ejemplos	¿Compatible con la modalidad entre cuentas?
DATAPOINT_COUNT	TS TS[]	S TS	Devuelve un recuento de los puntos de datos que han informado valores. Esto es útil para calcular promedios de métricas dispersas. **Note** Se recomienda que no utilice esta función en alarmas de CloudWatch. Cada vez que una alarma evalúa si se debe cambiar de estado, CloudWatch intenta recuperar un número de puntos de datos más elevado que el número especificado como Evaluation Periods (Períodos	SUM(m1) / DATAPOINT_COUNT(m1) DATAPOINT_COUNT(METRICS())	✓

Función	Argumentos	Tipo de retorno *	Descripción	Ejemplos	¿Compatible con la modalidad entre cuentas?
			de evaluación). Esta función actúa de manera diferente cuando se solicitan datos adicionales.		

Función	Argumentos	Tipo de retorno *	Descripción	Ejemplos	¿Compatible con la modalidad entre cuentas?
DB_PERF_INSIGHTS	Cadena Cadena Cadena Cadena cadena, cadena[	TS (si se proporc na una sola cadena) TS [] (si se proporc na una matriz de cadenas	Devuelve métricas del contador Performance Insights Counter para bases de datos como Amazon Relational Database Service y Amazon DocumentDB (con compatibilidad con MongoDB). Esta función devuelve la misma cantidad de datos que puede obtener consultando directamente las API de Performance Insights. Puede usar estas métricas en CloudWatch para graficar y crear alarmas. **Important** Al utilizar esta función, debe especificar el ID de los recursos de base de datos único de la base de datos. Es	DB_PERF_INSIGHTS(‘RDS’, ‘db-ABCDEFGHIJKLMNOPQRSTUVWXY1’, ‘os.cpuUtilization.user.avg’) DB_PERF_INSIGHTS(‘DOCDB, ‘db-ABCDEFGHIJKLMNOPQRSTUVWXY1’, [‘os.cpuUtilization.idle.avg’, ‘os.cpuUtilization.user.max’])	

Función	Argumentos	Tipo de retorno *	Descripción	Ejemplos	¿Compatible con la modalidad entre cuentas?
			diferente del identificador de la base de datos. Para encontrar el ID de recurso de base de datos en la consola de Amazon RDS, elija la instancia de base de datos para ver los detalles. A continuación, elija la pestaña Configuration (Configuración). El ID de recurso se muestra en la sección Configuración. DB_PERF_INSIGHTS también incluye la métrica DBLoad en intervalos de menos de un minuto.		

Función	Argumentos	Tipo de retorno *	Descripción	Ejemplos	¿Compatible con la modalidad entre cuentas?
			Las métricas de Performance Insights recuperadas con esta función no se almacenan en CloudWatch. Por lo tanto, algunas características de CloudWatch, como la observabilidad entre cuentas, la detección de anomalías, los flujos de métricas, el explorador de métricas y Metric Insights, no funcionan con las métricas de Performance Insights que se recuperan con DB_PERF_INSIGHTS. Una sola solicitud que utilice la función DB_PERF_INSIGHTS puede recuperar los siguientes números de puntos de datos. • 1080 puntos de datos para periodos de alta		

Función	Argumentos	Tipo de retorno *	Descripción	Ejemplos	¿Compatible con la modalidad entre cuentas?
			resolución (1 s, 10 s, 30 s) • 1440 puntos de datos para periodos de resolución estándar (1 min, 5 min, 1 h, 1 día) La función DB_PERF_INSIGHTS solo admite las siguientes duraciones de período: • 1 segundo • 10 segundos • 30 segundos • 1 minuto • 5 minutos • 1 hora • 1 día Para obtener más información sobre las métricas de contador de Información de rendimiento de Amazon RDS, consulte Métricas de contador		

Función	Argumentos	Tipo de retorno *	Descripción	Ejemplos	¿Compatible con la modalidad entre cuentas?
			de Información de rendimiento. Para obtener más información sobre las métricas de contador de Información de rendimiento de Amazon DocumentDB, consulte Métricas de contador de Información de rendimiento. **Note** Las métricas de alta resolución con una granularidad inferior a un minuto recuperadas por DB_PERF_INSIGHTS solo se aplican a la métrica DBLoad, o a las métricas del sistema operativo si ha activado		

Función	Argumentos	Tipo de retorno *	Descripción	Ejemplos	¿Compatible con la modalidad entre cuentas?
			la supervisión mejorada con una resolución más alta. Para obtener más información sobre la supervisión mejorada de Amazon RDS, consulte Supervisión de las métricas del SO con la supervisión mejorada. Puede crear una alarma de alta resolución mediante la función DB_PERF_INSIGHTS durante un intervalo de tiempo máximo de tres horas. Puede utilizar la consola de CloudWatch		

Función	Argumentos	Tipo de retorno *	Descripción	Ejemplos	¿Compatible con la modalidad entre cuentas?
			para representar gráficamente las métricas recuperadas con la función DB_PERF_INSIGHTS para cualquier intervalo de tiempo.		
DIFF	TS TS[]	TS TS[]	Devuelve la diferencia entre cada valor en la serie temporal y el valor anterior de esa serie temporal.	DIFF(m1)	✓
DIFF_TIME	TS TS[]	TS TS[]	Devuelve la diferencia en segundos entre la marca temporal de cada valor de la serie temporal y la marca temporal del valor anterior de esa serie temporal.	DIFF_TIME(METRICS())	✓

Función	Argumentos	Tipo de retorno *	Descripción	Ejemplos	¿Compatible con la modalidad entre cuentas?
FILL	TS, [S | REPEA | LINEAR TS[], [TS | S | REPEA | LINEAR	TS TS[]	Rellena los valores que faltan de una serie temporal. Hay varias opciones para que los valores se utilicen como relleno de los valores que faltan: • Puede especificar un valor para usar como valor de relleno. • Puede especificar una métrica para utilizarla como valor de relleno. • Puede utilizar REPEAT (Repetir) para rellenar los valores que faltan con el valor real más reciente de la métrica antes del valor que falta. • Puede utilizar LINEAL (Lineal) para rellenar los valores faltantes con valores que crean una interpolación lineal entre los valores	FILL(m1,10) FILL(METRICS(), 0) FILL(METRICS(), m1) FILL(m1, MIN(m1)) FILL(m1, REPEAT) FILL(METRICS(), LINEAR)	✓

Función	Argumentos	Tipo de retorno *	Descripción	Ejemplos	¿Compatible con la modalidad entre cuentas?
			al principio y al final de la brecha. **Note** Cuando utiliza esta función en una alarma, se puede encontrar con un problema si sus métricas se publican con un ligero retraso y el minuto más reciente nunca tiene datos. En este caso, FILL (Rellenar) reemplaza ese punto de datos faltante con el valor solicitado. Esto hace que el último punto de datos de la métrica sea siempre el valor de relleno, lo que		

Función	Argumentos	Tipo de retorno *	Descripción	Ejemplos	¿Compatible con la modalidad entre cuentas?
			puede provocar que la alarma se atasque en el estado OK o ALARM (ALARMA). Puede evitarlo si utiliza una alarma M de N. Para obtener más información, consulte Evaluación de una alarma.		

Función	Argumentos	Tipo de retorno *	Descripción	Ejemplos	¿Compatible con la modalidad entre cuentas?
FIRST LAST	TS[]	TS	Devuelve la primera o la última serie temporal de una matriz de series temporales. Esto resulta útil cuando se utiliza con la función SORT. También se puede utilizar para obtener los umbrales superiores e inferiores de la función ANOMALY_DETECTION_BAND.	IF(FIRST(SORT(METRICS(), AVG, DESC))>100, 1, 0) Toma la métrica superior de una matriz, ordenada según AVG. A continuación, devuelve 1 o 0 para cada punto de datos, según si el valor de ese punto de datos es mayor que 100. LAST(ANOMALY_DETECTION_BAND(m1)) devuelve el límite superior de la banda de predicción de anomalías.	✓
FLOOR	TS TS[]	TS TS[]	Devuelve el valor de suelo de cada métrica. El valor de suelo es el mayor entero que es menor o igual que cada valor.	FLOOR(m1) FLOOR(METRICS())	✓

Función	Argumentos	Tipo de retorno *	Descripción	Ejemplos	¿Compatible con la modalidad entre cuentas?
IF	Expresi IF	TS	Utilice IF junto con un operador de comparación para filtrar los puntos de datos de una serie temporal o crear una serie temporal mixta compuesta por varias series de temporales intercaladas. Para obtener más información, consulte Uso de expresiones IF.	Para ver ejemplos, consulte Uso de expresiones IF.	✓
INSIGHT_RULE_METRIC	INSIGHULE_MIC(ruleNe, metricNe)	TS	Utilice INSIGHT_RULE_METRIC para extraer estadísticas de una regla en Contributor Insights. Para obtener más información, consulte Representación gráfica de métricas generadas por reglas.		

Función	Argumentos	Tipo de retorno*	Descripción	Ejemplos	¿Compatible con la modalidad entre cuentas?
LAMBDA	LAMBDA(FunctionName [, optionalarg]*)	TS TS{}	Llama a una función de Lambda para consultar las métricas de un origen de datos que no es CloudWatch. Para obtener más información, consulte Cómo pasar argumentos a la función de Lambda.		
LOG	TS TS[]	TS TS[]	El LOG (Registro) de una serie temporal devuelve el valor del logaritmo natural de cada valor de la serie temporal.	LOG(METRICS())	✓
LOG10	TS TS[]	TS TS[]	El LOG10 de una serie temporal devuelve el valor del logaritmo base-10 de cada valor de la serie temporal.	LOG10(m1)	✓

Función	Argumentos	Tipo de retorno *	Descripción	Ejemplos	¿Compatible con la modalidad entre cuentas?
MAX	TS TS[]	S TS	La función MAX de una serie temporal individual devuelve un valor escalar que representa el valor máximo de todos los puntos de datos de la métrica. Si introduce una matriz de series temporales, la función MAX crea y devuelve una serie temporal que consta del valor más alto de cada punto de datos, entre las series temporales que se utilizaron como entrada. **Note** Se recomienda que no utilice esta función en alarmas de CloudWatch si quiere que la función devuelva un valor escalar.	MAX(m1)/m1 MAX(METRICS())	✓

Función	Argumentos	Tipo de retorno *	Descripción	Ejemplos	¿Compatible con la modalidad entre cuentas?
			Por ejemplo, `MAX(m2)` cada vez que una alarma evalúa si se debe cambiar de estado, CloudWatch intenta recuperar un número de puntos de datos más elevado que el número especificado como Períodos de evaluación. Esta función actúa de manera diferente cuando se solicitan datos adicionales.		
METRIC_COUNT	TS[]	S	Devuelve el número de métricas en la matriz de series temporales.	m1/METRIC_COUNT(METRICS())	✓

Función	Argumentos	Tipo de retorno *	Descripción	Ejemplos	¿Compatible con la modalidad entre cuentas?
METRICS	null cadena	TS[]	La función METRICS() devuelve todas las métricas de CloudWatch de la solicitud. Las expresiones matemáticas no se incluyen. Puede utilizar METRICS() dentro de una expresión mayor que produce una serie temporal única o una matriz de series temporales. Puede utilizar la función METRICS() con una cadena para devolver solo las métricas incluidas en el gráfico que contienen esa cadena en su campo Id. Por ejemplo, la expresión SUM(METRICS(“errores”)) devuelve una serie temporal que es la suma de los valores de todas las métricas incluidas en el gráfico que tienen	AVG(METRICS()) SUM(METRICS(“errors”))	✓

Función	Argumentos	Tipo de retorno *	Descripción	Ejemplos	¿Compatible con la modalidad entre cuentas?
			“errores” en su campo Id. También puede usar SUM([METRICS(“4xx”), METRICS(“5xx”)]) para establecer la coincidencia con varias cadenas.		

Función	Argumentos	Tipo de retorno *	Descripción	Ejemplos	¿Compatible con la modalidad entre cuentas?
MIN	TS TS[]	S TS	La función MIN de una serie temporal individual devuelve un valor escalar que representa el valor mínimo de todos los puntos de datos de la métrica. Si introduce una matriz de series temporales, la función MIN crea y devuelve una serie temporal que consta del valor más bajo de cada punto de datos de las series temporales que se utilizaron como entrada. Si introduce una matriz de series temporales, la función MIN crea y devuelve una serie temporal que consta del valor más alto de cada punto de datos de las series temporales que se utilizaron como entrada.	m1-MIN(m1) MIN(METRICS())	✓

Función	Argumentos	Tipo de retorno *	Descripción	Ejemplos	¿Compatible con la modalidad entre cuentas?
			Note Se recomienda que no utilice esta función en alarmas de CloudWatch si quiere que la función devuelva un valor escalar. Por ejemplo, `MIN(m2)` cada vez que una alarma evalúa si se debe cambiar de estado, CloudWatch intenta recuperar un número de puntos de datos más elevado que el número especificado como Períodos de evaluación. Esta función actúa de manera		

Función	Argumentos	Tipo de retorno *	Descripción	Ejemplos	¿Compatible con la modalidad entre cuentas?
			diferente cuando se solicitan datos adicionales.		

Función	Argumentos	Tipo de retorno *	Descripción	Ejemplos	¿Compatible con la modalidad entre cuentas?

MINUTE
HOUR
DAY
DATE
MONTH
YEAR
EPOCH
	TS	TS	Estas funciones toman el período y el rango de la serie temporal y devuelven una nueva serie temporal no dispersa donde cada valor se basa en su marca de tiempo.
			• MINUTE (Minuto) devuelve una serie temporal no dispersa de enteros entre 0 y 59 que representan el minuto UTC de cada marca de tiempo de la serie temporal original.
			• HOUR (Hora) devuelve una serie temporal no dispersa de enteros entre 0 y 23 que representan la hora UTC de cada marca de tiempo de la serie temporal original.
			• DAY (Día) devuelve una serie temporal no
				MINUTE(m1)
				IF(DAY(m1)<6,m1) devuelve las métricas solo de lunes a viernes.
				IF(MONTH(m1) == 4,m1) devuelve sólo las métricas publicadas en abril.
					✓

Función	Argumentos	Tipo de retorno *	Descripción	Ejemplos	¿Compatible con la modalidad entre cuentas?
			dispersa de enteros entre 1 y 7 que representan el día UTC de la semana de cada marca de tiempo en la serie temporal original. 1 representa lunes y 7 representa domingo. • DATE (Fecha) devuelve una serie temporal no dispersa de enteros entre 1 y 31 que representan el día UTC del mes de cada marca de tiempo de la serie temporal original. • MONTH (Mes) devuelve una serie temporal no dispersa de enteros entre 1 y 12 que representan el mes UTC de cada marca de tiempo de la serie temporal original. 1 representa enero y 12 representa diciembre.		

Función	Argumentos	Tipo de retorno *	Descripción	Ejemplos	¿Compatible con la modalidad entre cuentas?
			• YEAR (Año) devuelve una serie temporal no dispersa de enteros que representan el año UTC de cada marca de tiempo de la serie temporal original. • EPOCH (Fecha de inicio) devuelve una serie temporal no dispersa de enteros que representan el tiempo UTC en segundos desde la fecha de inicio de cada marca de tiempo en la serie temporal original. La fecha de inicio es el 1 de enero de 1970.		
PERIOD	TS	S	Devuelve el periodo de la métrica en segundos. La entrada válida son las métricas, no los resultados de otras expresiones.	m1/PERIOD(m1)	✓

Función	Argumentos	Tipo de retorno *	Descripción	Ejemplos	¿Compatible con la modalidad entre cuentas?
RATE	TS TS[]	TS TS[]	Devuelve la velocidad de cambio de la métrica, por segundo. Se calcula como la diferencia entre el valor de punto de datos más reciente y el valor de punto de datos anterior, dividido por la diferencia de tiempo en segundos entre los dos valores. **Important** La configuración de alarmas en expresiones que utilizan la función RATE en métricas con datos escasos puede comportarse de forma impredecible, ya que el rango de puntos de datos obtenidos al evaluar la alarma puede	RATE(m1) RATE(METRICS())	✓

Función	Argumentos	Tipo de retorno *	Descripción	Ejemplos	¿Compatible con la modalidad entre cuentas?
			variar en función de cuándo se publicaron los puntos de datos por última vez.		

Función	Argumentos	Tipo de retorno *	Descripción	Ejemplos	¿Compatible con la modalidad entre cuentas?
REMOVE_EMPTY	TS[]	TS[]	Elimina todas las series temporales que no contienen puntos de datos de una matriz de series temporales. El resultado es una matriz de series temporales en la que todas las series temporales contienen al menos un punto de datos. **Note** Se recomienda que no utilice esta función en alarmas de CloudWatch. Cada vez que una alarma evalúa si se debe cambiar de estado, CloudWatch intenta recuperar un número de puntos de datos más elevado	REMOVE_EMPTY(METRICS())	✓

Función	Argumentos	Tipo de retorno *	Descripción	Ejemplos	¿Compatible con la modalidad entre cuentas?
			que el número especificado como Evaluation Periods (Períodos de evaluación). Esta función actúa de manera diferente cuando se solicitan datos adicionales.		

Función	Argumentos	Tipo de retorno *	Descripción	Ejemplos	¿Compatible con la modalidad entre cuentas?
RUNNING_SUM	TS TS[]	TS TS[]	Devuelve una serie temporal con la suma continua de los valores de la serie temporal original. **Note** Se recomienda que no utilice esta función en alarmas de CloudWatch. Cada vez que una alarma evalúa si se debe cambiar de estado, CloudWatch intenta recuperar un número de puntos de datos más elevado que el número especificado como Evaluation Periods (Períodos de evaluació	RUNNING_SUM([m1,m2])	✓

Función	Argumentos	Tipo de retorno *	Descripción	Ejemplos	¿Compatible con la modalidad entre cuentas?
			n). Esta función actúa de manera diferente cuando se solicitan datos adicionales.		

Función	Argumentos	Tipo de retorno *	Descripción	Ejemplos	¿Compatible con la modalidad entre cuentas?
SEARCH	Expresión de búsqueda	Una o varias series temporales	Devuelve una o varias series temporales que coinciden con una serie temporal que especifique. La función SEARCH le permite añadir varias series temporales relacionadas a un gráfico con una sola expresión. El gráfico se actualiza dinámicamente para incluir nuevas métricas que se añadan más tarde y coincidan con los criterios de búsqueda. Para obtener más información, consulte Usar expresiones de búsqueda en gráficos. No es posible crear una alarma con una expresión SEARCH (de búsqueda). Esto se debe a que las expresiones de búsqueda devuelven varias series temporales, y una alarma basada		✓

Función	Argumentos	Tipo de retorno *	Descripción	Ejemplos	¿Compatible con la modalidad entre cuentas?
			en una expresión matemática sólo puede ver una serie temporal. Si inició sesión en una cuenta de supervisión para la observabilidad entre cuentas de CloudWatch, la función SEARCH (BUSCAR) busca métricas en las cuentas de origen y en la cuenta de supervisión.		
SERVICE_QUOTA	TS que es una métrica de uso	TS	Devuelve la cuota de servicio de la métrica de uso especificada. Puede usarla para visualizar cómo se compara su uso actual con la cuota, así como para programar alarmas que le avisen al acercarse a la cuota. Para obtener más información, consulte Métricas de uso de AWS.		✓

Función	Argumentos	Tipo de retorno *	Descripción	Ejemplos	¿Compatible con la modalidad entre cuentas?
SLICE	(TS[], S, S) o (TS[], S)	TS[] TS	Recupera parte de una matriz de series temporales. Resulta especialmente útil cuando se combina con SORT. Por ejemplo, puede excluir el resultado superior de una matriz de series temporales. Puede utilizar dos argumentos escalares para definir el conjunto de series temporales que desea devolver. Los dos escalares definen el inicio (incluido) y el final (excluido) de la matriz que se va a devolver. La matriz está indexada a cero, por lo que la primera serie temporal de la matriz es la serie temporal 0. Si el usuario lo prefiere, puede especificar un solo valor y CloudWatch devolverá todas las series temporales que	SLICE(SORT(METRICS(), SUM, DESC), 0, 10) devuelve las 10 métricas de la matriz de métricas de la solicitud que tengan el valor de SUM más alto. SLICE(SORT(METRICS(), AVG, ASC), 5) ordena la matriz de métricas según la estadística AVG; a continuación, devuelve todas las series temporales excepto las 5 con el valor de AVG más bajo.	✓

Función	Argumentos	Tipo de retorno *	Descripción	Ejemplos	¿Compatible con la modalidad entre cuentas?
			comiencen por ese valor.		

Función	Argumentos	Tipo de retorno *	Descripción	Ejemplos	¿Compatible con la modalidad entre cuentas?
SORT	(TS[], FUNCT SORT_(R) (TS[], FUNCT SORT_(R, S)	TS[]	Ordena una matriz de series temporales según la función que especifiq ue. La función utilizada puede ser AVG, MIN, MAX o SUM. El orden puede ser ASC si es ascendente (los valores más bajos primero) o DESC para ordenar los valores más altos al principio. Opcionalm ente, puede especific ar un número después del criterio de ordenació n para que actúe como límite. Por ejemplo, si especifica un límite de 5, se devolverán solo las 5 primeras series temporales de la ordenación. Cuando esta función matemática se muestra en un gráfico, las etiquetas de cada métrica del gráfico	SORT(METRICS(), AVG, DESC, 10) calcula el valor medio de cada serie temporal, ordena las series temporale s con los valores más altos al principio de la ordenación y devuelve solo las 10 series temporales con los promedios más altos. SORT(METRICS(), MAX, ASC) ordena la matriz de métricas según la estadística MAX; a continuación, las devuelve todas en orden ascendente.	✓

Función	Argumentos	Tipo de retorno *	Descripción	Ejemplos	¿Compatible con la modalidad entre cuentas?
			también se ordenan y numeran.		

Función	Argumentos	Tipo de retorno *	Descripción	Ejemplos	¿Compatible con la modalidad entre cuentas?
STDDEV	TS TS[]	S TS	La función STDDEV de una serie temporal individual devuelve un valor escalar que representa el desvío estándar de todos los puntos de datos de la métrica. La función STDDEV de una matriz de series temporales devuelve una sola serie temporal. **Note** Se recomienda a que no utilice esta función en alarmas de CloudWatch si quiere que la función devuelva un valor escalar. Por ejemplo, `STDDEV(m2)` cada vez que una alarma evalúa si se debe cambiar	m1/STDDEV(m1) STDDEV(METRICS())	✓

Función	Argumentos	Tipo de retorno *	Descripción	Ejemplos	¿Compatible con la modalidad entre cuentas?
			de estado, CloudWatch intenta recuperar un número de puntos de datos más elevado que el número especificado como Períodos de evaluación. Esta función actúa de manera diferente cuando se solicitan datos adicionales.		

Función	Argumentos	Tipo de retorno *	Descripción	Ejemplos	¿Compatible con la modalidad entre cuentas?
SUM	TS TS[]	S TS	La función SUM de una serie temporal individual devuelve un valor escalar que representa la suma de los valores de todos los puntos de datos de la métrica. La función SUM de una matriz de series temporales devuelve una sola serie temporal. **Note** Se recomienda a que no utilice esta función en alarmas de CloudWatch si quiere que la función devuelva un valor escalar. Por ejemplo, `SUM(m1)`. Cada vez que una alarma evalúa si se debe cambiar de estado, CloudWatc	SUM(METRICS())/SUM(m1) SUM([m1,m2]) SUM(METRICS(“errores”))/SUM(METRICS(“solicitudes”))*100	✓

Función	Argumentos	Tipo de retorno *	Descripción	Ejemplos	¿Compatible con la modalidad entre cuentas?
			h intenta recuperar un número de puntos de datos más elevado que el número especificado como Evaluation Periods (Períodos de evaluación). Esta función actúa de manera diferente cuando se solicitan datos adicionales.		
TIME_SERIES	S	TS	Devuelve una serie temporal no dispersa en donde cada valor se establece en un argumento escalar.	TIME_SERIES(MAX(m1)) TIME_SERIES(5*AVG(m1)) TIME_SERIES(10)	✓

*No es válido usar una función que solo devuelva un número escalar, ya que el resultado final de las expresiones debe ser una sola serie temporal o una matriz de series temporales. En su lugar, utilice estas funciones como parte de una expresión más grande que devuelva una serie temporal.

Uso de expresiones IF

Utilice IF junto con un operador de comparación para filtrar los puntos de datos de una serie temporal o crear una serie temporal mixta compuesta por varias series de temporales intercaladas.

IF adopta los siguientes argumentos:

```
IF(condition, trueValue, falseValue)
```

La condición se evalúa como FALSE si el valor del punto de datos de la condición es 0 y como TRUE si el valor de la condición es cualquier otro valor, ya sea positivo o negativo. Si la condición es una serie temporal, se evalúa por separado para cada marca temporal.

A continuación se enumeran las sintaxis válidas. Para cada una de estas sintaxis, la salida es una sola serie temporal.

- IF(TS ***operador de comparación*** S, S | TS, S | TS)

  > **Note**
  >
  > Si el `TS comparison operator S` es TRUE, pero `metric2` no tiene un punto de datos correspondiente, la salida será 0.

- IF(TS, TS, TS)
- IF(TS, S, TS)
- IF(TS, TS, S)
- IF(TS, S, S)
- IF(S, TS, TS)

En las siguientes secciones se proporcionan más detalles y ejemplos de estas sintaxis.

IF(TS ***operación de comparación*** S, scalar2 | metric2, scalar3 | metric3)

El valor correspondiente de la serie temporal de salida:

- tiene el valor de scalar2 o metric2 si el TS ***Comparison Operator*** (Operador de comparación TS) S es VERDADERO.
- tiene el valor de scalar3 o metric3, si TS ***Comparison Operator*** (Operador de comparación TS), S es FALSO.
- tiene el valor de 0 si el ***operador de comparación*** TS es TRUE y el punto de datos correspondiente en metric2 no existe.
- tiene el valor de 0 si el ***operador de comparación*** TS es FALSE y el punto de datos correspondiente en metric3 no existe.
- es una serie temporal vacía si el punto de datos correspondiente no existe en metric3 o si scalar3/metric3 se omite de la expresión.

IF(metric1, metric2, metric3)

Para cada punto de datos de metric1, el valor de la serie temporal de salida correspondiente:

- tiene el valor de metric2 si el punto de datos correspondiente de metric1 es TRUE;
- tiene el valor de metric3 si el punto de datos correspondiente de metric1 es FALSE;
- tiene el valor de 0 si el punto de datos correspondiente de metric1 es TRUE y el punto de datos correspondiente no existe en metric2.
- se elimina si el punto de datos correspondiente de metric1 es FALSE y el punto de datos correspondiente no existe en metric3
- se elimina si el punto de datos correspondiente de metric1 es FALSE y metric3 se omite de la expresión.
- se elimina si no existe el punto de datos correspondiente de metric1.

En la tabla siguiente se muestra un ejemplo de esta sintaxis.

Métrica o función	Valores
(metric1)	[1, 1, 0, 0, -]
(metric2)	[30, -, 0, 0, 30]
(metric3)	[0, 0, 20, -, 20]

Métrica o función	Valores
IF(metric1, metric2, metric3)	[30, 0, 20, 0, -]

IF(metric1, scalar2, metric3)

Para cada punto de datos de metric1, el valor de la serie temporal de salida correspondiente:

- tiene el valor de scalar2 si el punto de datos correspondiente de metric1 es TRUE;
- tiene el valor de metric3 si el punto de datos correspondiente de metric1 es FALSE;
- se elimina si el punto de datos correspondiente de metric1 es FALSE y el punto de datos correspondiente no existe en metric3 o si metric3 se omite en la expresión.

Métrica o función	Valores
(metric1)	[1, 1, 0, 0, -]
scalar2	5
(metric3)	[0, 0, 20, -, 20]
IF(metric1, scalar2, metric3)	[5, 5, 20, -, -]

IF(metric1, metric2, scalar3)

Para cada punto de datos de metric1, el valor de la serie temporal de salida correspondiente:

- tiene el valor de metric2 si el punto de datos correspondiente de metric1 es TRUE;
- tiene el valor de scalar3 si el punto de datos correspondiente de metric1 es FALSE;
- tiene el valor de 0 si el punto de datos correspondiente de metric1 es TRUE y el punto de datos correspondiente no existe en metric2.
- se elimina si no existe el punto de datos correspondiente en metric1.

Métrica o función	Valores
(metric1)	[1, 1, 0, 0, -]
(metric2)	[30, -, 0, 0, 30]
scalar3	5
IF(metric1, metric2, scalar3)	[30, 0, 5, 5, -]

IF(scalar1, metric2, metric3)

El valor correspondiente de la serie temporal de salida:

- tiene el valor de metric2 si scalar1 es TRUE;
- tiene el valor de metric3 si scalar1 es FALSE;
- es una serie temporal vacía si se omite metric3 de la expresión.

Ejemplos de casos de uso para expresiones condicionales

Los siguientes ejemplos ilustran los posibles usos de la función IF .

- Para mostrar solo los valores bajos de una métrica:

  IF(metric1<400, metric1)
- Para cambiar cada punto de datos de una métrica a uno de dos valores, con el fin de mostrar los máximos y mínimos relativos de la métrica original:

  SI (metric1 < 400, 10, 2)
- Para mostrar un 1 para cada marca temporal cuya latencia supere el umbral y mostrar un 0 para todos los demás puntos de datos:

  IF(latency>threshold, 1, 0)

Usar matemáticas de métricas con la operación de la API GetMetricData

Puede utilizar `GetMetricData` para realizar cálculos mediante expresiones matemáticas, así como para recuperar lotes grandes de datos de métricas en una sola llamada a la API. Para obtener más información, consulte GetMetricData.

Detección de anomalías en matemáticas de métricas

La detección de anomalías en matemáticas de métricas es una característica que se puede utilizar para crear alarmas de detección de anomalías en métricas individuales y en los resultados de expresiones matemáticas métricas. Puede utilizar estas expresiones para crear gráficos en los que se visualicen bandas de detección de anomalías. La característica admite funciones aritméticas básicas, operadores lógicos y de comparación y la mayoría de las demás funciones.

La detección de anomalías en matemáticas de métricas no admite las siguientes funciones:

- Expresiones que contienen más de una ANOMALY_DETECTION_BAND en el mismo renglón.
- Expresiones que contienen más de 10 métricas o expresiones matemáticas.
- Expresiones que contienen la expresión METRICS.
- Expresiones que contienen la función SEARCH.
- Expresiones que utilizan la función DP_PERF_INSIGHTS.
- Expresiones que utilizan métricas con diferentes periodos.
- Detectores de anomalías de matemáticas de métricas que utilizan métricas de alta resolución como entrada.

Para obtener más información acerca de esta característica, consulte Uso de la detección de anomalías de CloudWatch en la Guía del usuario de Amazon CloudWatch.

Usar expresiones de búsqueda en gráficos

Las expresiones de búsqueda son un tipo de expresión matemática que puede agregar a los gráficos de CloudWatch. Las expresiones de búsqueda le permiten añadir de manera rápida varias métricas relacionadas a un gráfico. También le permiten crear gráficos dinámicos que añadan automáticamente las métricas correspondientes al modelo de representación, aunque dichas métricas no existan en el momento de crear el gráfico.

Por ejemplo, puede crear una expresión de búsqueda que muestre la métrica `AWS/EC2` `CPUUtilization` de todas las instancias de la región. Si posteriormente lanza una nueva instancia, la métrica `CPUUtilization` de la nueva instancia se añade automáticamente al gráfico.

Cuando se utiliza una expresión de búsqueda en un gráfico, se busca la expresión de búsqueda en los nombres de las métricas, espacios de nombres, nombres de dimensión y valores de dimensión. Puede utilizar operadores booleanos para búsquedas más complejas y eficaces. Una expresión de búsqueda solo puede encontrar métricas que hayan registrado datos en las últimas dos semanas.

No es posible crear una alarma mediante la expresión SEARCH (de búsqueda). Esto se debe a que las expresiones de búsqueda devuelven varias series temporales, y una alarma basada en una expresión matemática sólo puede ver una serie temporal.

Si usa una cuenta de supervisión para la observabilidad entre cuentas de CloudWatch, las expresiones de búsqueda pueden buscar métricas en las cuentas de origen vinculadas a la cuenta de supervisión.

Temas

Sintaxis de la expresión de búsqueda de CloudWatch

Una expresión de búsqueda válida tiene el siguiente formato.

```
SEARCH(' {Namespace, DimensionName1, DimensionName2, ...} SearchTerm', 'Statistic')
```

Por ejemplo:

```
SEARCH('{AWS/EC2,InstanceId} MetricName="CPUUtilization"', 'Average')
```

- La primera parte de la consulta detrás de la palabra `SEARCH`, incluida entre llaves, es el esquema de la métrica que se va a buscar. El esquema de la métrica contiene un espacio de nombres de métricas y uno o varios nombres de dimensión. La inclusión de un esquema de métrica en una consulta de búsqueda es opcional. Si se especifica, el esquema de la métrica debe contener un espacio de nombres y puede contener uno o varios nombres de dimensión que sean válidos en ese espacio de nombres.

No es necesario utilizar comillas dentro del esquema de la métrica, a menos que un espacio de nombres o un nombre de dimensión incluya espacios o caracteres no alfanuméricos. En tal caso, debe entrecomillar el nombre que contiene los caracteres con comillas dobles.

- `SearchTerm` también es opcional, pero una búsqueda válida debe contener el esquema de la métrica, `SearchTerm` o ambos. `SearchTerm` contiene normalmente uno o más ID de cuenta, nombres de métrica o valores de dimensión. `SearchTerm` puede incluir varios términos de búsqueda, tanto para coincidencias parciales como exactas. También puede contener operadores booleanos.

  El uso de un ID de cuenta en `SearchTerm` solo funciona en las cuentas que estén configuradas como cuentas de supervisión para la observabilidad entre cuentas de CloudWatch. La sintaxis de un ID de cuenta en `SearchTerm` es `:aws.AccountId = "444455556666"`. También puede usar `'LOCAL'` para especificar la propia cuenta de supervisión: `:aws.AccountId = 'LOCAL'`

  Para obtener más información, consulte Observabilidad entre cuentas de CloudWatch.

  `SearchTerm` puede incluir uno o varios indicadores, como `MetricName=` como en este ejemplo, pero el uso de indicadores no es obligatorio.

  El esquema de la métrica y `SearchTerm` deben incluirse entre un par de comillas simples.
- `Statistic` es el nombre de cualquier estadística de CloudWatch válida. Debe ir entre comillas simples. Para obtener más información, consulte Statistics.

El ejemplo anterior busca el espacio de nombres de `AWS/EC2` para cualquier métrica que tenga `InstanceId` como nombre de dimensión. Devuelve todas las métricas `CPUUtilization` que encuentra y el gráfico muestra la estadística `Average`.

Una expresión de búsqueda solo puede encontrar métricas que hayan registrado datos en las últimas dos semanas.

Límites de la expresión de búsqueda

El tamaño máximo de una consulta de expresión de búsqueda es de 1024 caracteres. Puede tener 100 expresiones de búsqueda como máximo en un gráfico. Un gráfico puede mostrar hasta 500 series temporales.

Expresiones de búsqueda de CloudWatch: tokenización

Cuando especifica un `SearchTerm`, la función de búsqueda busca tokens, que son subcadenas que CloudWatch genera automáticamente a partir de los nombres de métricas, los nombres de dimensiones, los valores de las dimensiones y los espacios de nombres completos. CloudWatch genera tokens que se distinguen por la notación camello en la cadena original. Los caracteres numéricos sirven también como inicio de nuevos tokens y los caracteres no alfanuméricos sirven como delimitadores, de manera que los tokens se crean antes y después de los caracteres alfanuméricos.

Una cadena continua del mismo tipo de carácter delimitador de token produce un solo token.

Todos los tokens generados están en minúsculas. En la siguiente tabla se muestran algunos ejemplos de tokens generados.

Cadena original	Tokens generados
CustomCount1	`customcount1` , `custom`, `count`, `1`
SDBFailure	`sdbfailure` , `sdb`, `failure`
Project2-trial333	`project2trial333` , `project`, `2`, `trial`, `333`

Expresiones de búsqueda de CloudWatch: concordancias parciales

Cuando se especifica un `SearchTerm` , el término de búsqueda también está tokenizado. CloudWatch encuentra métricas en función de las concordancias parciales, que son concordancias de un token único generado de un término de búsqueda con un token único generado a partir de un nombre de métrica, un espacio de nombres, un nombre de dimensión o de un valor de dimensión.

Las búsquedas de coincidencias parciales que buscan un solo token no distinguen entre mayúsculas y minúsculas. Por ejemplo, cualquiera de los siguientes términos de búsqueda puede devolver la métrica `CustomCount1`:

- `count`
- `Count`
- `COUNT`

Sin embargo, si se utiliza `couNT` como un término de búsqueda, no se encuentra `CustomCount1`, ya que el uso de mayúsculas y minúsculas en el término de búsqueda `couNT` se ha tokenizado en `cou` y `NT`.

Las búsquedas también buscan tokens compuestos, que son varios tokens que aparecen de manera consecutiva en el nombre original. Para buscar un token compuesto coincidente, la búsqueda distingue entre mayúsculas y minúsculas. Por ejemplo, si el término original es `CustomCount1`, las búsquedas de `CustomCount` o `Count1` devuelven resultados, pero no así las búsquedas de `customcount` o `count1`.

Expresiones de búsqueda de CloudWatch: concordancias exactas

Puede definir una búsqueda para que busque solo coincidencias exactas del término de búsqueda utilizando dobles comillas en la parte del término de búsqueda que requiere una coincidencia exacta. Estas comillas dobles se incluyen entre las comillas simples utilizadas alrededor de todo el término de búsqueda. Por ejemplo, **`SEARCH(' {MyNamespace}, "CustomCount1" ', 'Maximum')`** encuentra la cadena exacta `CustomCount1` si existe como un nombre de métrica, nombre de dimensión o valor de dimensión en el espacio de nombres denominado `MyNamespace`. Sin embargo, las búsquedas **`SEARCH(' {MyNamespace}, "customcount1" ', 'Maximum')`** o **`SEARCH(' {MyNamespace}, "Custom" ', 'Maximum')`** no encuentran esta cadena.

Puede combinar términos de coincidencia parcial y términos de coincidencia exacta en la misma expresión de búsqueda. Por ejemplo, **`SEARCH(' {AWS/NetworkELB, LoadBalancer} "ConsumedLCUs" OR flow ', 'Maximum')`** devuelve la métrica Elastic Load Balancing llamada `ConsumedLCUs` además de todas las métricas de Elastic Load Balancing o las dimensiones que contienen el token`flow`.

El uso de coincidencias exactas también es una buena forma de buscar nombres con caracteres especiales, como caracteres no alfanuméricos o espacios, tal y como se muestra en el siguiente ejemplo.

```
SEARCH(' {"My Namespace", "Dimension@Name"}, "Custom:Name[Special_Characters" ',
 'Maximum')
```

Expresiones de búsqueda de CloudWatch: exclusión de un esquema de métrica

Todos los ejemplos mostrados hasta ahora incluyen un esquema de métrica entre llaves. Las búsquedas que omiten un esquema de métrica también son válidas.

Por ejemplo, **`SEARCH(' "CPUUtilization" ', 'Average')`** devuelve todos los nombres de métrica, nombres de dimensión, valores de dimensión y espacios de nombres que coinciden exactamente con la cadena `CPUUtilization`. En los espacios de nombres de métricas de AWS, se pueden incluir métricas de varios servicios incluidos Amazon EC2, Amazon ECS, SageMaker, entre otros.

Para acotar esta búsqueda a un solo servicio de AWS, la práctica recomendada es especificar el espacio de nombres y todas las dimensiones necesarias en el esquema de la métrica, tal y como se muestra en el siguiente ejemplo. Aunque esto acota la búsqueda al espacio de nombres `AWS/EC2`, devolvería igualmente resultados de otras métricas si ha definido `CPUUtilization` como un valor de dimensión para dichas métricas.

```
SEARCH(' {AWS/EC2, InstanceType} "CPUUtilization" ', 'Average')
```

También podría añadir el espacio de nombres al `SearchTerm` tal y como se muestra en el siguiente ejemplo. Sin embargo, en este ejemplo, la búsqueda coincidiría con cualquier cadena `AWS/EC2`, incluso si se tratase de un nombre o valor de dimensión personalizado.

```
SEARCH(' "AWS/EC2" MetricName="CPUUtilization" ', 'Average')
```

Expresiones de búsqueda de CloudWatch: especificación de los nombres de las propiedades en la búsqueda

La siguiente búsqueda de coincidencia exacta de `"CustomCount1"` devuelve todas las métricas que tienen ese nombre exacto.

```
SEARCH(' "CustomCount1" ', 'Maximum')
```

Pero también devuelve métricas con nombres de dimensión, valores de dimensión o espacios de nombres `CustomCount1`. Para estructurar aún más la búsqueda, puede especificar el nombre de propiedad del tipo de objeto que desea buscar en sus búsquedas. En el siguiente ejemplo, se busca en todos los espacios de nombres y se devuelven las métricas denominadas `CustomCount1`.

```
SEARCH(' MetricName="CustomCount1" ', 'Maximum')
```

También puede utilizar pares de nombre-valor de espacios de nombres y dimensiones como nombres de propiedad, tal y como se muestra en los siguientes ejemplos. El primero de estos

ejemplos ilustra que puede utilizar también nombres de propiedad con búsquedas de coincidencia parcial.

```
SEARCH(' InstanceType=micro ', 'Average')
```

```
SEARCH(' InstanceType="t2.micro" Namespace="AWS/EC2" ', 'Average')
```

Expresiones de búsqueda de CloudWatch: caracteres no alfanuméricos

Los caracteres no alfanuméricos sirven como delimitadores e indican dónde los nombres de métricas, dimensiones, espacios de nombres y términos de búsqueda se dividen en tokens. Cuando los términos están tokenizados, los caracteres no alfanuméricos se eliminan y no aparecen en los tokens. Por ejemplo, `Network-Errors_2` genera los tokens `network`, `errors` y `2`.

El término de búsqueda puede incluir cualquier carácter no alfanumérico. Si estos caracteres aparecen en su término de búsqueda, pueden especificar tokens compuestos en una coincidencia parcial. Por ejemplo, todas las búsquedas siguientes buscarían métricas llamadas `Network-Errors-2` o `NetworkErrors2`.

```
network/errors
network+errors
network-errors
Network_Errors
```

Cuando realiza una búsqueda de un valor exacto, todos los caracteres no alfanuméricos utilizados en la búsqueda exacta deben ser los caracteres correctos que aparecen en la cadena que se busca. Por ejemplo, si desea buscar `Network-Errors-2`, la búsqueda de `"Network-Errors-2"` devuelve resultados, pero no así la búsqueda `"Network_Errors_2"`.

Cuando realiza una búsqueda de coincidencia exacta, los siguientes caracteres deben incluirse entre caracteres de escape con una barra diagonal invertida.

```
" \ ( )
```

Por ejemplo, para buscar una coincidencia exacta del nombre de la métrica `Europe\France Traffic(Network)`, utilice el término de búsqueda **`"Europe\\France Traffic\(Network\)"`**.

Expresiones de búsqueda de CloudWatch: operadores booleanos

La búsqueda permite el uso de los operadores booleanos `AND`, `OR` y `NOT` dentro del `SearchTerm`. Los operadores booleanos se incluyen entre las comillas simples que utiliza para entrecomillar todo el término de búsqueda. Los operadores booleanos distinguen entre mayúsculas y minúsculas, por lo que `and`, `or` y `not` no son válidos como operadores booleanos.

Puede utilizar `AND` explícitamente en la búsqueda, como **`SEARCH('{AWS/EC2,InstanceId} network AND packets', 'Average')`**. Si no se utilizan operadores booleanos entre los términos de búsqueda, las búsquedas se realizan como si hubiera un operador `AND`, por lo que **`SEARCH(' {AWS/EC2,InstanceId} network packets ', 'Average')`** produce los mismos resultados de búsqueda.

Utilice `NOT` para excluir subconjuntos de datos en los resultados. Por ejemplo, **`SEARCH(' {AWS/EC2,InstanceId} MetricName="CPUUtilization" NOT i-1234567890123456 ', 'Average')`** devuelve la métrica `CPUUtilization` de todas las instancias, excepto de la instancia `i-1234567890123456`. También puede utilizar una cláusula `NOT` como el único término de búsqueda. Por ejemplo, **`SEARCH( 'NOT Namespace=AWS ', 'Maximum')`** devuelve todas las métricas personalizadas (las métricas con espacios de nombres que no incluyen `AWS`).

Puede utilizar varias frases `NOT` en una consulta. Por ejemplo, **`SEARCH(' {AWS/EC2,InstanceId} MetricName="CPUUtilization" NOT "ProjectA" NOT "ProjectB" ', 'Average')`** devuelve la métrica `CPUUtilization` de todas las instancias de la región, excepto de aquellas con los valores de dimensión `ProjectA` o `ProjectB`.

Puede combinar operadores booleanos para realizar búsquedas más eficaces y detalladas, tal y como se muestra en los siguientes ejemplos. Utilice paréntesis para agrupar los operadores.

Los dos ejemplos siguientes devuelven todos los nombres de métrica que contienen `ReadOps` de los espacios de nombres de EC2 y EBS.

```
SEARCH(' (EC2 OR EBS) AND MetricName=ReadOps ', 'Maximum')
```

```
SEARCH(' (EC2 OR EBS) MetricName=ReadOps ', 'Maximum')
```

El siguiente ejemplo acota la búsqueda anterior a solo los resultados que incluyen `ProjectA`, que podría ser el valor de una dimensión.

```
SEARCH(' (EC2 OR EBS) AND ReadOps AND ProjectA ', 'Maximum')
```

En el ejemplo siguiente se utiliza la agrupación anidada. Devuelve las métricas Lambda para `Errors` de todas las funciones y las `Invocations` de las funciones con nombres que incluyen las cadenas de `ProjectA` o `ProjectB`.

```
SEARCH(' {AWS/Lambda,FunctionName} MetricName="Errors" OR (MetricName="Invocations" AND
 (ProjectA OR ProjectB)) ', 'Average')
```

Expresiones de búsqueda de CloudWatch: uso de expresiones matemáticas

Puede utilizar una expresión de búsqueda dentro de una expresión matemática en un gráfico.

Por ejemplo, **`SUM(SEARCH(' {AWS/Lambda, FunctionName} MetricName="Errors" ', 'Sum'))`** devuelve la suma de la métrica `Errors` de todas las funciones de Lambda.

El uso de líneas distintas para la expresión de búsqueda y la expresión matemática podría producir resultados más útiles. Suponga, por ejemplo, que usa las siguientes dos expresiones en un gráfico. La primera línea muestra líneas `Errors` distintas para cada una de las funciones de Lambda. El ID de esta expresión es `e1`. La segunda línea añade otra línea que muestra la suma de los errores de todas las funciones.

```
SEARCH(' {AWS/Lambda, FunctionName}, MetricName="Errors" ', 'Sum')
SUM(e1)
```

Ejemplos de expresiones de búsqueda de CloudWatch

Los siguientes ejemplos ilustran otros usos y sintaxis de expresiones de búsqueda. Comencemos con una búsqueda de `CPUUtilization` en todas las instancias de la región y luego veremos las variaciones.

Este ejemplo muestra una línea para cada instancia de la región, que muestra la métrica `CPUUtilization` del espacio de nombres `AWS/EC2`.

```
SEARCH(' {AWS/EC2,InstanceId} MetricName="CPUUtilization" ', 'Average')
```

Si `InstanceId` se cambia por `InstanceType`, el gráfico cambia para mostrar una línea para cada tipo de instancia utilizado en la región. Los datos de todas las instancias de cada tipo se agrupan en una línea para ese tipo de instancia.

```
SEARCH(' {AWS/EC2,InstanceType} MetricName="CPUUtilization" ', 'Average')
```

El siguiente ejemplo agrupa la métrica `CPUUtilization` por tipo de instancia y muestra una línea para cada tipo de instancia que incluye la cadena `micro`.

```
SEARCH('{AWS/EC2,InstanceType} InstanceType=micro MetricName="CPUUtilization" ',
 'Average')
```

Este ejemplo acota los resultados del ejemplo anterior, al cambiar `InstanceType` por una búsqueda exacta de instancias t2.micro.

```
SEARCH('{AWS/EC2,InstanceType} InstanceType="t2.micro" MetricName="CPUUtilization" ',
 'Average')
```

La siguiente búsqueda elimina la parte `{metric schema}` de la consulta, por lo que la métrica `CPUUtilization` de todos los espacios de nombres aparece en el gráfico. Esta búsqueda puede devolver bastantes resultados, ya que el gráfico incluye varias líneas para la métrica `CPUUtilization` de cada servicio de AWS, agrupadas por dimensiones diferentes.

```
SEARCH('MetricName="CPUUtilization" ', 'Average')
```

Para reducir un poco estos resultados, puede especificar dos espacios de nombres de métricas específicos.

```
SEARCH('MetricName="CPUUtilization" AND ("AWS/ECS" OR "AWS/ES") ', 'Average')
```

El ejemplo anterior es la única manera de hacer una búsqueda de varios espacios de nombres con una sola consulta de búsqueda, ya que solo se puede especificar un esquema de métrica en cada consulta. Sin embargo, para estructurar más la búsqueda, podría utilizar dos consultas en el gráfico, tal y como se muestra en el siguiente ejemplo. Este ejemplo también añade más estructura al especificar la dimensión que se va a utilizar para agrupar los datos de Amazon ECS.

```
SEARCH('{AWS/ECS ClusterName}, MetricName="CPUUtilization" ', 'Average')
SEARCH(' {AWS/EBS} MetricName="CPUUtilization" ', 'Average')
```

En el siguiente ejemplo se devuelve la métrica Elastic Load Balancing denominada `ConsumedLCUs`, además de todas las métricas o dimensiones de Elastic Load Balancing que contienen el token `flow`.

```
SEARCH('{AWS/NetworkELB, LoadBalancer} "ConsumedLCUs" OR flow ', 'Maximum')
```

En el ejemplo siguiente se utiliza la agrupación anidada. Devuelve las métricas de Lambda para `Errors` de todas las funciones y las `Invocations` de funciones con nombres que incluyen las cadenas de `ProjectA` o `ProjectB`.

```
SEARCH('{AWS/Lambda,FunctionName} MetricName="Errors" OR (MetricName="Invocations" AND
 (ProjectA OR ProjectB)) ', 'Average')
```

El siguiente ejemplo muestra todas las métricas personalizadas, salvo las métricas generadas por servicios de AWS.

```
SEARCH('NOT Namespace=AWS ', 'Average')
```

El siguiente ejemplo muestra las métricas con nombres de métrica, espacios de nombres, nombres de dimensión y valores de dimensión que contienen la cadena `Errors` como parte de su nombre.

```
SEARCH('Errors', 'Average')
```

El siguiente ejemplo acota esa búsqueda para que solo se devuelvan coincidencias exactas. Por ejemplo, esta búsqueda busca el nombre de métrica `Errors`, pero no las métricas llamadas `ConnectionErrors` o `errors`.

```
SEARCH(' "Errors" ', 'Average')
```

En el siguiente ejemplo se muestra cómo especificar nombres que contengan espacios o caracteres especiales en la parte del esquema de métrica del término de búsqueda.

```
SEARCH('{"Custom-Namespace", "Dimension Name With Spaces"}, ErrorCount ', 'Maximum')
```

Ejemplos de expresiones de búsqueda para la observabilidad entre cuentas de CloudWatch

Ejemplos de observabilidad entre cuentas de CloudWatch

Si inicia sesión en una cuenta configurada como cuenta de supervisión para la observabilidad entre cuentas de CloudWatch, puede usar la función SEARCH (BUSCAR) para obtener métricas de cuentas de origen especificadas. Para obtener más información, consulte Observabilidad entre cuentas de CloudWatch.

En el siguiente ejemplo se recuperan todas las métricas de Lambda de la cuenta con ID 111122223333.

```
SEARCH(' AWS/Lambda :aws.AccountId = "111122223333" ', 'Average')
```

En el siguiente ejemplo se recuperan todas las métricas `AWS/EC2` de dos cuentas: 111122223333 y 777788889999.

```
SEARCH(' AWS/EC2 :aws.AccountId = ("111122223333" OR "777788889999") ', 'Average')
```

En el siguiente ejemplo se recuperan todas las métricas `AWS/EC2` de la cuenta de origen 111122223333 y de la propia cuenta de supervisión.

```
SEARCH(' AWS/EC2 :aws.AccountId = ("111122223333" OR 'LOCAL') ', 'Average')
```

En el siguiente ejemplo, se recupera el `SUM` de la métrica `MetaDataToken` de la cuenta `444455556666` con la dimensión `InstanceId`.

```
SEARCH('{AWS/EC2,InstanceId} :aws.AccountId=444455556666 MetricName=\"MetadataNoToken
\"','Sum')
```

Crear un gráfico de CloudWatch con una expresión de búsqueda

En la consola de CloudWatch, puede acceder a la función de búsqueda cuando agrega un gráfico a un panel o mediante la vista Metrics (Métricas).

No es posible crear una alarma con una expresión SEARCH (de búsqueda). Esto se debe a que las expresiones de búsqueda devuelven varias series temporales, y una alarma basada en una expresión matemática sólo puede ver una serie temporal.

Para añadir un gráfico con una expresión de búsqueda a un panel existente

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Dashboards (Paneles) y seleccione un panel.
3. Elija Add widget (Añadir widget).
4. Elija Line (Línea) o Stacked area (Área apilada) y seleccione Configure (Configurar).
5. En la pestaña Graphed metrics (Métricas diagramadas), elija Add a math expression (Añadir una expresión matemática).

6. En Details (Detalles), escriba la expresión de búsqueda que desee. Por ejemplo, **`SEARCH('{AWS/EC2,InstanceId} MetricName="CPUUtilization"', 'Average')`**
7. (Opcional) Para añadir otra expresión de búsqueda o expresión matemática al gráfico, elija Add a math expresión (Añadir una expresión matemática).
8. (Opcional) Después de añadir una expresión de búsqueda, puede especificar una etiqueta dinámica para que aparezca en la leyenda del gráfico de cada métrica. Las etiquetas dinámicas muestran una estadística acerca de la métrica y se actualizan automáticamente cuando se actualiza el panel o el gráfico. Para añadir una etiqueta dinámica, elija Graphed metrics (Métricas representadas gráficamente) y, a continuación, Dynamic labels (Etiquetas dinámicas).

   De forma predeterminada, los valores dinámicos que añade a la etiqueta aparecen al principio de la etiqueta. A continuación, puede elegir el valor de Label (Etiqueta) de la métrica para editar la etiqueta. Para obtener más información, consulte Uso de etiquetas dinámicas.
9. (Opcional) Para añadir una sola métrica al gráfico, elija la pestaña All metrics (Todas las métricas) y desplácese hasta la métrica que desee.
10. (Opcional) Para cambiar el intervalo de tiempo que se muestra en el gráfico, elija custom (personalizado) en la parte superior del gráfico o uno de los períodos de tiempo a la izquierda de custom (personalizado).
11. (Opcional) Las anotaciones horizontales ayudan a los usuarios del panel a ver rápidamente cuándo una métrica ha aumentado hasta un determinado nivel o si la métrica está dentro de un intervalo predefinido. Para añadir una anotación horizontal, elija Graph options (Opciones de gráfico) y después Add horizontal annotation (Añadir anotación horizontal):

    a. En Label (Etiqueta), escriba la etiqueta de la anotación.
    b. En Value (Valor), escriba el valor de métrica en el que aparece la anotación horizontal.
    c. En Fill, especifique si se usará sombreado de relleno con esta anotación. Por ejemplo, elija `Above` o `Below` para el área correspondiente que se rellenará. Si especifica `Between`, aparece otro campo `Value` y se rellena el área del gráfico entre los dos valores.
    d. En Axis (Eje), especifique si los números de `Value` hacen referencia a la métrica asociada con el eje Y izquierdo o con el eje Y derecho, en caso de que el gráfico incluya varias métricas.

       Puede cambiar el color de relleno de una anotación eligiendo el cuadrado de color en la columna izquierda de la anotación.

    Repita estos pasos para agregar varias anotaciones horizontales al mismo gráfico.

Para ocultar una anotación, quite la marca de la casilla en la columna izquierda de dicha anotación.

Para eliminar una anotación, elija x en la columna Actions.

12. (Opcional) Las anotaciones verticales le ayudan a marcar hitos en un gráfico como, por ejemplo, eventos operativos o el principio y el final de una implementación. Para añadir una anotación vertical, elija Graph options (Opciones del gráfico) y después Add vertical annotation (Añadir anotación vertical):

    a. En Label (Etiqueta), escriba la etiqueta de la anotación. Para mostrar solo la fecha y la hora en la anotación, deje el campo Label (Etiqueta) vacío.
    b. Para Date (Fecha), especifique la fecha y la hora en que aparece la anotación vertical.
    c. En Fill (Relleno), especifique si se utiliza sombreado de relleno antes o después de una anotación vertical o entre dos anotaciones verticales. Por ejemplo, elija `Before` o `After` para el área correspondiente que se rellenará. Si especifica `Between`, aparece otro campo `Date` y se rellena el área del gráfico entre los dos valores.

    Repita estos pasos para agregar varias anotaciones verticales al mismo gráfico.

    Para ocultar una anotación, quite la marca de la casilla en la columna izquierda de dicha anotación.

    Para eliminar una anotación, elija x en la columna Actions.
13. Elija Create widget (Crear widget).
14. Elija Save dashboard (Guardar panel).

Para utilizar la vista de métricas para representar gráficamente las métricas buscadas

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, seleccione Métricas.
3. En el campo de búsqueda, escriba los tokens que desea buscar (por ejemplo, **`cpuutilization t2.small`**).

    Se muestran los resultados que coinciden con su búsqueda.
4. Para representar gráficamente todas las métricas que coinciden con su búsqueda, elija Graph search (Diagramar búsqueda).

o

Para acotar la búsqueda, elija uno de los espacios de nombres que apareció en los resultados de la búsqueda.

5. Si ha seleccionado un espacio de nombres para restringir los resultados, puede hacer lo siguiente:

   a. Para representar gráficamente una o varias métricas, seleccione la casilla de verificación junto a cada métrica. Para seleccionar todas las métricas, seleccione la casilla de verificación en la fila de encabezado de la tabla.

   b. Para acotar la búsqueda, coloque el cursor sobre un nombre de métrica y elija Add to search (Añadir a búsqueda) o Search for this only (Buscar solo esto).

   c. Para ver la ayuda de una métrica, elija el nombre de la métrica y, a continuación, seleccione What is this?.

   La métricas seleccionadas aparecen en el gráfico.

6. (Opcional) Seleccione uno de los botones de la barra de búsqueda para editar esa parte del término de búsqueda.

7. (Opcional) Para agregar su gráfico a un panel, elija Actions (Acciones) y después Add to dashboard (Añadir al panel).

Obtener estadísticas de una métrica

Definiciones de estadísticas de CloudWatch

Las estadísticas son agregaciones de datos de métricas correspondientes a periodos especificados. Cuando grafica o recupera las estadísticas para una métrica, especifique el Período de tiempo, por ejemplo, cinco minutos, que se utilizarán para calcular cada valor estadístico. Por ejemplo, si el Periodo (Período) es de cinco minutos, la Sum (Suma) es la suma de todos los valores de muestra recopilados durante el período de cinco minutos, mientras que el Minimum (Mínimo) es el valor más bajo recogido durante el período de cinco minutos.

CloudWatch es compatible con las siguientes estadísticas para métricas.

- Recuento de muestras es el número de puntos de datos durante el período.

- Sum (Suma) es la suma de los valores de todos los puntos de datos recopilados durante el período.
- Promedio es el valor de `Sum/SampleCount` durante el período especificado.
- Mínimo es el valor más bajo observado durante el período especificado.
- Máximo es el valor más alto observado durante el período especificado.
- Percentil (p) indica la posición relativa de un valor en un conjunto de datos. Por ejemplo, p95 es el percentile número 95 y significa que el 95 por ciento de los datos en el período está por debajo de este valor y el 5 por ciento de los datos está por encima del mismo. Los percentiles le ayudan a entender mejor la distribución de los datos de métricas.
- Media recortada (TM) es la media de todos los valores que se encuentran entre dos límites especificados. Los valores fuera de los límites se ignoran cuando se calcula la media. Los límites se definen como uno o dos números entre 0 y 100, hasta con 10 decimales. Los números pueden ser valores absolutos o porcentajes. Por ejemplo, tm90 calcula el promedio después de eliminar el 10 % de los puntos de datos con los valores más altos. TM (2 %:98 %) calcula el promedio después de eliminar los puntos de datos más bajos del 2 % y los puntos de datos más altos del 2 %. TM (150:1000) calcula el promedio después de eliminar todos los puntos de datos que son inferiores o iguales a 150 o superiores a 1000.
- Media semiintercuartil (IQM) es la media recortada del rango intercuartil o el 50 % de los valores. Equivale a TM (25 %:75 %).
- Media winsorizada (WM) es similar a la media recortada. Sin embargo, con la media winsorizada, los valores que están fuera del límite no se ignoran, sino que se consideran iguales al valor en el borde del límite apropiado. Después de esta normalización, se calcula el promedio. Los contornos se definen como uno o dos números entre 0 y 100, con hasta 10 decimales. Por ejemplo, wm98 calcula el promedio mientras se trata el 2 % de los valores más altos para que sea igual al valor en el percentil número 98. WM (10 %:90 %) calcula el promedio mientras se trata el 10 % más alto de los puntos de datos como el valor del límite del 90 %, y se trata el 10 % más bajo de los puntos de datos como el valor del límite del 10 %.
- Rango del percentil (PR) es el porcentaje de valores que cumplen un umbral fijo. Por ejemplo, PR (:300) devuelve el porcentaje de puntos de datos que tienen un valor de 300 o menos. PR (100:2000) devuelve el porcentaje de puntos de datos que tienen un valor entre 100 y 2000.

  El rango percentil es exclusivo en el límite inferior e inclusivo en el límite superior.
- Recuento recortado (TC) es el número de puntos de datos en el rango elegido para una estadística media recortada. Por ejemplo, tc90 devuelve el número de puntos de datos sin incluir ningún punto

de datos que se sitúe en el 10 % más alto de los valores. TC (0,005:0,030) devuelve el número de puntos de datos con valores entre 0,005 (exclusivo) y 0,030 (inclusive).

- Suma recortada (TS) es la suma de los valores de los puntos de datos en un rango elegido para una estadística media recortada. Es equivalente a (Media recortada) * (Recuento recortado). Por ejemplo, ts90 devuelve la suma de los puntos de datos sin incluir los puntos de datos que se sitúan en el 10 % más alto de los valores. TS (80 %:) devuelve la suma de los valores de punto de datos, sin incluir ningún punto de datos con valores en el 80 % más bajo del rango de valores.

> **Note**
>
> Si para calcular la media recortada, el recuento recortado, la suma recortada y la media winsorizada define dos límites como valores fijos en lugar de porcentajes, el cálculo incluirá valores iguales al límite superior, pero no valores iguales al límite inferior.

Sintaxis

Para calcular la media recortada, el recuento recortado, la suma recortada y la media winsorizada, se aplican las siguientes reglas de sintaxis:

- El uso de paréntesis con uno o dos números con signos de porcentaje define los límites que se utilizarán como los valores del conjunto de datos que se encuentran entre los dos percentiles especificados. Por ejemplo, TM (10 %:90 %) utiliza sólo los valores entre los percentiles número 10 y 90. TM (:95 %) utiliza los valores desde el extremo más bajo de los datos configurados hasta el percentil número 95, e ignora el 5 % de los puntos de datos con los valores más altos.
- El uso de paréntesis con uno o dos números sin signos de porcentaje define los límites que se utilizarán como los valores del conjunto de datos que se encuentran entre los valores explícitos que especifique. Por ejemplo, TC (80:500) utiliza sólo los valores que están entre 80 (exclusivo) y 500 (inclusive). TC (:0,5) utiliza sólo los valores que equivalen a 0,5 o son inferiores.
- El uso de un número sin paréntesis calcula con porcentajes, e ignora los puntos de datos que son superiores al percentil especificado. Por ejemplo, tm99 calcula la media mientras ignora el 1 % de los puntos de datos con el valor más alto. Es lo mismo que TM (:99 %).
- La media recortada, el recuento recortado, la suma recortada y la media winsorizada se pueden abreviar con letras mayúsculas al especificar un rango, como TM (5 %:95 %), TM (100:200) o TM (:95 %). Solo se pueden abreviar con letras minúsculas cuando se especifica un número, por ejemplo, tm99.

Casos de uso de estadísticas

- Media recortada es más útil para métricas con un tamaño de muestra grande, como la latencia de la página web. Por ejemplo, tm99 ignora los valores extremos altos atípicos que podrían ser el resultado de problemas de red o errores humanos, para dar un número más preciso para la latencia promedio de las solicitudes típicas. Del mismo modo, TM (10 %:) ignora el 10 % más bajo de los valores de latencia, como los resultantes de los aciertos de la caché. Y TM (10 %:99 %) excluye ambos tipos de valores atípicos. Se recomienda utilizar la media recortada para supervisar la latencia.
- Es una buena idea vigilar el recuento recortado cada vez que utilice la media recortada para asegurarse de que el número de valores que se utilizan en los cálculos de la media recortada sea suficiente para obtener un valor estadísticamente significativo.
- El rango de percentiles le permite poner valores en 'bins' de rangos, lo que le permitirá usar esto para crear manualmente un histograma. Para ello, divida los valores en varios bins, como PR (:1), PR (1:5), PR (5:10), y PR (10:). Coloque cada uno de los bins en una visualización como gráficos de barras, y tendrá un histograma.

  El rango percentil es exclusivo en el límite inferior e inclusivo en el límite superior.

Percentiles versus media recortada

Un percentil, como p99, y una media recortada, como tm99, miden valores similares, pero no idénticos. Ambos, p99 y tm99, ignoran el 1 % de los puntos de datos con los valores más altos, que se consideran valores atípicos. Luego, p99 es el valor máximo del 99 % restante, mientras que tm99 es el promedio del 99 % restante. Si mira la latencia de las solicitudes web, p99 muestra la peor experiencia del cliente e ignora los valores atípicos, mientras que tm99 le indica la experiencia promedio del cliente e ignora los valores atípicos.

La media recortada es una buena estadística de latencia para ver si usted busca optimizar la experiencia del cliente.

Requisitos para usar percentiles, media recortada y algunas otras estadísticas

CloudWatch necesita puntos de datos sin procesar para calcular las siguientes estadísticas:

- Percentiles
- Media recortada
- Media intercuartil

- Media winsorizada
- Suma recortada
- Recuento recortado
- Rango de percentiles

Si publica datos para obtener estadísticas personalizadas a través de un conjunto de estadísticas en lugar de datos sin procesar, solo puede recuperar estos tipos de estadísticas para estos datos si una de las siguientes condiciones es VERDADERA:

- El valor de SampleCount del conjunto estadístico es 1 y el mínimo, el máximo y la suma son todos iguales.
- El mínimo y el máximo son iguales y la suma es igual al mínimo multiplicado por SampleCount.

Los siguientes servicios de AWS incluyen métricas que admiten este tipo de estadísticas.

- API Gateway
- Application Load Balancer
- Amazon EC2
- Elastic Load Balancing
- Kinesis
- Amazon RDS

Además, estos tipos de estadísticas no están disponibles para las métricas cuando alguno de los valores de las métricas es un número negativo.

Los siguientes ejemplos muestran cómo se obtienen las estadísticas para las métricas de CloudWatch para sus recursos, como las instancias EC2.

Ejemplos

- Obtener estadísticas de un recurso específico
- Acumular estadísticas a través de recursos
- Acumular estadísticas por grupo de Auto Scaling
- Acumular estadísticas por imagen de máquina de Amazon (AMI)

Obtener estadísticas de un recurso específico

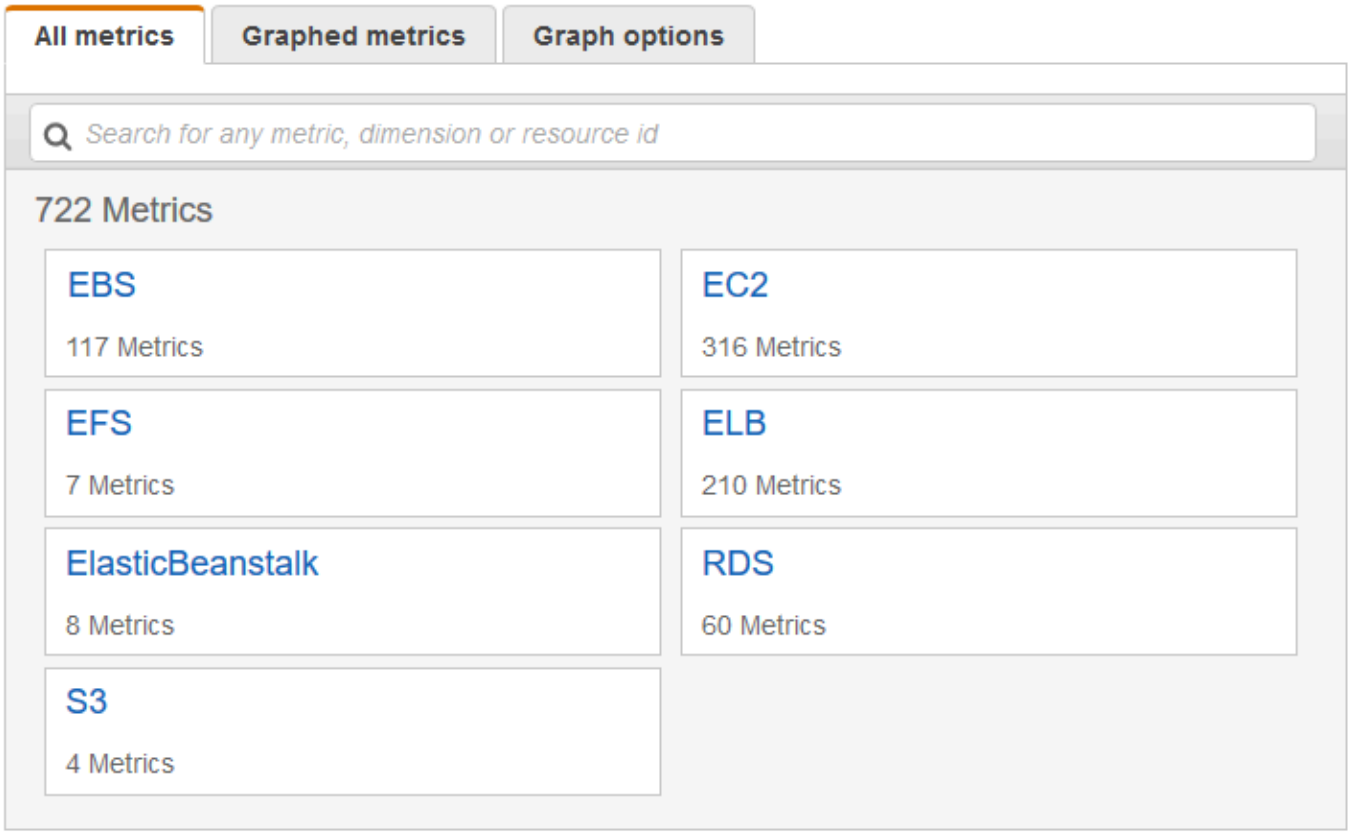
En el siguiente ejemplo se muestra cómo determinar la utilización de CPU máxima de una instancia EC2 específica.

Requisitos

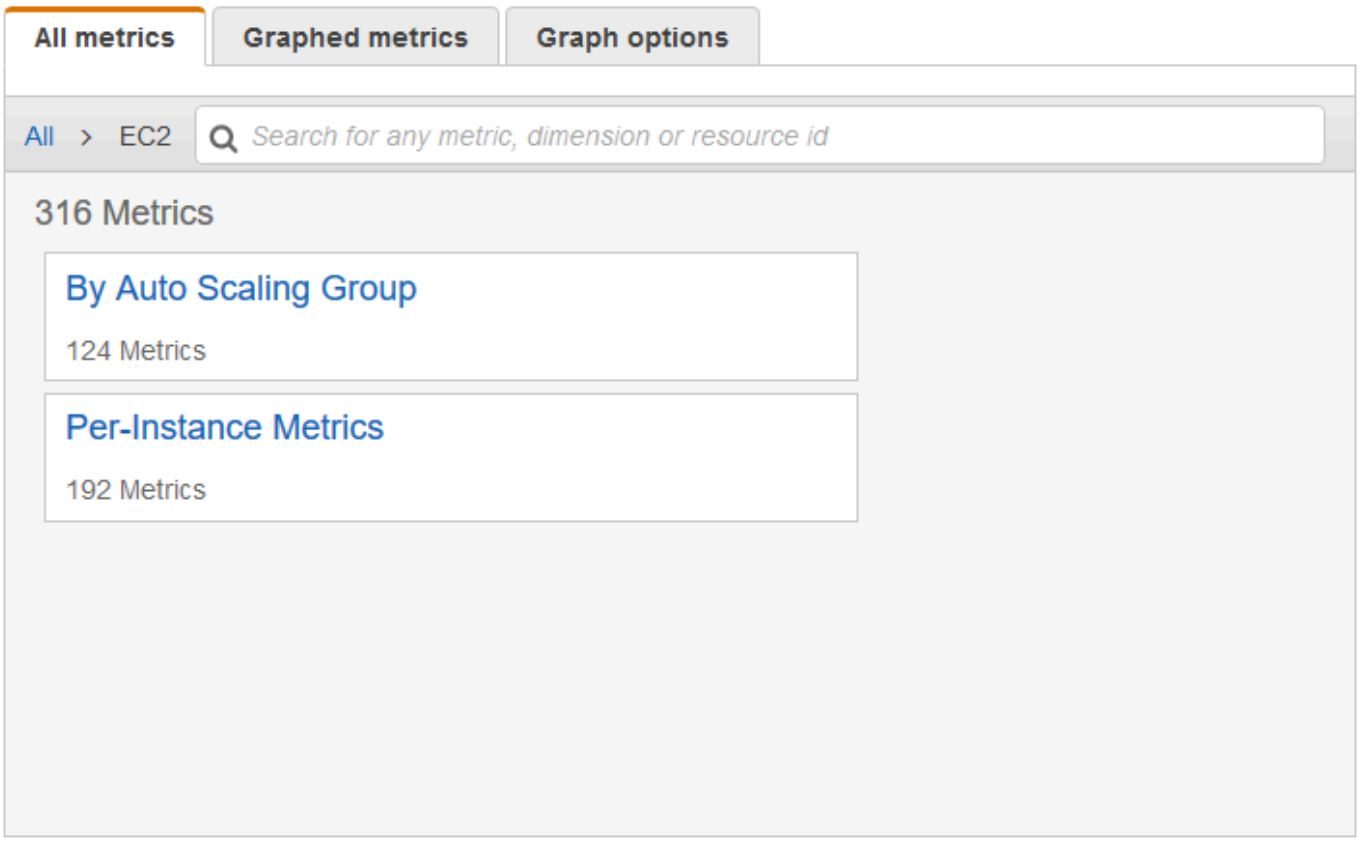
- Debe tener el ID de la instancia. Puede obtener el ID de la instancia mediante la consola de Amazon EC2 o el comando describe-instances.
- De forma predeterminada, la supervisión básica está habilitada, pero puede activar la supervisión detallada. Para obtener más información, consulte Enable or Disable Detailed Monitoring for Your Instances (Habilitación o desactivación de la supervisión detallada para las instancias) en la Guía del usuario de Amazon EC2 para instancias de Linux.

Para mostrar la utilización promedio de la CPU para una instancia concreta mediante la consola

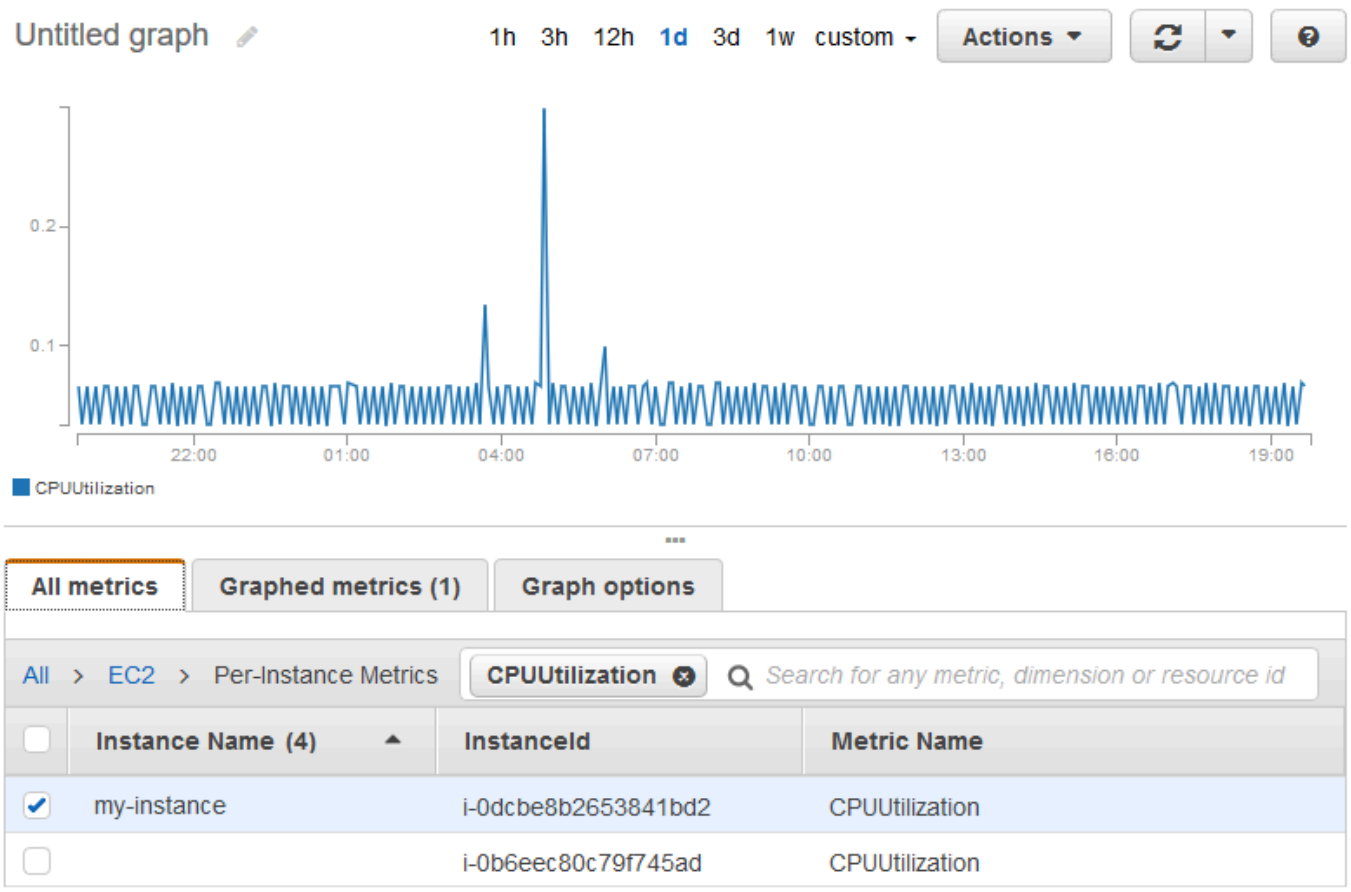
1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, seleccione Métricas.
3. Seleccione el espacio de nombres de métricas EC2.



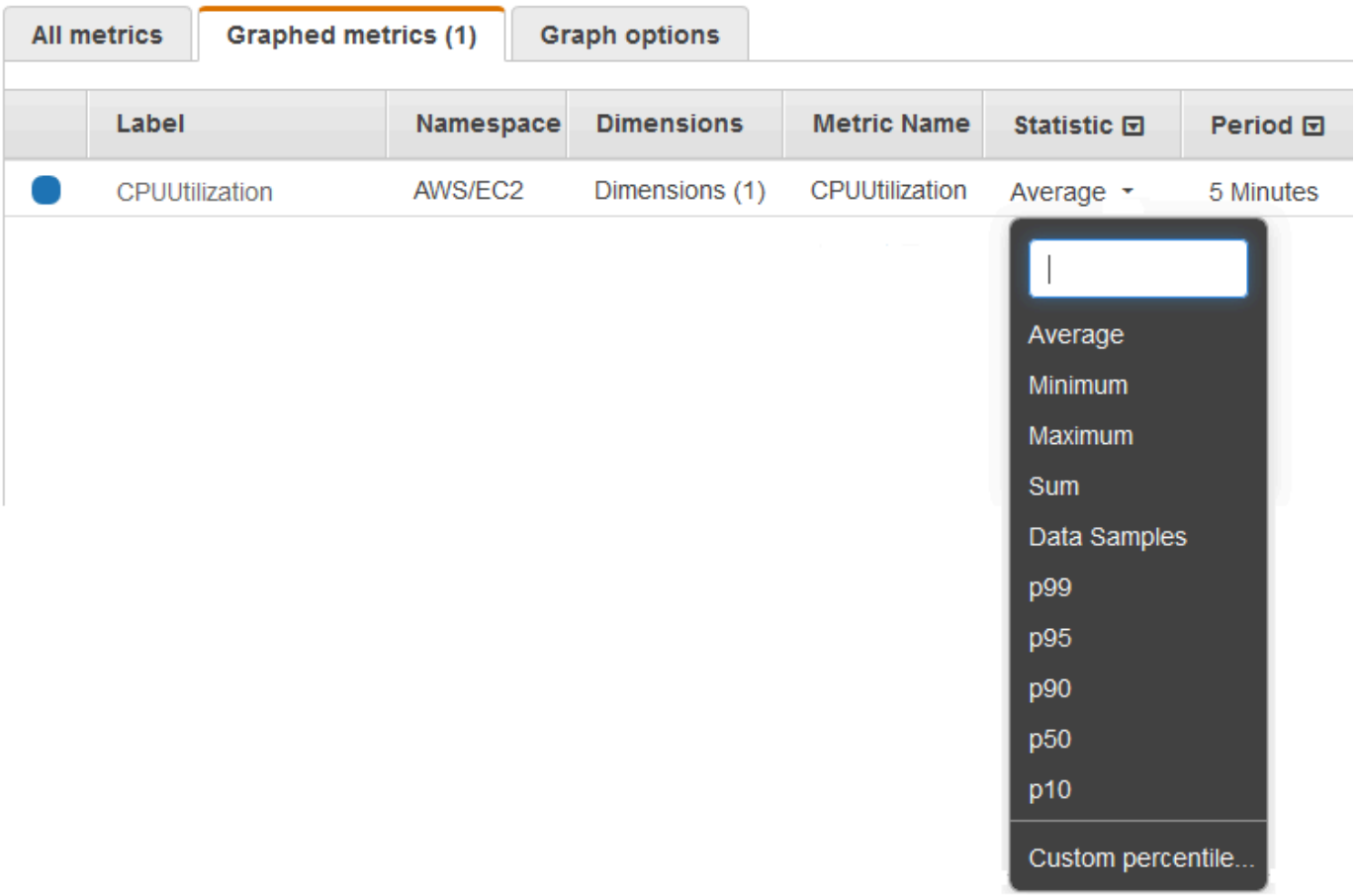
4. Seleccione la dimensión Per-Instance Metrics (Métricas por instancia).



5. En el campo de búsqueda, escriba **`CPUUtilization`** y pulse Intro. Seleccione la fila de la instancia concreta, que muestra un gráfico para la métrica `CPUUtilization` de la instancia. Para cambiar el nombre del gráfico, seleccione el icono de lápiz. Para cambiar el intervalo de tiempo, seleccione uno de los valores predefinidos o elija custom (personalizado).



6. Para cambiar la estadística, elija la pestaña Graphed metrics. Elija el encabezado de columna o un valor individual y, a continuación, elija una de las estadísticas o percentiles predefinidos o especifique un percentil personalizado (por ejemplo, **`p99.999`**).



7. Para cambiar el periodo, elija la pestaña Graphed metrics. Elija el encabezado de columna o un valor individual y, a continuación, elija un valor diferente.

Para obtener la utilización de la CPU por cada instancia EC2 mediante la AWS CLI

Utilice el comando get-metric-statistics como se indica a continuación para obtener la métrica `CPUUtilization` para la instancia especificada.

```
aws cloudwatch get-metric-statistics --namespace AWS/EC2 --metric-name CPUUtilization \
--dimensions Name=InstanceId,Value=i-1234567890abcdef0 --statistics Maximum \
--start-time 2016-10-18T23:18:00 --end-time 2016-10-19T23:18:00 --period 360
```

Las estadísticas devueltas son valores de seis minutos para el intervalo de tiempo solicitado de 24 horas. Cada valor representa el porcentaje máximo de utilización de CPU para la instancia especificada en un periodo de tiempo particular de seis minutos. Los puntos de datos no se devuelven en orden cronológico. A continuación se muestra el comienzo de la salida de ejemplo (la salida completa incluye todos los puntos de datos para cada 6 minutos del periodo de 24 horas).

```
{
    "Datapoints": [
        {
            "Timestamp": "2016-10-19T00:18:00Z",
            "Maximum": 0.33000000000000002,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2016-10-19T03:18:00Z",
            "Maximum": 99.670000000000002,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2016-10-19T07:18:00Z",
            "Maximum": 0.34000000000000002,
            "Unit": "Percent"
        },
        ...
    ],
    "Label": "CPUUtilization"
}
```

Acumular estadísticas a través de recursos

Puede acumular las métricas para recursos de AWS a través de múltiples recursos. Las métricas están completamente separadas entre las Regiones, pero puede utilizar cálculos métricos para acumular métricas entre las Regiones. Para obtener más información, consulte Uso de la calculadora de métricas.

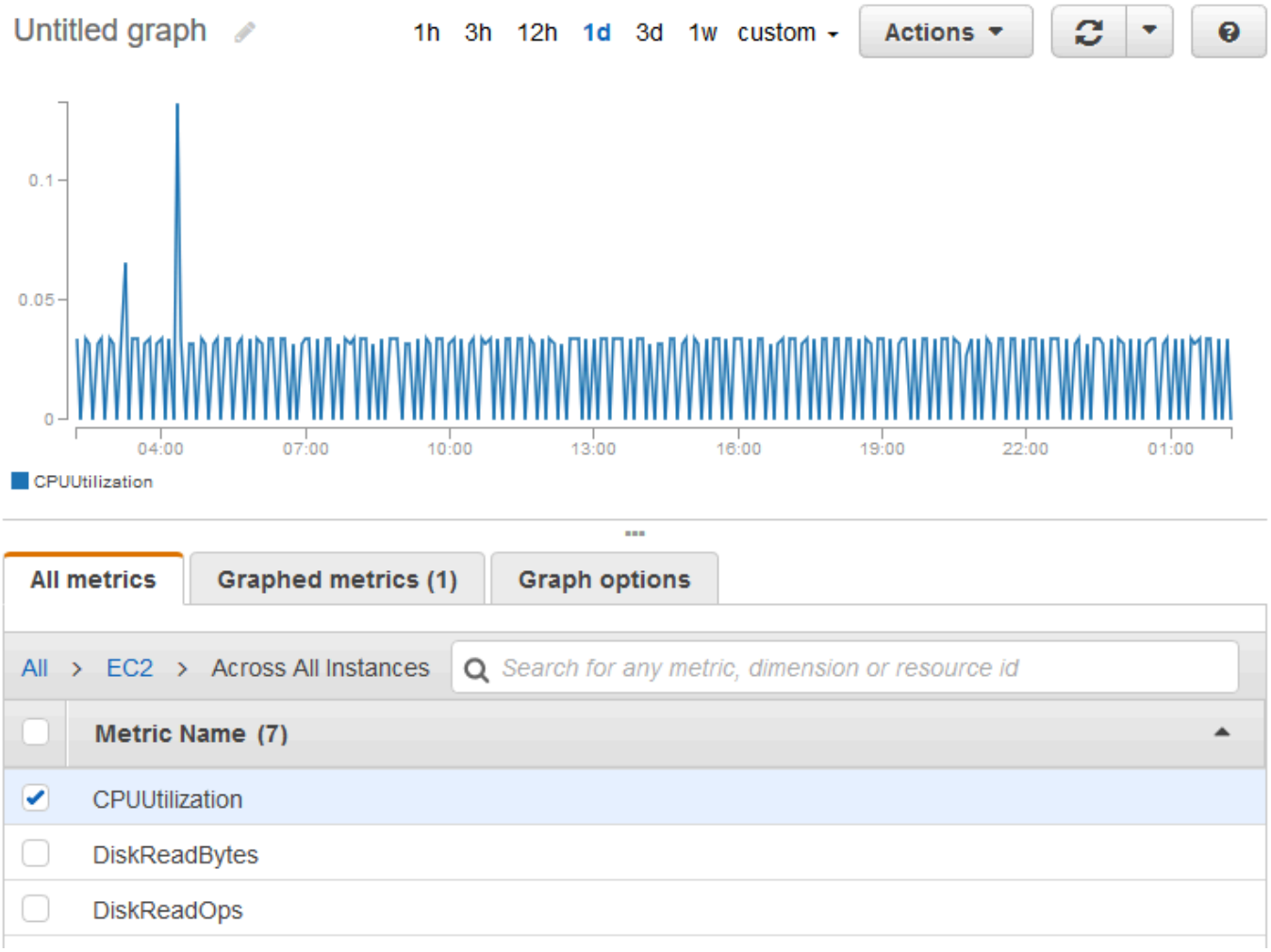
Por ejemplo, puede acumular estadísticas para las instancias EC2 que tengan la supervisión detallada habilitada. Las instancias que utilizan la supervisión básica no están incluidas. Por lo tanto, debe habilitar la supervisión detallada (a un cargo adicional), que proporciona datos en periodos de 1 minuto. Para obtener más información, consulte Enable or Disable Detailed Monitoring for Your Instances (Habilitación o desactivación de la supervisión detallada para las instancias) en la Guía del usuario de Amazon EC2 para instancias de Linux.

Este ejemplo muestra cómo obtener el uso promedio de CPU para sus instancias EC2. Dado que no se especifica ninguna dimensión, CloudWatch devuelve estadísticas para todas las dimensiones en el espacio de nombres `AWS/EC2`. Para obtener estadísticas para otras métricas, consulte Servicios de AWS que publican métricas de CloudWatch.

> **Important**
>
> Esta técnica para recuperar todas las dimensiones mediante un espacio de nombres de AWS no funciona para espacios de nombres personalizados que publique en CloudWatch. Con el uso de espacios de nombres personalizados, debe especificar el conjunto completo de dimensiones que hay asociadas a cualquier punto de datos dado para recuperar estadísticas que incluyen el punto de datos.

Para mostrar la utilización media de la CPU para las instancias EC2

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, seleccione Métricas.
3. Elija el espacio de nombres de EC2 y seleccione Across All Instances.
4. Seleccione la fila que contiene `CPUUtilization`, que muestra un gráfico para la métrica de todas sus instancias EC2. Para cambiar el nombre del gráfico, seleccione el icono de lápiz. Para cambiar el intervalo de tiempo, seleccione uno de los valores predefinidos o elija custom (personalizado).



5. Para cambiar la estadística, elija la pestaña Graphed metrics. Elija el encabezado de columna o un valor individual y, a continuación, elija una de las estadísticas o percentiles predefinidos o especifique un percentil personalizado (por ejemplo, **`p95.45`**).
6. Para cambiar el periodo, elija la pestaña Graphed metrics. Elija el encabezado de columna o un valor individual y, a continuación, elija un valor diferente.

Para obtener la utilización de CPU promedio en sus instancias EC2 mediante la utilización de la AWS CLI

Utilice el comando get-metric-statistics como se indica a continuación:

```
aws cloudwatch get-metric-statistics --namespace AWS/EC2 --metric-name CPUUtilization
 --statistics "Average" "SampleCount" \
--start-time 2016-10-11T23:18:00 --end-time 2016-10-12T23:18:00 --period 3600
```

A continuación, se muestra un ejemplo de la salida:

```
{
    "Datapoints": [
        {
            "SampleCount": 238.0,
            "Timestamp": "2016-10-12T07:18:00Z",
            "Average": 0.038235294117647062,
            "Unit": "Percent"
        },
        {
            "SampleCount": 240.0,
            "Timestamp": "2016-10-12T09:18:00Z",
            "Average": 0.16670833333333332,
            "Unit": "Percent"
        },
        {
            "SampleCount": 238.0,
            "Timestamp": "2016-10-11T23:18:00Z",
            "Average": 0.041596638655462197,
            "Unit": "Percent"
        },
        ...
    ],
    "Label": "CPUUtilization"
}
```

Acumular estadísticas por grupo de Auto Scaling

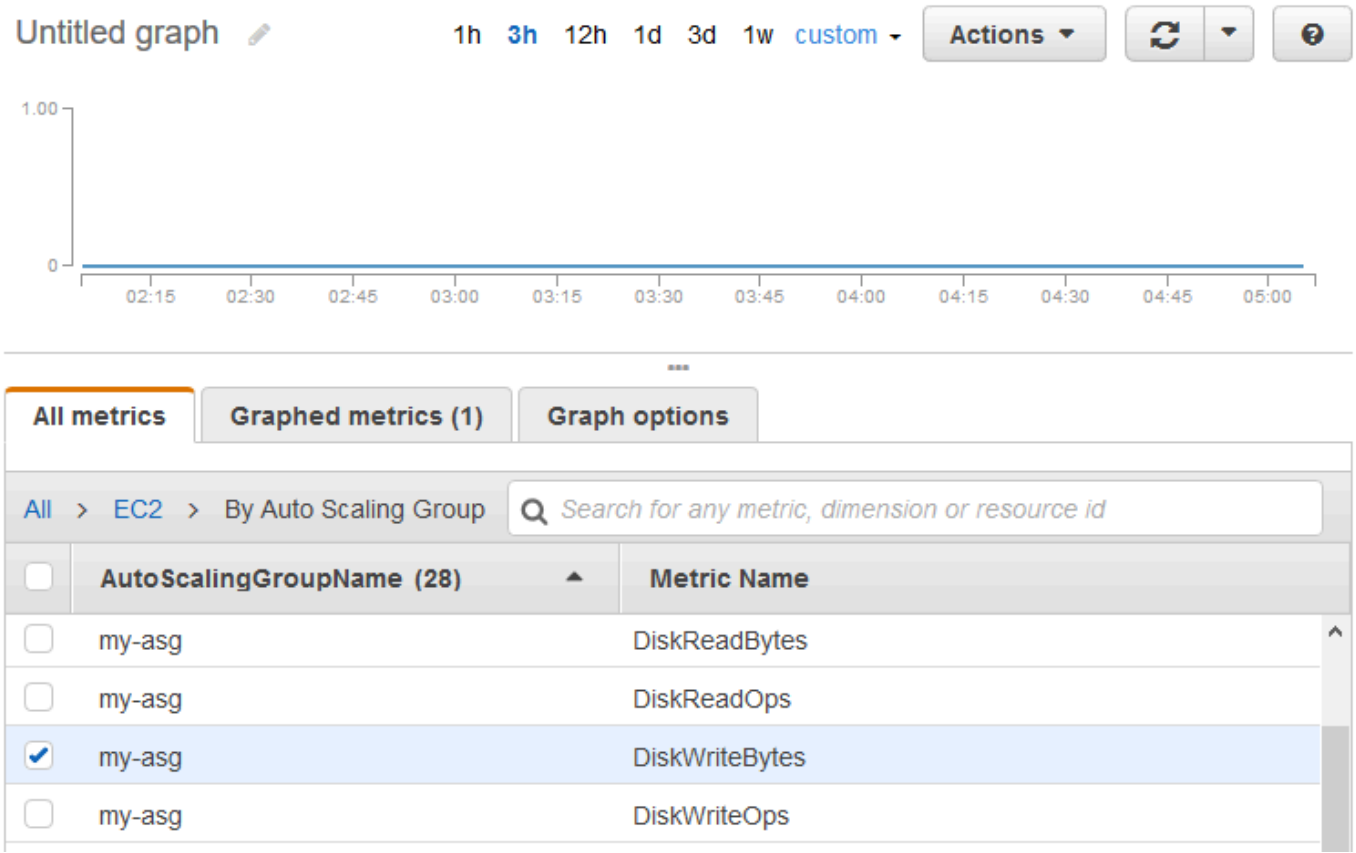
Puede acumular estadísticas para las instancias de EC2 en un grupo de Auto Scaling. Las métricas están completamente separadas entre las Regiones, pero puede usar cálculos métricos de CloudWatch para acumular y transformar las métricas desde varias Regiones. También puede utilizar el panel de control entre cuentas para realizar cálculos matemáticos de métricas en métricas de distintas cuentas.

Este ejemplo muestra cómo se obtienen los bytes totales que se registran en el disco para un grupo de Auto Scaling. El total se calcula para períodos de un minuto en un intervalo de 24 horas en todas las instancias EC2 en el grupo de Auto Scaling especificado.

Para visualizar DiskWriteBytes para las instancias en un grupo de Auto Scaling mediante la consola

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.

2. En el panel de navegación, seleccione Metrics (Métricas).
3. Elija el espacio de nombres EC2 y, a continuación, seleccione By Auto Scaling Group (Por grupo de Auto Scaling).
4. Seleccione la fila para la métrica DiskWriteBytes y el grupo de Auto Scaling específico que muestra un gráfico para la métrica para las instancias en el grupo de Auto Scaling. Para cambiar el nombre del gráfico, seleccione el icono de lápiz. Para cambiar el intervalo de tiempo, seleccione uno de los valores predefinidos o elija custom (personalizado).



5. Para cambiar la estadística, elija la pestaña Graphed metrics. Elija el encabezado de columna o un valor individual y, a continuación, elija una de las estadísticas o percentiles predefinidos o especifique un percentil personalizado (por ejemplo, **`p95.45`**).
6. Para cambiar el periodo, elija la pestaña Graphed metrics. Elija el encabezado de columna o un valor individual y, a continuación, elija un valor diferente.

Para obtener DiskWriteBytes para las instancias en un grupo de Auto Scaling mediante AWS CLI

Utilice el comando get-metric-statistics como se indica a continuación.

```
aws cloudwatch get-metric-statistics --namespace AWS/EC2 --metric-name DiskWriteBytes
```

```
--dimensions Name=AutoScalingGroupName,Value=my-asg --statistics "Sum" "SampleCount" \
--start-time 2016-10-16T23:18:00 --end-time 2016-10-18T23:18:00 --period 360
```

A continuación, se muestra un ejemplo del resultado.

```
{
    "Datapoints": [
        {
            "SampleCount": 18.0,
            "Timestamp": "2016-10-19T21:36:00Z",
            "Sum": 0.0,
            "Unit": "Bytes"
        },
        {
            "SampleCount": 5.0,
            "Timestamp": "2016-10-19T21:42:00Z",
            "Sum": 0.0,
            "Unit": "Bytes"
        }
    ],
    "Label": "DiskWriteBytes"
}
```

Acumular estadísticas por imagen de máquina de Amazon (AMI)

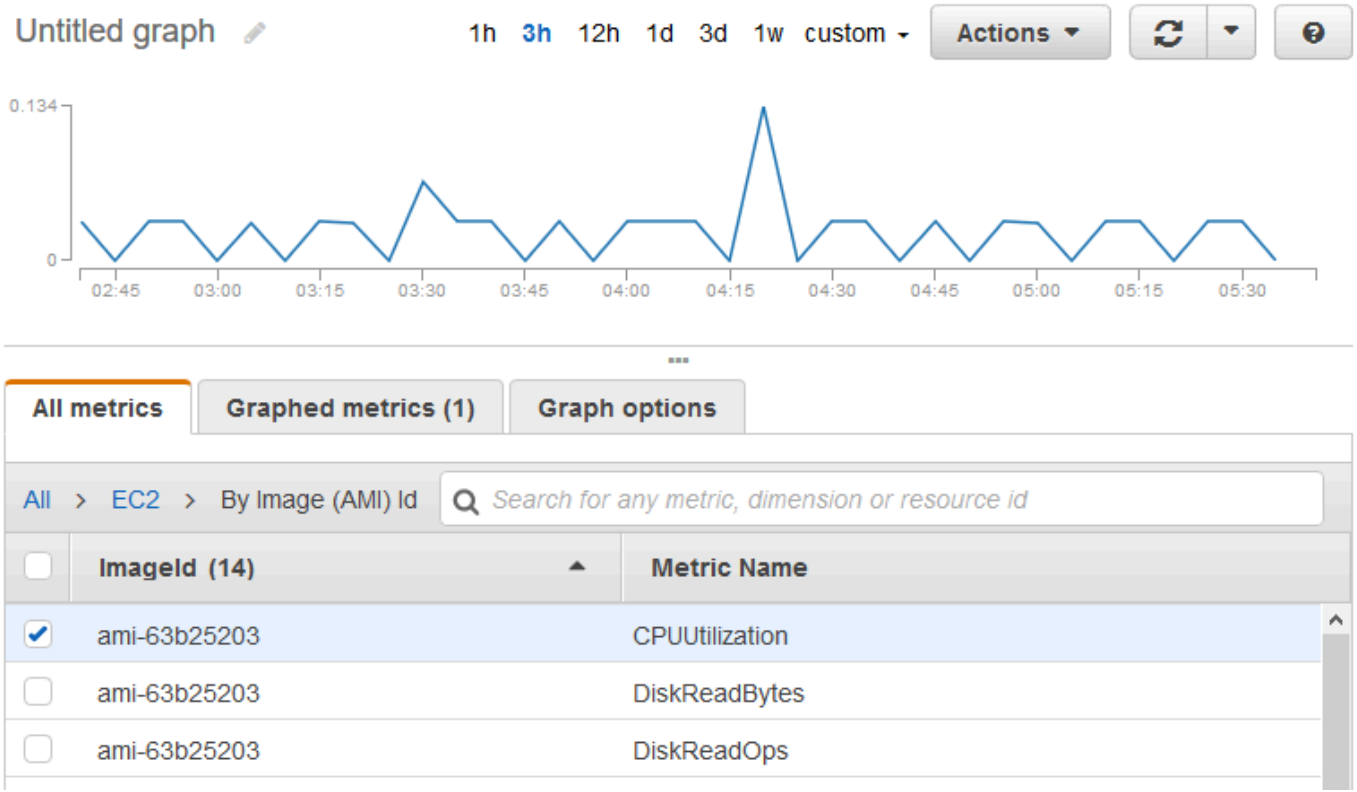
Puede acumular estadísticas para las instancias EC2 que tengan la supervisión detallada habilitada. Las instancias que utilizan la supervisión básica no están incluidas. Para obtener más información, consulte Enable or Disable Detailed Monitoring for Your Instances (Habilitación o desactivación de la supervisión detallada para las instancias) en la Guía del usuario de Amazon EC2 para instancias de Linux.

Este ejemplo muestra cómo determinar la utilización promedio de la CPU de todas las instancias que utilizan la AMI especificada. La media está por encima de intervalos de tiempo de 60 segundos durante un periodo de un día.

Para mostrar el uso de CPU promedio por AMI mediante la consola

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, seleccione Metrics (Métricas).
3. Elija el espacio de nombres EC2 y, a continuación, seleccione By Image (AMI) Id (Por ID de imagen (AMI)).

4. Seleccione la fila para la métrica `CPUUtilization` y la AMI específica, que muestra un gráfico para la métrica para la AMI especificada. Para cambiar el nombre del gráfico, seleccione el icono de lápiz. Para cambiar el intervalo de tiempo, seleccione uno de los valores predefinidos o elija custom (personalizado).



5. Para cambiar la estadística, elija la pestaña Graphed metrics. Elija el encabezado de columna o un valor individual y, a continuación, elija una de las estadísticas o percentiles predefinidos o especifique un percentil personalizado (por ejemplo, **`p95.45`**).
6. Para cambiar el periodo, elija la pestaña Graphed metrics. Elija el encabezado de columna o un valor individual y, a continuación, elija un valor diferente.

Para obtener la utilización de la CPU promedio por AMI con la AWS CLI

Utilice el comando get-metric-statistics como se indica a continuación.

```
aws cloudwatch get-metric-statistics --namespace AWS/EC2 --metric-name CPUUtilization \
--dimensions Name=ImageId,Value=ami-3c47a355 --statistics Average \
--start-time 2016-10-10T00:00:00 --end-time 2016-10-11T00:00:00 --period 3600
```

La operación devuelve estadísticas que son valores de una hora para el intervalo de un día. Cada valor representa un porcentaje de utilización de CPU promedio para instancias EC2 que ejecutan la AMI especificada. A continuación, se muestra un ejemplo del resultado.


```
{
    "Datapoints": [
        {
            "Timestamp": "2016-10-10T07:00:00Z",
            "Average": 0.041000000000000009,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2016-10-10T14:00:00Z",
            "Average": 0.079579831932773085,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2016-10-10T06:00:00Z",
            "Average": 0.036000000000000011,
            "Unit": "Percent"
        },
        ...
    ],
    "Label": "CPUUtilization"
}
```


Publicar métricas personalizadas de

Puede publicar sus propias métricas en CloudWatch mediante AWS CLI o una API. Puede ver gráficos de estadísticas de las métricas publicadas con la AWS Management Console.

CloudWatch almacena datos de una métrica como una serie de puntos de datos. Cada punto de datos tiene una marca temporal asociada. Puede incluso publicar un conjunto de puntos de datos acumulados denominado conjunto estadístico.

Temas

- Métricas de alta resolución
- Uso de dimensiones
- Publicar puntos de datos únicos
- Publicar conjuntos estadísticos

- Publicar el valor cero
- Dejar de publicar métricas

Métricas de alta resolución

Cada métrica es una de las siguientes:

- Resolución estándar, con datos cuya granularidad es de un minuto
- Alta resolución, con datos cuya granularidad es de un segundo

De forma predeterminada, las métricas producidas por los servicios de AWS son de resolución estándar. Al publicar una métrica personalizada, puede definirla como de resolución estándar o de alta resolución. Cuando publica una métrica de alta resolución, CloudWatch la almacena con una resolución de 1 segundo, y puede leerla y recuperarla con un periodo de 1 segundo, 5 segundos, 10 segundos, 30 segundos o cualquier múltiplo de 60 segundos.

Las métricas de alta resolución pueden ofrecerle más información inmediata acerca de las actividades de su aplicación, cuya duración sea inferior a un minuto. Tenga en cuenta que cada llamada a `PutMetricData` para una métrica personalizada se cobra; por tanto, realizar llamadas a `PutMetricData` con más frecuencia en una métrica de alta resolución podría derivar en cargos más elevados. Para obtener más información sobre los precios de CloudWatch, consulte Precios de Amazon CloudWatch.

Si configura una alarma en una métrica de alta resolución, puede especificar una alarma de alta resolución con un periodo de 10 o 30 segundos, o puede definir una alarma normal con un periodo de cualquier múltiplo de 60 segundos. El cargo es mayor para las alarmas de alta resolución con un periodo de 10 o 30 segundos.

Uso de dimensiones

En métricas personalizadas, el parámetro `--dimensions` es habitual. Una dimensión aclara mejor qué es la métrica y qué datos almacena. Puede tener hasta 30 dimensiones asignadas a una métrica y cada dimensión se define mediante un par de nombre y valor.

La forma en que especifique una dimensión es distinta cuando se usan distintos comandos. Con put-metric-data, especifica cada dimensión como *`MyName`*=*`MyValue`*, y con get-metric-statistics o put-metric-alarm utiliza el formato `Name=`*`MyName`*, `Value=`*`MyValue`*. Por ejemplo, el siguiente comando publica una métrica `Buffers` con dos dimensiones denominadas `InstanceId` e `InstanceType`.

```
aws cloudwatch put-metric-data --metric-name Buffers --namespace MyNameSpace --unit
 Bytes --value 231434333 --dimensions InstanceId=1-23456789,InstanceType=m1.small
```

Este comando recupera estadísticas para la misma métrica. Separe las partes Nombre y Valor de una dimensión con comas, pero si tiene varias dimensiones, utilice un espacio entre una dimensión y la siguiente.

```
aws cloudwatch get-metric-statistics --metric-name Buffers --namespace MyNameSpace --
dimensions Name=InstanceId,Value=1-23456789 Name=InstanceType,Value=m1.small --start-
time 2016-10-15T04:00:00Z --end-time 2016-10-19T07:00:00Z --statistics Average --period
 60
```

Si una métrica incluye varias dimensiones, debe especificar un valor para cada dimensión definida cuando utilice get-metric-statistics. Por ejemplo, la métrica `BucketSizeBytes` de Amazon S3 incluye las dimensiones `BucketName` y `StorageType`, por lo tanto, debe especificar ambas dimensiones con get-metric-statistics.

```
aws cloudwatch get-metric-statistics --metric-name BucketSizeBytes --start-time
 2017-01-23T14:23:00Z --end-time 2017-01-26T19:30:00Z --period 3600 --namespace
 AWS/S3 --statistics Maximum --dimensions Name=BucketName,Value=MyBucketName
 Name=StorageType,Value=StandardStorage --output table
```

Para ver qué dimensiones hay definidas en una métrica, utilice el comando list-metrics.

Publicar puntos de datos únicos

Para publicar un punto de datos único para una métrica nueva o existente, utilice el comando put-metric-data con un valor y marca temporal. Por ejemplo, cada una de las siguientes acciones publica un punto de datos.

```
aws cloudwatch put-metric-data --metric-name PageViewCount --namespace MyService --
value 2 --timestamp 2016-10-20T12:00:00.000Z
aws cloudwatch put-metric-data --metric-name PageViewCount --namespace MyService --
value 4 --timestamp 2016-10-20T12:00:01.000Z
aws cloudwatch put-metric-data --metric-name PageViewCount --namespace MyService --
value 5 --timestamp 2016-10-20T12:00:02.000Z
```

Si nombra este comando con un nuevo nombre de métrica, CloudWatch crea una métrica automáticamente. De lo contrario, CloudWatch asocia sus datos con la métrica existente que haya especificado.

> **Note**
>
> Cuando se crea una métrica, pueden transcurrir hasta dos minutos antes de poder recuperar las estadísticas de la nueva métrica mediante el comando get-metric-statistics. Sin embargo, pueden transcurrir hasta 15 minutos antes de que la nueva métrica aparezca en la lista de métricas recuperadas mediante el comando list-metrics.

Aunque puede publicar puntos de datos con marcas de tiempo pormenorizados hasta una milésima de segundo, CloudWatch acumula los datos con una granularidad mínima de 1 minuto. CloudWatch registra el promedio (la suma de todos los elementos dividida entre el número de elementos) de los valores recibidos por cada período, así como el número de muestras, el valor máximo y el valor mínimo para el mismo período de tiempo. Por ejemplo, la métrica `PageViewCount` de los ejemplos anteriores contiene tres puntos de datos con marcas temporales separadas solo unos segundos. Si ha establecido el período en 1 minuto, CloudWatch acumula los tres puntos de datos, ya que todos tienen marcas temporales dentro de un periodo de 1 minuto.

Puede utilizar el comando get-metric-statistics para recuperar estadísticas basadas en los puntos de datos que ha publicado.

```
aws cloudwatch get-metric-statistics --namespace MyService --metric-name PageViewCount
 \
--statistics "Sum" "Maximum" "Minimum" "Average" "SampleCount" \
--start-time 2016-10-20T12:00:00.000Z --end-time 2016-10-20T12:05:00.000Z --period 60
```

A continuación, se muestra un ejemplo del resultado.

```
{
    "Datapoints": [
        {
            "SampleCount": 3.0,
            "Timestamp": "2016-10-20T12:00:00Z",
            "Average": 3.6666666666666665,
            "Maximum": 5.0,
            "Minimum": 2.0,
            "Sum": 11.0,
            "Unit": "None"
        }
    ],
    "Label": "PageViewCount"
```

```
}
```

Publicar conjuntos estadísticos

Puede acumular los datos antes de publicar en CloudWatch. Cuando tenga varios puntos de datos por minuto, la acumulación de datos minimiza el número de llamadas a put-metric-data. Por ejemplo, en lugar de llamar a put-metric-data varias veces para los tres puntos de datos que se encuentran separados por tres segundos entre sí, puede acumular los datos en un conjunto de estadísticas que publique con una llamada, utilizando el parámetro `--statistic-values`.

```
aws cloudwatch put-metric-data --metric-name PageViewCount --namespace MyService
 --statistic-values Sum=11,Minimum=2,Maximum=5,SampleCount=3 --
timestamp 2016-10-14T12:00:00.000Z
```

CloudWatch necesita puntos de datos sin procesar para calcular percentiles. Si publica datos a través de un conjunto estadístico en su lugar, no puede recuperar estadísticas de percentil para estos datos, a menos que una de las siguientes condiciones sea cierta:

- El valor de `SampleCount` del conjunto de estadísticas es 1.
- Los valores `Minimum` y `Maximum` del conjunto de estadísticas son iguales.

Publicar el valor cero

Cuando los datos son más esporádicos y tiene períodos que no tienen datos asociados, puede elegir publicar el valor cero (`0`) para dicho periodo o ningún valor en absoluto. Si realiza llamadas periódicas a `PutMetricData` para supervisar el estado de la aplicación, es recomendable publicar cero en lugar de ningún valor. Por ejemplo, puede definir una alarma de CloudWatch que le avise si la aplicación no publica las métricas cada cinco minutos. Quiere que dicha aplicación publique ceros en los períodos sin datos asociados.

También puede publicar ceros si desea realizar un seguimiento del número total de puntos de datos o si desea estadísticas como por ejemplo mínimo y media para incluir puntos de datos con el valor 0.

Dejar de publicar métricas

Para dejar de publicar métricas personalizadas en CloudWatch, cambie el código de la aplicación o del servicio para dejar de usar PutMetricData. CloudWatch no extrae métricas de las aplicaciones,

solo recibe lo que se le envía, por lo que para dejar de publicar sus métricas debe detenerlas en el origen.

Uso de las alarmas de Amazon CloudWatch

Puede crear alarmas de métricas y compuestas en Amazon CloudWatch.

- Una alarma de métrica supervisa una única métrica de CloudWatch o el resultado de una expresión matemática basada en métricas de CloudWatch. La alarma realiza una o varias acciones según el valor de la métrica o expresión con respecto a un umbral durante varios períodos de tiempo. La acción puede ser el envío de una notificación a un tema de Amazon SNS, la ejecución de una acción de Amazon EC2 o una acción de Amazon EC2 Auto Scaling o la creación de un OpsItem o incidente en Systems Manager.
- Una alarma compuesta incluye una expresión de regla que tiene en cuenta los estados de alarma de otras alarmas que haya creado. La alarma compuesta entra en estado ALARM solo si se cumplen todas las condiciones de la regla. Las alarmas especificadas en la expresión de regla de una alarma compuesta pueden incluir alarmas de métricas y otras alarmas compuestas.

  El uso de alarmas compuestas puede reducir el ruido de las alarmas. Puede crear varias alarmas de métricas, así como crear una alarma compuesta y configurar alertas solo para la alarma compuesta. Por ejemplo, una alarma compuesta podría entrar en estado ALARM solo cuando todas las alarmas de métricas subyacentes estén en estado ALARM.

  Las alarmas compuestas pueden enviar notificaciones de Amazon SNS cuando cambian de estado y pueden crear OpsItems de Systems Manager o incidentes cuando entran en estado ALARMA, pero no pueden realizar acciones de EC2 ni acciones de Auto Scaling.

> **Note**
>
> Puede crear todas las alarmas que desee en su cuenta de AWS.

Puede agregar alarmas a los paneles para poder supervisar y recibir alertas sobre los recursos y aplicaciones de AWS en varias regiones. Después de agregar una alarma a un panel, la alarma se vuelve gris cuando está en estado `INSUFFICIENT_DATA` y roja cuando está en el estado `ALARM`. La alarma se muestra sin color cuando está en el estado `OK`.

También puede seleccionar como favoritas las alarmas visitadas recientemente desde la opción Favorites and recents (Favoritos y recientes) en el panel de navegación de la consola de

CloudWatch. La opción Favorites and recents (Favoritos y recientes) tiene columnas para las alarmas favoritas y las alarmas visitadas recientemente.

Una alarma invoca acciones sólo cuando cambia de estado. La excepción es para las alarmas con acciones de Auto Scaling. En el caso de las acciones de Auto Scaling, la alarma sigue invocando la acción una vez por minuto que la alarma permanece en el nuevo estado.

Una alarma puede supervisar una métrica en la misma cuenta. Si ha habilitado la funcionalidad para cuentas cruzadas en la consola de CloudWatch, también puede crear alarmas que supervisen métricas en otras cuentas de AWS. No se admite la creación de alarmas compuestas en cuentas cruzadas. Se admite la creación de alarmas en cuentas cruzadas que utilicen expresiones matemáticas, con la excepción de que las funciones `ANOMALY_DETECTION_BAND`, `INSIGHT_RULE` y `SERVICE_QUOTA` no son compatibles con las alarmas en cuentas cruzadas.

> **Note**
>
> CloudWatch no prueba o valida las acciones que especifique, ni detecta errores de Amazon EC2 Auto Scaling o de Amazon SNS derivados de un intento de invocar acciones inexistentes. Asegúrese de que las acciones de alarma existan.

Estados de las alarmas de métricas

Una alarma de métrica tiene los siguientes estados posibles:

- `OK`: la métrica o expresión está dentro del umbral definido.
- `ALARM`: la métrica o expresión está fuera del umbral definido.
- `INSUFFICIENT_DATA`: la alarma acaba de iniciarse, la métrica no está disponible o no hay suficientes datos disponibles en la métrica para determinar el estado de la alarma.

Evaluación de una alarma

Cuando crea una alarma, especifica tres valores para habilitar CloudWatch a fin de evaluar cuándo se debe cambiar el estado de la alarma:

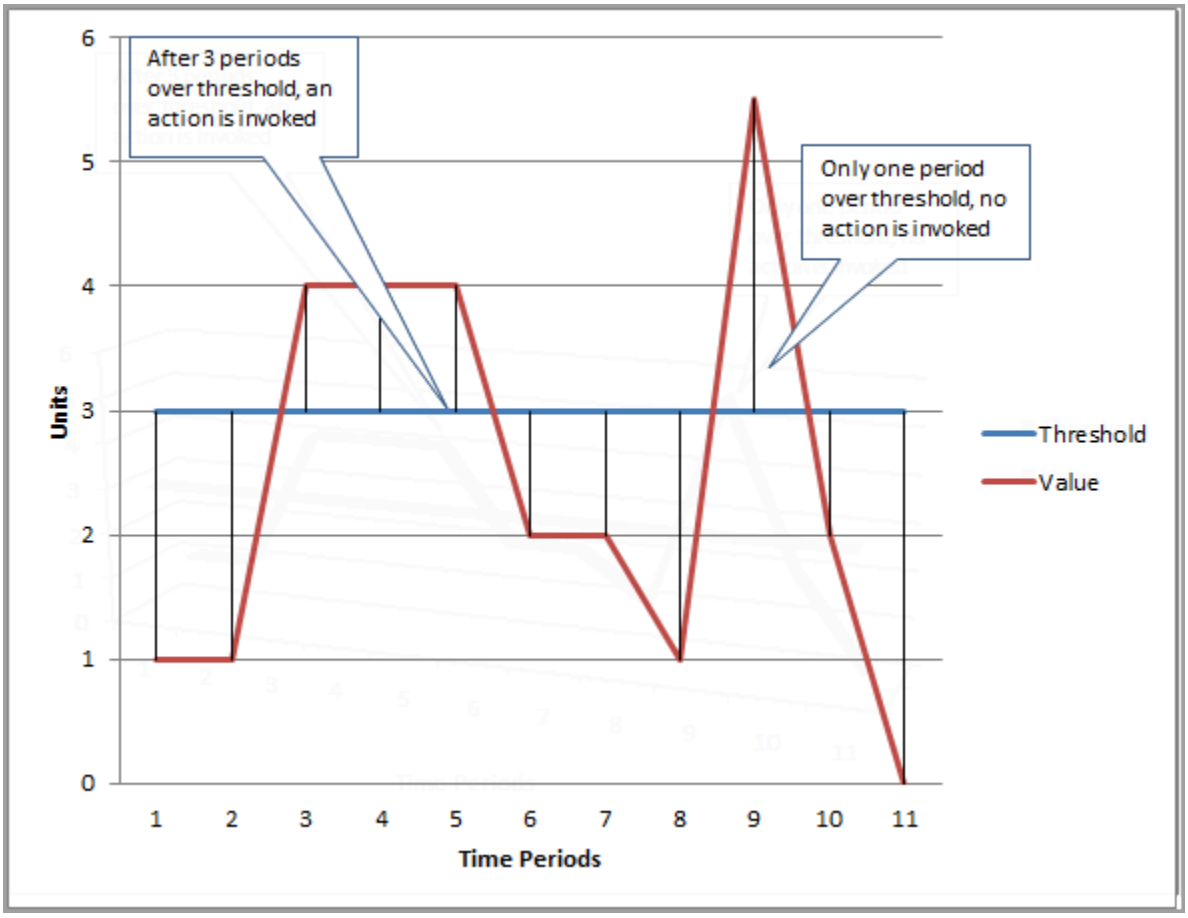
- Período es la duración de tiempo para evaluar la métrica o expresión para crear cada punto de datos individual para una alarma. Se expresa en segundos.

- Evaluation Periods (Período de evaluación) es el número de los periodos más recientes, o de los puntos de datos, para evaluar a la hora de determinar el estado de la alarma.
- Datapoints to Alarm (Puntos de datos para la alarma) es el número de puntos de datos en el periodo de evaluación que deben estar fuera del umbral para provocar que la alarma tenga el estado `ALARM`. No es necesario que los puntos de datos fuera del umbral sean consecutivos, pero todos ellos deben estar dentro de la última cantidad de puntos de datos igual al Evaluation Period (Periodo de evaluación).

Para cualquier período de un minuto o más, se evalúa una alarma cada minuto y la evaluación se basa en el intervalo de tiempo definido por el Período y los Períodos de evaluación. Por ejemplo, si el Período es de 5 minutos (300 segundos) y los Períodos de evaluación son 1, al final del minuto 5, la alarma se evalúa en función de los datos de los minutos 1 a 5. Luego, al final del minuto 6, la alarma se evalúa en función de los datos de los minutos 2 a 6.

Si el período de alarma es de 10 o 30 segundos, la alarma se evalúa cada 10 segundos.

En la siguiente figura, el umbral de alarma de una métrica de alarma está definido en tres unidades. El Evaluation Period (Período de evaluación) y los Datapoints to Alarm (Puntos de datos para la alarma) son 3. Es decir, cuando los puntos de datos existentes en los últimos tres periodos consecutivos superan el umbral, la alarma pasa al estado `ALARM`. En la figura, esto sucede en los periodos de tiempo del tercero al quinto. En el sexto período, el valor cae por debajo del umbral, por lo que uno de los períodos que se evalúa no está fuera del umbral y el estado de la alarma cambia a `OK`. Durante el noveno período de tiempo, el umbral se incumple de nuevo, pero solo para un periodo. Por lo tanto, el estado de la alarma se mantiene `OK`.



Al configurar Evaluation Periods (Períodos de evaluación) y Datapoints to Alarm (Puntos de datos para la alarma) como valores distintos, se establece una alarma 'M de N'. Datapoints to Alarm (Puntos de datos para la alarma) es ('M') y Evaluation Periods (Períodos de evaluación) es ('N'). El intervalo de evaluación es el número de periodos de evaluación multiplicado por la duración del periodo. Por ejemplo, si configura 4 de 5 puntos de datos con un periodo de 1 minuto, el intervalo de evaluación es de 5 minutos. Si configura 3 de 3 puntos de datos con un periodo de 10 minutos, el intervalo de evaluación es de 30 minutos.

> **Note**
>
> Si faltan puntos de datos poco después de crear una alarma y la métrica se estaba notificando a CloudWatch antes de crear la alarma, CloudWatch recupera los puntos de datos más recientes antes de que se creara la alarma a la hora de evaluar la alarma.

Acciones de la alarma

Puede especificar las acciones que realiza una alarma cuando cambia de estado entre los estados OK, ALARM (ALARMA) y INSUFFICIENT_DATA (DATOS INSUFICIENTES).

Se puede configurar la mayoría de las acciones para la transición a cada uno de los tres estados. A excepción de las acciones de escalado automático, las acciones solo se producen en las transiciones de estado y no se vuelven a realizar si la condición persiste durante horas o días. Puede aprovechar el hecho de que se pueden realizar varias acciones para que una alarma envíe un correo electrónico cuando se supere un umbral y, después, otro cuando finalice la condición de incumplimiento. Esto le ayuda a comprobar que sus acciones de escalado o recuperación se activan cuando se espera y funcionan como se desea.

Las siguientes se admiten como acciones de la alarma.

- Notifique a uno o más suscriptores usando un tema de Amazon Simple Notification Service. Los suscriptores pueden ser tanto aplicaciones como personas. Para obtener información completa sobre Amazon SNS, consulte What is Amazon SNS? (¿Qué es Amazon SNS?)
- Invoque una función de Lambda. Esta es la forma más sencilla de automatizar las acciones personalizadas en cambios de estado de alarma.
- Las alarmas basadas en métricas de EC2 también pueden realizar acciones de EC2, como detener, terminar, reiniciar o recuperar una instancia EC2. Para obtener más información, consulte Crear alarmas para detener, terminar, reiniciar o recuperar una instancia EC2.
- Las alarmas pueden realizar acciones para escalar un grupo de escalado automático. Para obtener más información, consulte Step and simple scaling policies for Amazon EC2 Auto Scaling (Pasos y políticas de escalado simples para Amazon EC2 Auto Scaling).
- También puede crear OpsItems en el Centro de operaciones de Systems Manager o crear incidentes en Incident Manager de Systems Manager de AWS. Estas acciones se realizan solo cuando la alarma entra en estado de ALARMA. Para obtener más información, consulte Configuring CloudWatch to create OpsItems from alarms (Configuración de CloudWatch para crear OpsItems a partir de alarmas) y Incident creation (Creación de incidentes).

Acciones de la alarma de Lambda

Las alarmas de CloudWatch garantizan una invocación asíncrona de la función de Lambda para un cambio de estado determinado, excepto en los siguientes casos:

- Cuando la función no existe.
- Cuando CloudWatch no está autorizado a invocar la función de Lambda.

Si CloudWatch no puede acceder al servicio de Lambda o el mensaje se rechaza por otro motivo, CloudWatch lo volverá a intentar hasta que la invocación se realice correctamente. Lambda pone en cola el mensaje y maneja los reintentos de ejecución. Para obtener más información sobre este modelo de ejecución, incluida la información sobre cómo Lambda gestiona los errores, consulte Invocación asíncrona en la Guía para desarrolladores de AWS Lambda.

Puede invocar una función de Lambda en la misma cuenta o en otras cuentas de AWS.

Al especificar una alarma para invocar una función de Lambda como acción de alarma, puede elegir especificar el nombre de la función, el alias de la función o una versión específica de la función.

Al especificar una función de Lambda como acción de la alarma, debe crear una política de recursos para la función para permitir que la entidad principal del servicio de CloudWatch invoque la función.

Una forma de hacerlo es utilizar AWS CLI, como en el siguiente ejemplo:

```
aws lambda add-permission \
--function-name my-function-name \
--statement-id AlarmAction \
--action 'lambda:InvokeFunction' \
--principal lambda.alarms.cloudwatch.amazonaws.com \
--source-account 111122223333 \
--source-arn arn:aws:cloudwatch:us-east-1:111122223333:alarm:alarm-name
```

Como alternativa, puede crear una política similar a uno de los ejemplos siguientes y, a continuación, asignarla a la función.

El siguiente ejemplo especifica la cuenta en la que se encuentra la alarma, de modo que solo las alarmas de esa cuenta (111122223333) pueden invocar la función.

```
{
    "Version": "2012-10-17",
    "Id": "default",
    "Statement": [{
        "Sid": "AlarmAction",
        "Effect": "Allow",
        "Principal": {
            "Service": "lambda.alarms.cloudwatch.amazonaws.com"
```

```
        },
        "Action": "lambda:InvokeFunction",
        "Resource": "arn:aws:lambda:us-east-1:444455556666:function:function-name",
        "Condition": {
            "StringEquals": {
                "AWS:SourceAccount": "111122223333"
            }
        }
    }]
}
```

El siguiente ejemplo tiene un ámbito más limitado, lo que permite que solo la alarma especificada en la cuenta especificada invoque la función.

```
{
  "Version": "2012-10-17",
  "Id": "default",
  "Statement": [
    {
      "Sid": "AlarmAction",
      "Effect": "Allow",
      "Principal": {
        "Service": "lambda.alarms.cloudwatch.amazonaws.com"
      },
      "Action": "lambda:InvokeFunction",
      "Resource": "arn:aws:lambda:us-east-1:444455556666:function:function-name",
      "Condition": {
        "StringEquals": {
          "AWS:SourceAccount": "111122223333",
          "AWS:SourceArn": "arn:aws:cloudwatch:us-east-1:111122223333:alarm:alarm-name"
        }
      }
    }]
}
```

No recomendamos crear una política que no especifique una cuenta de origen, ya que dichas políticas son vulnerables a errores del suplente confuso.

Objeto de evento enviado desde CloudWatch a Lambda

Al configurar una función de Lambda como acción de la alarma, CloudWatch entrega una carga útil de JSON a la función de Lambda cuando la invoca. Esta carga útil de JSON sirve como objeto

de evento para la función. Puede extraer datos de este objeto JSON y utilizarlo en la función. El siguiente ejemplo muestra un objeto de evento de una alarma de métricas.


```
{
  'source': 'aws.cloudwatch',
  'alarmArn': 'arn:aws:cloudwatch:us-east-1:444455556666:alarm:lambda-demo-metric-
alarm',
  'accountId': '444455556666',
  'time': '2023-08-04T12:36:15.490+0000',
  'region': 'us-east-1',
  'alarmData': {
    'alarmName': 'lambda-demo-metric-alarm',
    'state': {
      'value': 'ALARM',
      'reason': 'test',
      'timestamp': '2023-08-04T12:36:15.490+0000'
    },
    'previousState': {
      'value': 'INSUFFICIENT_DATA',
      'reason': 'Insufficient Data: 5 datapoints were unknown.',
      'reasonData':
 '{"version":"1.0","queryDate":"2023-08-04T12:31:29.591+0000","statistic":"Average","period":60
[],"threshold":5.0,"evaluatedDatapoints":[{"timestamp":"2023-08-04T12:30:00.000+0000"},
{"timestamp":"2023-08-04T12:29:00.000+0000"},
{"timestamp":"2023-08-04T12:28:00.000+0000"},
{"timestamp":"2023-08-04T12:27:00.000+0000"},
{"timestamp":"2023-08-04T12:26:00.000+0000"}]}',
      'timestamp': '2023-08-04T12:31:29.595+0000'
    },
    'configuration': {
      'description': 'Metric Alarm to test Lambda actions',
      'metrics': [
        {
          'id': '1234e046-06f0-a3da-9534-EXAMPLEe4c',
          'metricStat': {
            'metric': {
              'namespace': 'AWS/Logs',
              'name': 'CallCount',
              'dimensions': {
                'InstanceId': 'i-12345678'
              }
            },
            'period': 60,
```

```
            'stat': 'Average',
            'unit': 'Percent'
          },
          'returnData': True
        }
      ]
    }
  }
}
```

El siguiente ejemplo muestra un objeto de evento de una alarma compuesta.

```
{
  'source': 'aws.cloudwatch',
  'alarmArn': 'arn:aws:cloudwatch:us-east-1:111122223333:alarm:SuppressionDemo.Main',
  'accountId': '111122223333',
  'time': '2023-08-04T12:56:46.138+0000',
  'region': 'us-east-1',
  'alarmData': {
    'alarmName': 'CompositeDemo.Main',
    'state': {
      'value': 'ALARM',
      'reason': 'arn:aws:cloudwatch:us-
east-1:111122223333:alarm:CompositeDemo.FirstChild transitioned to ALARM at Friday 04
 August, 2023 12:54:46 UTC',
      'reasonData': '{"triggeringAlarms":[{"arn":"arn:aws:cloudwatch:us-
east-1:111122223333:alarm:CompositeDemo.FirstChild","state":
{"value":"ALARM","timestamp":"2023-08-04T12:54:46.138+0000"}}]}',
      'timestamp': '2023-08-04T12:56:46.138+0000'
    },
    'previousState': {
      'value': 'ALARM',
      'reason': 'arn:aws:cloudwatch:us-
east-1:111122223333:alarm:CompositeDemo.FirstChild transitioned to ALARM at Friday 04
 August, 2023 12:54:46 UTC',
      'reasonData': '{"triggeringAlarms":[{"arn":"arn:aws:cloudwatch:us-
east-1:111122223333:alarm:CompositeDemo.FirstChild","state":
{"value":"ALARM","timestamp":"2023-08-04T12:54:46.138+0000"}}]}',
      'timestamp': '2023-08-04T12:54:46.138+0000',
      'actionsSuppressedBy': 'WaitPeriod',
      'actionsSuppressedReason': 'Actions suppressed by WaitPeriod'
    },
    'configuration': {
```

```
      'alarmRule': 'ALARM(CompositeDemo.FirstChild) OR
 ALARM(CompositeDemo.SecondChild)',
      'actionsSuppressor': 'CompositeDemo.ActionsSuppressor',
      'actionsSuppressorWaitPeriod': 120,
      'actionsSuppressorExtensionPeriod': 180
    }
  }
}
```

Configuración de la forma en la que las alarmas de CloudWatch tratan los datos que faltan

A veces, no todos los puntos de datos esperados para una métrica se notifican a CloudWatch. Por ejemplo, esto puede ocurrir cuando se pierde una conexión, un servidor deja de funcionar o cuando una métrica indica datos solo de forma intermitente por diseño.

CloudWatch le permite especificar cómo se tratan los puntos de datos que faltan a la hora de evaluar una alarma. Esto puede ayudarle a configurar la alarma para que pase al estado `ALARM` solo cuando sea adecuado para el tipo de datos que se supervisan. Puede evitar falsos positivos cuando los datos que faltan no indican un problema.

De forma similar al modo en que cada alarma siempre está en uno de los tres estados, cada punto de datos específico que se notifica a CloudWatch entra en una de las tres categorías:

- Sin infracción (dentro del umbral)
- Con infracción (se infringe el umbral)
- Missing (Ausente)

Para cada alarma, puede especificar que CloudWatch trate los puntos de datos que faltan de las siguientes maneras:

- `notBreaching`: los puntos de datos que faltan se tratan como “buenos” y como si estuvieran dentro del límite.
- `breaching`: los puntos de datos que faltan se tratan como ‘malos’ y como si estuvieran fuera del umbral
- `ignore`: se mantiene la alarma actual

- `missing`: si faltan todos los puntos de datos del rango de evaluación de la alarma, la alarma cambia a INSUFFICIENT_DATA.

La mejor opción depende del tipo de métrica y del propósito de la alarma. Por ejemplo, si va a crear una alarma de reversión de una aplicación mediante una métrica que reporta datos de forma continua, puede ser conveniente considerar los puntos de datos faltantes como una infracción, ya que podrían indicar problemas potenciales. Pero para una métrica que genera puntos de datos solo cuando se produce un error, como `ThrottledRequests` en Amazon DynamoDB, es posible que desee tratar los datos que faltan como `notBreaching`. El comportamiento predeterminado es `missing`.

> **Important**
>
> Las alarmas configuradas en las métricas de Amazon EC2 pueden entrar temporalmente en el estado INSUFFICIENT_DATA si faltan puntos de datos de las métricas. Esto es poco frecuente, pero puede ocurrir cuando se interrumpe la generación de informes de métricas, incluso cuando la instancia de Amazon EC2 está en buen estado. En el caso de las alarmas de las métricas de Amazon EC2 que estén configuradas para realizar acciones de detención, finalización, reinicio o recuperación, le recomendamos que configure esas alarmas para tratar los datos faltantes como `missing` y hacer que estas alarmas se activen solo cuando estén en el estado ALARMA.

Elegir la mejor opción para su alarma evita cambios innecesarios y confusos en la condición de alarma y además indica con mayor precisión el estado de su sistema.

> **Important**
>
> Alarmas que evalúan las métricas en el espacio de nombres de `AWS/DynamoDB` siempre ignora los datos que faltan incluso si elige una opción diferente de cómo debe tratar la alarma los datos que faltan. Cuando la métrica `AWS/DynamoDB` tiene datos faltantes, las alarmas que evalúan esa métrica permanecen en su estado actual.

Cómo se evalúa el estado de alarma cuando faltan datos

Cada vez que una alarma evalúa si se debe cambiar de estado, CloudWatch intenta recuperar un número de puntos de datos más elevado del número que se especifica como Evaluation Periods (Períodos de evaluación). El número exacto de puntos de datos que intenta recuperar depende de la duración del periodo de alarma y de si se basa en una métrica con resolución estándar o con alta resolución. El plazo de los puntos de datos que intenta recuperar es el rango de evaluación.

Una vez que CloudWatch recupera estos puntos de datos, ocurre lo siguiente:

- Si no falta ningún punto de datos en el rango de evaluación, CloudWatch evalúa la alarma en función de los puntos de datos que se recopilaron recientemente. El número de puntos de datos evaluados es igual a los Evaluation Periods (Períodos de evaluación) para la alarma. Los puntos de datos adicionales que se encuentran más atrás en el rango de evaluación no son necesarios y no tienen en cuenta.
- Si falta algún punto de datos en el rango de evaluación, pero el número de puntos de datos existentes que se recuperaron exitosamente es igual o superior a los Evaluation Periods (Períodos de evaluación) de la alarma, CloudWatch evalúa el estado de la alarma en función de los puntos de datos existentes más recientes que se han recuperado correctamente, incluidos los puntos de datos necesarios que se encuentran más atrás en el rango de evaluación. En este caso, el valor que establezca acerca de cómo tratar los datos que faltan no es necesario y no se tiene en cuenta.
- Si falta algún punto de datos del rango de evaluación y el número de puntos de datos existente que se recuperaron es inferior al número de Evaluation Periods (Períodos de evaluación) de la alarma, CloudWatch rellena los puntos de datos que faltan con el resultado que ha especificado acerca de cómo se tratan los datos que faltan y, a continuación, evalúa la alarma. Sin embargo, cualquier punto de datos existentes en el rango de evaluación se incluye en la evaluación. CloudWatch utiliza los puntos de datos que faltan solo el menor número de veces posible.

> **Note**
>
> Un caso concreto de este comportamiento es que las alarmas de CloudWatch pueden evaluar una y otra vez el último conjunto de puntos de datos durante un período de tiempo después de que la métrica ha dejado de fluir. Esta reevaluación puede provocar que la alarma cambie de estado y que se vuelvan a ejecutar acciones, si cambió de estado inmediatamente antes de detenerse el flujo de la métrica. Para mitigar este comportamiento, utilice períodos más cortos.

Las tablas siguientes muestran ejemplos del comportamiento de evaluación de alarma. En la primera tabla, Datapoints to Alarm (Puntos de datos para la alarma) y Evaluation Periods (Períodos de evaluación) son ambos 3. CloudWatch recupera los 5 puntos de datos más recientes a la hora de evaluar la alarma, en caso de que falten algunos de los 3 puntos de datos más recientes. 5 es el rango de evaluación para la alarma.

En la columna 1 se muestran los 5 puntos de datos más recientes, ya que el rango de evaluación es 5. Estos puntos de datos se muestran con el punto de datos más reciente a la derecha. 0 es un punto de datos dentro del umbral, X es un punto de datos fuera del umbral y - es un punto de datos que falta.

En la columna 2 se indica cuántos de los tres puntos de datos necesarios faltan. Aunque se evalúan los últimos cinco puntos de datos, solo tres (el valor de Evaluation Periods [Períodos de evaluación]) son necesarios para evaluar el estado de la alarma. El número de puntos de datos de la columna 2 es el número de puntos de datos que deben completarse, utilizando la configuración de cómo se tratan los datos que faltan.

En las columnas 3-6, las cabeceras de columna son los valores posibles para tratar los datos que faltan. Las filas de estas columnas muestran el estado de la alarma que se establece para cada una de estas posibles formas de tratar los datos que faltan.

Puntos de datos	Número de puntos de datos que deben llenarse	MISSING	IGNORE	INFRACCIÓN	SIN INFRACCIÓN
0 - X - X	0	OK	OK	OK	OK
- - - - 0	2	OK	OK	OK	OK
- - - - -	3	INSUFFICIENT_DATA	Mantener el estado actual	ALARM	OK
0 X X - X	0	ALARM	ALARM	ALARM	ALARM
- - X - -	2	ALARM	Mantener el estado actual	ALARM	OK

En la segunda fila de la tabla anterior, la alarma permanece OK incluso si los datos que faltan se tratan como infracción, ya que uno de los puntos de datos existente no se incumple y esto se evalúa junto con los dos puntos de datos que faltan que se tratan como infracción. La próxima vez que esta alarma se evalúe, si siguen faltando los datos, cambiará al estado ALARM, ya que el punto de datos dentro del umbral ya no estará en el rango de evaluación.

La tercera fila, donde faltan los cinco puntos de datos más recientes, ilustra cómo las distintas configuraciones para tratar los datos que faltan afectan el estado de la alarma. Si se considera que los puntos de datos faltantes están fuera del umbral, la alarma entra en el estado ALARM (ALARMA), mientras que si se considera que están dentro del umbral, la alarma cambia al estado OK. Si se ignoran los puntos de datos que faltan, la alarma retiene el estado actual que tenía antes de los puntos de datos que faltan. Y si los puntos de datos faltantes se consideran faltantes, entonces la alarma no tiene suficientes datos reales recientes para hacer una evaluación, y cambia al estado DATOS INSUFICIENTES.

En la cuarta fila, la alarma cambia al estado ALARM en todos los casos porque los tres puntos de datos más recientes están fuera del umbral, y los Evaluation Periods (Períodos de evaluación) de la alarma y Datapoints to Alarm (Puntos de datos a la alarma) se establecen en 3. En este caso, el punto de datos que falta no se tiene en cuenta y no se necesita la configuración para evaluar los datos faltantes, ya que hay tres puntos de datos existentes para evaluar.

La fila número 5 representa un caso especial de evaluación de alarmas llamado estado de alarma prematuro. Para obtener más información, consulte Evitar transiciones prematuras al estado ALARM (ALARMA).

En la tabla siguiente, el Period (Período) se vuelve a establecer en 5 minutos y Datapoints to Alarm (Puntos de datos para alarma) es solo 2 mientras que Evaluation Periods (Períodos de evaluación) es 3. Se trata de una alarma 2 de 3, M de N.

El rango de evaluación es 5. Este es el número máximo de puntos de datos recientes que se recuperan y que se pueden utilizar en caso de que falten algunos.

Puntos de datos	Cantidad de los puntos de datos que faltan	AUSENTE	IGNORE	INFRACCIÓN	SIN INFRACCIÓN
0 - X - X	0	ALARM	ALARM	ALARM	ALARM

Puntos de datos	Cantidad de los puntos de datos que faltan	AUSENTE	IGNORE	INFRACCIÓN	SIN INFRACCIÓN
0 0 X 0 X	0	ALARM	ALARM	ALARM	ALARM
0 - X - -	1	OK	OK	ALARM	OK
- - - - 0	2	OK	OK	ALARM	OK
- - - - X	2	ALARM	Mantener el estado actual	ALARM	OK

En las filas 1 y 2, la alarma siempre pasa al estado ALARM (ALARMA) porque 2 de los 3 puntos de datos más recientes están fuera del umbral. En la fila 2, los dos puntos de datos más antiguos del rango de evaluación no son necesarios porque no falta ninguno de los 3 puntos de datos más recientes, por lo que estos dos puntos de datos antiguos no se tienen en cuenta.

En las filas 3 y 4, la alarma pasa al estado ALARM (ALARMA) solo si los datos que faltan se tratan como si estuvieras fuera del umbral, en cuyo caso los dos puntos de datos faltantes más recientes se tratan como fuera del umbral. En la fila 4, estos dos puntos de datos faltantes, que se tratan como fuera del umbral, proporcionan los dos puntos de datos fuera del umbral necesarios para activar el estado ALARM (ALARMA).

La fila 5 representa un caso especial de evaluación de alarmas llamado estado de alarma prematuro. Para más información, consulte la siguiente sección.

Evitar transiciones prematuras al estado ALARM (ALARMA)

La evaluación de alarmas de CloudWatch incluye lógica para tratar de evitar falsas alarmas, donde la alarma entra en el estado ALARM (ALARMA) prematuramente cuando los datos son intermitentes. El ejemplo que se muestra en la fila 5 de las tablas de la sección anterior ilustra esta lógica. En esas filas, y en los siguientes ejemplos, los Evaluation Periods (Períodos de evaluación) son 3 y el rango de evaluación es de 5 puntos de datos. Los Datapoints to Alarm (Puntos de datos a la alarma) son 3, excepto para el ejemplo M de N, donde los Datapoints to Alarm (Puntos de datos a la alarma) son 2.

Supongamos que los datos más recientes de una alarma son - - - - X, con cuatro puntos de datos que faltan y, a continuación, un punto de datos fuera del umbral como el punto de datos más

reciente. Debido a que el siguiente punto de datos puede estar dentro del umbral, la alarma no entra inmediatamente en estado ALARMA cuando los datos son - - - - X o - - - X - y los Datapoints to Alarm (Puntos de datos a la alarma) son 3. De esta forma, se evitan los falsos positivos cuando el siguiente punto de datos está dentro del umbral y hace que los datos sean - - - X 0 o - - X - 0.

Sin embargo, si los últimos puntos de datos son - - X - -, la alarma entra en estado ALARMA incluso si los puntos de datos que faltan se tratan como faltantes. Esto se debe a que las alarmas están diseñadas para entrar siempre en estado de ALARMA cuando el punto de datos fuera del límite más antiguo disponible durante la cantidad de puntos de datos de los periodos de evaluación es al menos tan antiguo como el valor de los Datapoints to Alarm (Puntos de datos a la alarma), y todos los demás puntos de datos más recientes están fuera del límite o faltan. En este caso, la alarma entra en estado ALARMA incluso si el número total de puntos de datos disponibles es inferior a M (Datapoints to Alarm [Puntos de datos a la alarma]).

Esta lógica de alarma también se aplica a las alarmas M de N. Si el punto de datos más antiguo fuera del umbral durante el intervalo de evaluación es al menos tan antiguo como el valor de los Datapoints to Alarm (Puntos de datos a la alarma), y todos los puntos de datos más recientes están fuera del umbral o faltan, la alarma entra en estado ALARMA sin importar el valor de M (Datapoints to Alarm [Puntos de datos a la alarma]).

Alarmas de alta resolución

Si configura una alarma en una métrica de alta resolución, puede especificar una alarma de alta resolución con un periodo de 10 o 30 segundos, o puede definir una alarma normal con un periodo de cualquier múltiplo de 60 segundos. El cargo es mayor en el caso de las alarmas de alta resolución. Para obtener más información acerca de las métricas de alta resolución, consulte Publicar métricas personalizadas de .

Alarmas en expresiones matemáticas

Puede configurar una alarma basada en el resultado de una expresión matemática que se basa en una o varias métricas de CloudWatch. Una expresión matemática utilizada para una alarma puede incluir hasta 10 métricas. Cada métrica debe utilizar el mismo periodo.

Para una alarma basada en una expresión matemática, puede especificar cómo desea que CloudWatch trate los puntos de datos que faltan. En este caso, se considera que falta el punto de datos si la expresión matemática no devuelve un valor para ese punto de datos.

Las alarmas basadas de expresiones matemáticas no pueden realizar acciones de Amazon EC2.

Para obtener más información acerca de las expresiones matemáticas en métricas y las sintaxis, consulte Uso de la calculadora de métricas.

Muestras de datos reducidos y alarmas de CloudWatch basadas en percentiles

Al establecer un percentil como estadística para una alarma, puede especificar qué es lo que debe hacer cuando no hay suficientes datos para una buena evaluación estadística. Puede elegir que la alarma evalúe la estadística de todas formas y posiblemente cambie el estado de alarma. O bien, puede hacer que la alarma ignore la métrica mientras el tamaño de la muestra sea reducido y esperar a evaluarlo hasta que haya suficientes datos significativos estadísticamente.

Para los percentiles entre 0,5 (incluido) y 1,00 (excluido), esta configuración se utiliza cuando hay menos de 10/(1-percentil) puntos de datos durante el periodo de evaluación. Por ejemplo, esta configuración se utilizaría si se hubiera menos de 1 000 muestras para una alarma en un percentil p99. Para los percentiles entre 0 y 0,5 (excluido), la configuración se utiliza cuando hay menos de 10/percentil puntos de datos.

Características comunes de las alarmas de CloudWatch

Las siguientes características se aplican a todas las alarmas de CloudWatch:

- No existe ningún límite respecto al número de alarmas que se pueden crear. Para crear o actualizar una alarma, utilice la consola de CloudWatch, la acción de la API PutMetricAlarm o el comando put-metric-alarm en la AWS CLI.
- Los nombres de alarma deben contener solo caracteres UTF-8 y no pueden contener caracteres de control ASCII
- Puede enumerar cualquiera o todas las alarmas configuradas actualmente y enumerar las alarmas en un estado determinado mediante la consola de CloudWatch, la acción de la API DescribeAlarms o el comando describe-alarms en la AWS CLI.
- Puede desactivar y habilitar las alarmas mediante las acciones de la API DisableAlarmActions y EnableAlarmActions o los comandos disable-alarm-actions y enable-alarm-actions en la AWS CLI.
- Para probar una alarma, configúrela en cualquier estado mediante la acción de la API SetAlarmState o el comando set-alarm-state en la AWS CLI. Este cambio de estado temporal dura solamente hasta que se produce la siguiente comparación de alarma.

- Puede crear una alarma para una métrica personalizada antes de crear esa métrica personalizada. Para que la alarma sea válida, debe incluir todas las dimensiones para la métrica personalizada, además del espacio de nombres de métrica y nombre de métrica en la definición de alarma. Para ello, puede utilizar la acción de la API PutMetricAlarm o el comando put-metric-alarm en la AWS CLI.
- Puede ver el historial de una alarma mediante la consola de CloudWatch, la acción de la API DescribeAlarmHistory o el comando describe-alarm-history en la AWS CLI. CloudWatch conserva el historial de las alarmas por 30 días. Cada transición de estado se marca con una marca temporal única. En casos excepcionales, el historial podría mostrar más de una notificación para un cambio de estado. La marca temporal le permite confirmar cambios de estado únicos.
- Para seleccionar alarmas como favoritas desde la opción Favorites and recents (Favoritos y recientes) del panel de navegación de la consola de CloudWatch, pase el cursor sobre la alarma que desea agregar como favorita y elija el símbolo estrella junto a ella.
- El número de periodos de evaluación para una alarma multiplicado por la duración de cada periodo de evaluación no puede superar un día.

> **Note**
>
> Algunos recursos de AWS no envían datos de métricas a CloudWatch bajo determinadas condiciones.
> Por ejemplo, Amazon EBS podría no enviar los datos de métricas a un volumen disponible que no se haya adjuntado a una instancia de Amazon EC2, ya que no hay ningún tipo de actividad métrica que supervisar para dicho volumen. Si tiene una alarma establecida para dicha métrica, es posible que observe que su estado cambia a `INSUFFICIENT_DATA`.
> Esto podría indicar que el recurso está inactivo y no significa necesariamente que exista un problema. Puede especificar cómo trata cada alarma los datos ausentes. Para obtener más información, consulte Configuración de la forma en la que las alarmas de CloudWatch tratan los datos que faltan.

Prácticas recomendadas sobre las alarmas para los servicios de AWS

CloudWatch ofrece recomendaciones de alarmas listas para usar. Se trata de alarmas de CloudWatch que le recomendamos que cree para las métricas publicadas por otros servicios

de AWS. Estas recomendaciones pueden ayudarlo a identificar las métricas para las que debe configurar las alarmas de acuerdo con las prácticas recomendadas de supervisión. Las recomendaciones también sugieren los umbrales de alarma que se deberían establecer. Seguir estas recomendaciones puede ayudarle a no descuidar ninguna supervisión importante en su infraestructura de AWS.

Para encontrar las recomendaciones de alarmas, utilice la sección de métricas de la consola de CloudWatch y seleccione el cambio de filtro de las recomendaciones de alarmas. Si navega hasta las alarmas recomendadas en la consola y, a continuación, crea una alarma recomendada, CloudWatch puede completar de forma predeterminada parte de la configuración de la alarma. En el caso de algunas alarmas recomendadas, el valor del umbral de alarma también viene cargado de forma predeterminada. También puede utilizar la consola para descargar definiciones de alarmas basadas en la infraestructura como código para las alarmas recomendadas y, a continuación, usar este código para crear la alarma en AWS CloudFormation, la AWS CLI o Terraform.

También puede ver la lista de alarmas recomendadas en Alarmas recomendadas.

Se le cobrará por las alarmas que cree al mismo precio que por cualquier otra alarma que cree en CloudWatch. El uso de las recomendaciones no genera cargos adicionales. Para obtener más información, consulte Precios de Amazon CloudWatch.

Buscar y crear las alarmas recomendadas

Siga estos pasos para buscar las métricas para las que CloudWatch recomienda configurar alarmas y, si lo desea, para crear una de estas alarmas. El primer procedimiento explica cómo encontrar las métricas que tienen alarmas recomendadas y cómo crear una de estas.

También puede descargar un lote de las definiciones de alarmas basadas en la infraestructura como código para todas las alarmas recomendadas en un espacio de nombres de AWS, como `AWS/Lambda` o `AWS/S3`. Esas instrucciones se encuentran más adelante en este tema.

Para encontrar las métricas con las alarmas recomendadas y crear una única alarma recomendada

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, seleccione Métricas y, a continuación, Todas las métricas.
3. Sobre la tabla Métricas, elija Recomendaciones de alarmas.

   La lista de espacios de nombres de métricas se filtra para incluir solo las métricas que tienen recomendaciones de alarmas y que publican los servicios de su cuenta.

4. Elija el espacio de nombres para un servicio.

   La lista de métricas de este espacio de nombres se filtra para incluir solo las que tienen recomendaciones de alarmas.

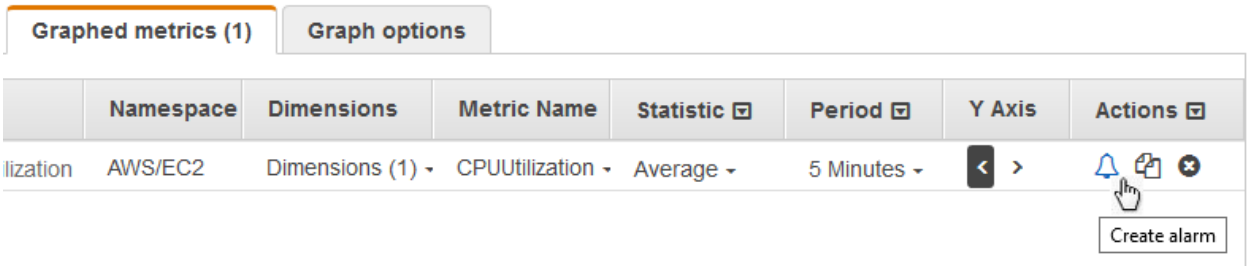
5. Elija Ver detalles para ver la finalidad de la alarma y el umbral recomendado para una métrica.

6. Para crear una alarma para una de las métricas, realice alguna de las siguientes acciones:

   - Para utilizar la consola para crear la alarma, haga lo siguiente:

     a. Seleccione la casilla de verificación de la métrica y elija la pestaña Métricas diagramadas.

     b. Elija el ícono de la alarma.



     Aparecerá el asistente de creación de alarmas, con el nombre de la métrica, la estadística y el período completados de forma predeterminada según la recomendación de la alarma. Si la recomendación incluye un valor de umbral específico, ese valor también se rellena de forma predeterminada.

     c. Elija Siguiente.

     d. En Notificación, seleccione el tema de SNS al que desee enviar la notificación cuando la alarma tenga una transición al estado `ALARM`, `OK` o `INSUFFICIENT_DATA`.

     Para que la alarma envíe varias notificaciones para el mismo estado de alarma o para estados de alarma diferentes, seleccione Add notificación (Añadir notificación).

     Para que la alarma no envíe notificaciones, elija Remove (Eliminar).

     e. Para que la alarma realice el Auto Scaling o acciones de EC2, elija el botón correspondiente, y elija el estado de alarma y la acción que se debe realizar.

     f. Cuando haya terminado, elija Next (Siguiente).

     g. Escriba un nombre y la descripción de la alarma. El nombre solo debe contener caracteres ASCII. A continuación, elija Next.

h. En Obtener vista previa y crear, confirme que la información y las condiciones son las que desea y, a continuación, elija Crear alarma.

- Para descargar una definición de alarma basada en la infraestructura como código para usarla en AWS CloudFormation, en la AWS CLI o en Terraform, elija Descargar el código de alarma y seleccione el formato que desee. El código descargado tendrá la configuración recomendada para el nombre, la estadística y el umbral de la métrica.

Para descargar las definiciones de alarma basada en la infraestructura como código para todas las alarmas recomendadas para un servicio de AWS

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, seleccione Métricas y, a continuación, Todas las métricas.
3. Sobre la tabla Métricas, elija Recomendaciones de alarmas.

   La lista de espacios de nombres de métricas se filtra para incluir solo las métricas que tienen recomendaciones de alarmas y que publican los servicios de su cuenta.
4. Elija el espacio de nombres para un servicio.

   La lista de métricas de este espacio de nombres se filtra para incluir solo las que tienen recomendaciones de alarmas.
5. En Descargar el código de alarma se muestra cuántas alarmas se recomiendan para las métricas de este espacio de nombres. Para descargar las definiciones de alarmas basadas en la infraestructura como código para todas las alarmas recomendadas, elija Descargar el código de alarma y, a continuación, elija el formato de código que desee.

Alarmas recomendadas

En las siguientes secciones se enumeran las métricas para las que sugerimos configurar las alarmas de las prácticas recomendadas. Para cada métrica, también se muestran las dimensiones, la finalidad de la alarma, el umbral recomendado, la justificación del umbral y el período y el número de puntos de datos.

Es posible que algunas métricas aparezcan dos veces en la lista. Esto sucede cuando se recomiendan diferentes alarmas para diferentes combinaciones de las dimensiones de esa métrica.

Los Puntos de datos para la alarma son el número de puntos de datos que se deben infringir para que la alarma pase al estado de ALARMA. Los Períodos de evaluación son el número de períodos

que se tienen en cuenta al evaluar la alarma. Si estos números son los mismos, la alarma pasará al estado de ALARMA solo cuando ese número de períodos consecutivos tenga valores que superen el umbral. Si los Puntos de datos para la alarma son inferiores a los de los Períodos de evaluación, se trata de una alarma de tipo “M de N” y la alarma pasa al estado de ALARMA si al menos los puntos de datos de los Puntos de datos para la alarma incumplen con cualquier conjunto de puntos de datos de los Períodos de evaluación. Para obtener más información, consulte Evaluación de una alarma.

Temas

- Amazon API Gateway
- Amazon EC2 Auto Scaling
- Amazon CloudFront
- Amazon Cognito
- Amazon DynamoDB
- Amazon EBS
- Amazon EC2
- Amazon ElastiCache
- Amazon EC2 (AWS/ElasticGPUs)
- Amazon ECS
- Amazon ECS con Información de contenedores
- Amazon EFS
- Amazon EKS con Información de contenedores
- Amazon Kinesis Data Streams
- Lambda
- Lambda Insights
- Amazon VPC (AWS/NATGateway)
- Enlace privado de AWS (AWS/PrivateLinkEndpoints)
- Enlace privado de AWS (AWS/PrivateLinkServices)
- Amazon RDS
- Amazon Route 53 Public Data Plane
- Amazon S3
- S3ObjectLambda
- Amazon SNS

- Amazon SQS
- AWS VPN

Amazon API Gateway

4XXError

Dimensiones: ApiName, Stage

Descripción de la alarma: esta alarma detecta una tasa elevada de errores del lado del cliente. Esto puede indicar un problema en los parámetros de autorización o de la solicitud del cliente. También, puede significar que se ha eliminado un recurso o que un cliente solicita uno que no existe. Considere la posibilidad de habilitar los Registros de CloudWatch y comprobar si hay algún error que pueda causar los errores 4XX. Además, considere la posibilidad de habilitar las métricas detalladas de CloudWatch para ver esta métrica por recurso y método y, así, reducir la búsqueda del origen de los errores. Los errores también pueden deberse a que se supera la limitación configurada. Si las respuestas y los registros indican tasas altas e inesperadas de errores 429, siga esta guía para solucionar el problema.

Finalidad: esta alarma puede detectar altas tasas de errores del lado del cliente en las solicitudes de la puerta de enlace de la API.

Estadística: Average

Umbral recomendado: 0,05

Justificación del umbral: el umbral sugerido detecta cuándo más del 5 % del total de las solicitudes reciben errores 4XX. Sin embargo, puede ajustar el umbral para adaptarlo al tráfico de las solicitudes y a las tasas de error aceptables. También puede analizar los datos históricos para determinar la tasa de error aceptable para la carga de trabajo de la aplicación y, a continuación, ajustar el umbral en consecuencia. Los errores 4XX que se producen con frecuencia deben ser objeto de alarma. Sin embargo, si se establece un valor muy bajo para el umbral, la alarma puede ser demasiado sensible.

Período: 60

Puntos de datos para la alarma: 5

Períodos de evaluación: 5

Operador de comparación: GREATER_THAN_THRESHOLD

5XXError

Dimensiones: ApiName, Stage

Descripción de la alarma: esta alarma ayuda a detectar una alta tasa de errores del lado del servidor. Esto puede indicar que hay algún problema en el backend de la API, en la red o en la integración entre la puerta de enlace de la API y la API del backend. Esta documentación puede ayudarlo a solucionar la causa de los errores 5XX.

Finalidad: esta alarma puede detectar altas tasas de errores del lado del servidor en las solicitudes de la puerta de enlace de la API.

Estadística: Average

Umbral recomendado: 0,05

Justificación del umbral: el umbral sugerido detecta cuándo más del 5 % del total de las solicitudes reciben errores 5XX. Sin embargo, puede ajustar el umbral para adaptarlo al tráfico de las solicitudes y a las tasas de error aceptables. También, puede analizar los datos históricos para determinar la tasa de error aceptable para la carga de trabajo de la aplicación y, a continuación, ajustar el umbral en consecuencia. Los errores 5XX que se producen con frecuencia deben ser objeto de alarma. Sin embargo, si se establece un valor muy bajo para el umbral, la alarma puede ser demasiado sensible.

Período: 60

Puntos de datos para la alarma: 3

Períodos de evaluación: 3

Operador de comparación: GREATER_THAN_THRESHOLD

Recuento

Dimensiones: ApiName, Stage

Descripción de la alarma: esta alarma ayuda a detectar un volumen de tráfico bajo en la etapa de la API de REST. Esto puede ser un indicador de un problema con la aplicación que llama a la API, como el uso de puntos de conexión incorrectos. También, podría indicar un problema con la configuración o los permisos de la API, lo que hace que los clientes no puedan acceder a ella.

Finalidad: esta alarma puede detectar un volumen de tráfico bajo imprevisto en la fase de la API de REST. Recomendamos que cree esta alarma si la API recibe un número predecible y constante de solicitudes en condiciones normales. Si tiene habilitadas las métricas detalladas de CloudWatch y puede predecir el volumen de tráfico normal por método y recurso, le recomendamos que cree alarmas alternativas para tener una supervisión más detallada de las caídas en el volumen de tráfico para cada recurso y método. Esta alarma no se recomienda para las API que no esperan un tráfico constante y uniforme.

Estadística: SampleCount

Umbral recomendado: depende de la situación

Justificación del umbral: establezca el umbral en función del análisis de datos históricos para determinar cuál es el recuento de solicitudes de referencia esperado para la API. Si se establece el umbral en un valor muy alto, es posible que la alarma sea demasiado sensible en los períodos de tráfico bajo normal y esperado. Por el contrario, si se establece en un valor muy bajo, la alarma podría pasar por alto descensos anómalos más pequeños en el volumen del tráfico.

Período: 60

Puntos de datos para la alarma: 10

Períodos de evaluación: 10

Operador de comparación: LESS_THAN_THRESHOLD

Recuento

Dimensiones: ApiName, Stage, Resource, Method

Descripción de la alarma: esta alarma ayuda a detectar un volumen de tráfico bajo para el recurso y el método de la API de REST en la etapa. Esto puede indicar un problema con la aplicación que llama a la API, como el uso de puntos de conexión incorrectos. También, podría indicar un problema con la configuración o los permisos de la API, lo que hace que los clientes no puedan acceder a ella.

Finalidad: esta alarma puede detectar un volumen de tráfico bajo imprevisto para el recurso y el método de la API de REST en la etapa. Recomendamos que cree esta alarma si la API recibe un número predecible y constante de solicitudes en condiciones normales. Esta alarma no se recomienda para las API que no esperan un tráfico constante y uniforme.

Estadística: SampleCount

Umbral recomendado: depende de la situación

Justificación del umbral: establezca el umbral en función del análisis de datos históricos para determinar cuál es el recuento de solicitudes de referencia esperado para la API. Si se establece el umbral en un valor muy alto, es posible que la alarma sea demasiado sensible en los períodos de tráfico bajo normal y esperado. Por el contrario, si se establece en un valor muy bajo, la alarma podría pasar por alto descensos anómalos más pequeños en el volumen del tráfico.

Período: 60

Puntos de datos para la alarma: 10

Períodos de evaluación: 10

Operador de comparación: LESS_THAN_THRESHOLD

Recuento

Dimensiones: ApiId, Stage

Descripción de la alarma: esta alarma ayuda a detectar un volumen de tráfico bajo en la etapa de la API HTTP. Esto puede indicar un problema con la aplicación que llama a la API, como el uso de puntos de conexión incorrectos. También, podría indicar un problema con la configuración o los permisos de la API, lo que hace que los clientes no puedan acceder a ella.

Finalidad: esta alarma puede detectar un volumen de tráfico bajo imprevisto en la fase de la API HTTP. Recomendamos que cree esta alarma si la API recibe un número predecible y constante de solicitudes en condiciones normales. Si tiene habilitadas las métricas detalladas de CloudWatch y puede predecir el volumen de tráfico normal por ruta, le recomendamos que cree alarmas alternativas a esta función para poder supervisar de forma más detallada las caídas en el volumen de tráfico de cada ruta. Esta alarma no se recomienda para las API que no esperan un tráfico constante y uniforme.

Estadística: SampleCount

Umbral recomendado: depende de la situación

Justificación del umbral: establezca el valor del umbral en función del análisis de datos históricos para determinar cuál es el recuento de solicitudes de referencia esperado para la API. Si se establece el umbral en un valor muy alto, es posible que la alarma sea demasiado sensible en los

períodos de tráfico bajo normal y esperado. Por el contrario, si se establece en un valor muy bajo, la alarma podría pasar por alto descensos anómalos más pequeños en el volumen del tráfico.

Período: 60

Puntos de datos para la alarma: 10

Períodos de evaluación: 10

Operador de comparación: LESS_THAN_THRESHOLD

Recuento

Dimensiones: ApiId, Stage, Resource, Method

Descripción de la alarma: esta alarma ayuda a detectar un bajo volumen de tráfico para la ruta de la API HTTP en la etapa. Esto puede indicar un problema con la aplicación que llama a la API, como el uso de puntos de conexión incorrectos. También, podría indicar un problema con la configuración o los permisos de la API, lo que hace que los clientes no puedan acceder a ella.

Finalidad: esta alarma puede detectar un volumen de tráfico bajo imprevisto en la ruta de la API HTTP en la etapa. Recomendamos que cree esta alarma si la API recibe un número predecible y constante de solicitudes en condiciones normales. Esta alarma no se recomienda para las API que no esperan un tráfico constante y uniforme.

Estadística: SampleCount

Umbral recomendado: depende de la situación

Justificación del umbral: establezca el valor del umbral en función del análisis de datos históricos para determinar cuál es el recuento de solicitudes de referencia esperado para la API. Si se establece el umbral en un valor muy alto, es posible que la alarma sea demasiado sensible en los períodos de tráfico bajo normal y esperado. Por el contrario, si se establece en un valor muy bajo, la alarma podría pasar por alto descensos anómalos más pequeños en el volumen del tráfico.

Período: 60

Puntos de datos para la alarma: 10

Períodos de evaluación: 10

Operador de comparación: LESS_THAN_THRESHOLD

IntegrationLatency

Dimensiones: ApiId, Stage

Descripción de la alarma: esta alarma ayuda a detectar si hay una latencia de integración alta para las solicitudes de API en una etapa. Puede correlacionar el valor de la métrica `IntegrationLatency` con la métrica de la latencia correspondiente de su backend, como la métrica `Duration` de las integraciones de Lambda. Esto le ayuda a determinar si el backend de la API tarda más tiempo en procesar las solicitudes de los clientes debido a problemas de rendimiento o si hay algún otro tipo de sobrecarga debido a la inicialización o al arranque en frío. Además, considere la posibilidad de habilitar los Registros de CloudWatch para su API y comprobar los registros para detectar cualquier error que pueda causar los problemas de latencia elevada. Asimismo, considere la posibilidad de habilitar las métricas detalladas de CloudWatch para obtener una vista de esta métrica por ruta, lo que le ayudará a ubicar el origen de la latencia de la integración.

Finalidad: esta alarma puede detectar cuándo las solicitudes de la puerta de enlace de la API en una etapa tienen una latencia de integración alta. Recomendamos esta alarma para las API de WebSocket y la consideramos opcional para las API HTTP, porque estas ya tienen recomendaciones de alarma independientes para la métrica de latencia. Si tiene habilitadas las métricas detalladas de CloudWatch y tiene diferentes requisitos de rendimiento de latencia de integración por ruta, le recomendamos que cree alarmas alternativas para tener una supervisión más detallada de la latencia de integración de cada ruta.

Estadística: p90

Umbral recomendado: 2000,0

Justificación del umbral: el valor de umbral sugerido no funciona para todas las cargas de trabajo de la API. Sin embargo, puede utilizarse como punto de partida para el umbral. A continuación, puede elegir diferentes valores de umbral en función de la carga de trabajo y de los requisitos aceptables de latencia, rendimiento y SLA para la API. Si es aceptable que la API tenga una latencia más alta en general, establezca un valor de umbral más alto para que la alarma sea menos sensible. Sin embargo, si se espera que la API proporcione respuestas casi en tiempo real, establezca un valor de umbral más bajo. También, puede analizar los datos históricos para determinar la latencia de referencia esperada para la carga de trabajo de la aplicación y, a continuación, utilizarlos para ajustar el valor del umbral en consecuencia.

Período: 60

Puntos de datos para la alarma: 5

Períodos de evaluación: 5

Operador de comparación: GREATER_THAN_OR_EQUAL_TO_THRESHOLD

IntegrationLatency

Dimensiones: ApiId, Stage, Route

Descripción de la alarma: esta alarma ayuda a detectar si hay una latencia de integración alta para las solicitudes de la API de WebSocket para una ruta en una etapa. Puede correlacionar el valor de la métrica `IntegrationLatency` con la métrica de la latencia correspondiente de su backend, como la métrica `Duration` de las integraciones de Lambda. Esto le ayuda a determinar si el backend de la API tarda más tiempo en procesar las solicitudes de los clientes debido a problemas de rendimiento o si hay algún otro tipo de sobrecarga debido a la inicialización o al arranque en frío. Además, considere la posibilidad de habilitar los Registros de CloudWatch para su API y comprobar los registros para detectar cualquier error que pueda causar los problemas de latencia elevada.

Finalidad: esta alarma puede detectar cuándo las solicitudes de la puerta de enlace de la API para una ruta en una etapa tienen una latencia de integración alta.

Estadística: p90

Umbral recomendado: 2000,0

Justificación del umbral: el valor de umbral sugerido no funciona para todas las cargas de trabajo de la API. Sin embargo, puede utilizarse como punto de partida para el umbral. A continuación, puede elegir diferentes valores de umbral en función de la carga de trabajo y de los requisitos aceptables de latencia, rendimiento y SLA para la API. Si es aceptable que la API tenga una latencia más alta en general, puede establecer un valor de umbral más alto para que la alarma sea menos sensible. Sin embargo, si se espera que la API proporcione respuestas casi en tiempo real, establezca un valor de umbral más bajo. También, puede analizar los datos históricos para determinar la latencia de referencia esperada para la carga de trabajo de la aplicación y, a continuación, utilizarlos para ajustar el valor del umbral en consecuencia.

Período: 60

Puntos de datos para la alarma: 5

Períodos de evaluación: 5

Operador de comparación: GREATER_THAN_OR_EQUAL_TO_THRESHOLD

Latency (Latencia)

Dimensiones: ApiName, Stage

Descripción de la alarma: esta alarma detecta una latencia elevada en una etapa. Encuentre el valor de la métrica `IntegrationLatency` para comprobar la latencia del backend de la API. Si las dos métricas están casi alineadas, el backend de la API es el origen de la latencia más alta, por lo que debería investigar si hay algún problema. Considere también la posibilidad de habilitar los Registros de CloudWatch y comprobar si hay errores que puedan causar la latencia elevada. Además, considere la posibilidad de habilitar métricas detalladas de CloudWatch para ver esta métrica por recurso y método y reducir la búsqueda del origen de la latencia. Si corresponde, consulte las guías ¿Cómo soluciono los problemas de latencia alta en mis solicitudes de API Gateway integradas con Lambda? o ¿Cómo puedo solucionar los problemas de latencia del punto de conexión de mi API de API Gateway optimizada en la periferia?.

Finalidad: esta alarma puede detectar cuándo las solicitudes de la puerta de enlace de la API en una etapa tienen una latencia elevada. Si tiene habilitadas las métricas detalladas de CloudWatch y tiene requisitos de rendimiento de latencia diferentes para cada método y recurso, le recomendamos que cree alarmas alternativas para tener una supervisión más detallada de la latencia de cada recurso y método.

Estadística: p90

Umbral recomendado: 2500,0

Justificación del umbral: el valor de umbral sugerido no funciona para todas las cargas de trabajo de la API. Sin embargo, puede utilizarse como punto de partida para el umbral. A continuación, puede elegir diferentes valores de umbral en función de la carga de trabajo y de los requisitos aceptables de latencia, rendimiento y SLA para la API. Si es aceptable que la API tenga una latencia más alta en general, puede establecer un valor de umbral más alto para que la alarma sea menos sensible. Sin embargo, si se espera que la API proporcione respuestas casi en tiempo real, establezca un valor de umbral más bajo. También, puede analizar los datos históricos para determinar cuál es la latencia de referencia esperada para la carga de trabajo de la aplicación y, a continuación, ajustar el valor del umbral en consecuencia.

Período: 60

Puntos de datos para la alarma: 5

Períodos de evaluación: 5

Operador de comparación: GREATER_THAN_OR_EQUAL_TO_THRESHOLD

Latency (Latencia)

Dimensiones: ApiName, Stage, Resource, Method

Descripción de la alarma: esta alarma detecta una latencia elevada para un recurso y un método en una etapa. Encuentre el valor de la métrica `IntegrationLatency` para comprobar la latencia del backend de la API. Si las dos métricas están casi alineadas, el backend de la API es el origen de la latencia más alta, por lo que debería investigarlo para detectar problemas de rendimiento. Considere también la posibilidad de habilitar los Registros de CloudWatch y comprobar si hay algún error que pueda causar la latencia elevada. También, puede consultar las guías de ¿Cómo soluciono los problemas de latencia alta en mis solicitudes de API Gateway integradas con Lambda? o de ¿Cómo puedo solucionar los problemas de latencia del punto de conexión de mi API de API Gateway optimizada en la periferia?, si es pertinente.

Finalidad: esta alarma puede detectar cuándo las solicitudes de la puerta de enlace de la API de un recurso y un método en una etapa tienen una latencia elevada.

Estadística: p90

Umbral recomendado: 2500,0

Justificación del umbral: el valor de umbral sugerido no funciona para todas las cargas de trabajo de la API. Sin embargo, puede utilizarse como punto de partida para el umbral. A continuación, puede elegir diferentes valores de umbral en función de la carga de trabajo y de los requisitos aceptables de latencia, rendimiento y SLA para la API. Si es aceptable que la API tenga una latencia más alta en general, puede establecer un valor de umbral más alto para que la alarma sea menos sensible. Sin embargo, si se espera que la API proporcione respuestas casi en tiempo real, establezca un valor de umbral más bajo. También, puede analizar los datos históricos para determinar la latencia de referencia esperada para la carga de trabajo de la aplicación y, a continuación, ajustar el valor del umbral en consecuencia.

Período: 60

Puntos de datos para la alarma: 5

Períodos de evaluación: 5

Operador de comparación: GREATER_THAN_OR_EQUAL_TO_THRESHOLD

Latency (Latencia)

Dimensiones: ApiId, Stage

Descripción de la alarma: esta alarma detecta una latencia elevada en una etapa. Encuentre el valor de la métrica `IntegrationLatency` para comprobar la latencia del backend de la API. Si las dos métricas están casi alineadas, el backend de la API es el origen de la latencia más alta, por lo que debería investigarlo para detectar problemas de rendimiento. Considere también la posibilidad de habilitar los Registros de CloudWatch y comprobar si hay algún error que pueda causar la latencia elevada. Además, considere la posibilidad de habilitar las métricas detalladas de CloudWatch para ver esta métrica por ruta y reducir la búsqueda del origen de la latencia. También, puede consultar ¿Cómo soluciono los problemas de latencia alta en mis solicitudes de API Gateway integradas con Lambda?, si corresponde.

Finalidad: esta alarma puede detectar cuándo las solicitudes de la puerta de enlace de la API en una etapa tienen una latencia elevada. Si tiene habilitadas las métricas detalladas de CloudWatch y tiene diferentes requisitos de rendimiento de latencia por ruta, le recomendamos que cree alarmas alternativas para tener una supervisión más detallada de la latencia de cada ruta.

Estadística: p90

Umbral recomendado: 2500,0

Justificación del umbral: el valor de umbral sugerido no funciona para todas las cargas de trabajo de la API. Sin embargo, se puede utilizar como punto de partida para el umbral. Entonces, puede elegir diferentes valores de umbral en función de la carga de trabajo y de los requisitos de latencia aceptable, de rendimiento y de SLA para la API. Si es aceptable que la API tenga una latencia más alta en general, puede establecer un valor de umbral más alto para que sea menos sensible. Sin embargo, si se espera que la API proporcione respuestas casi en tiempo real, establezca un valor de umbral más bajo. También, puede analizar los datos históricos para determinar la latencia de referencia esperada para la carga de trabajo de la aplicación y, a continuación, ajustar el valor del umbral en consecuencia.

Período: 60

Puntos de datos para la alarma: 5

Períodos de evaluación: 5

Operador de comparación: GREATER_THAN_OR_EQUAL_TO_THRESHOLD

Latency (Latencia)

Dimensiones: ApiId, Stage, Resource, Method

Descripción de la alarma: esta alarma detecta una latencia elevada para una ruta en una etapa. Encuentre el valor de la métrica `IntegrationLatency` para comprobar la latencia del backend de la API. Si las dos métricas están casi alineadas, el backend de la API es la fuente de mayor latencia y debe investigarse para detectar problemas de rendimiento. Considere también la posibilidad de habilitar los registros de CloudWatch y comprobar si hay algún error que pueda causar la latencia elevada. También, puede consultar ¿Cómo soluciono los problemas de latencia alta en mis solicitudes de API Gateway integradas con Lambda?, si corresponde.

Finalidad: esta alarma se usa para detectar cuándo las solicitudes de la puerta de enlace de la API para una ruta en una etapa tienen una latencia elevada.

Estadística: p90

Umbral recomendado: 2500,0

Justificación del umbral: el valor de umbral sugerido no funciona para todas las cargas de trabajo de la API. Sin embargo, se puede utilizar como punto de partida para el umbral. A continuación, puede elegir diferentes valores de umbral en función de la carga de trabajo y de los requisitos aceptables de latencia, rendimiento y SLA para la API. Si es aceptable que la API tenga una latencia más alta en general, puede establecer un valor de umbral más alto para que la alarma sea menos sensible. Sin embargo, si se espera que la API proporcione respuestas casi en tiempo real, establezca un valor de umbral más bajo. También, puede analizar los datos históricos para determinar la latencia de referencia esperada para la carga de trabajo de la aplicación y, a continuación, ajustar el valor del umbral en consecuencia.

Período: 60

Puntos de datos para la alarma: 5

Períodos de evaluación: 5

Operador de comparación: GREATER_THAN_OR_EQUAL_TO_THRESHOLD

4XX

Dimensiones: ApiId, Stage

Descripción de la alarma: esta alarma detecta una tasa elevada de errores del lado del cliente. Esto puede indicar un problema en los parámetros de autorización o de la solicitud del cliente. También podría significar que se eliminó una ruta o que un cliente solicita una que no existe en la API. Considere la posibilidad de habilitar los Registros de CloudWatch y comprobar si hay algún error que pueda causar los errores 4XX. Además, considere la posibilidad de habilitar métricas detalladas de CloudWatch para ver estas métricas por ruta, lo que le ayudará a reducir la búsqueda del origen de los errores. También se pueden producir errores si se supera el valor de limitación configurado. Si las respuestas y los registros indican tasas altas e inesperadas de errores 429, siga esta guía para solucionar el problema.

Finalidad: esta alarma puede detectar altas tasas de errores del lado del cliente en las solicitudes de la puerta de enlace de la API.

Estadística: Average

Umbral recomendado: 0,05

Justificación del umbral: el umbral sugerido detecta cuándo más del 5 % del total de las solicitudes reciben errores 4XX. Sin embargo, puede ajustar el umbral para adaptarlo al tráfico de las solicitudes y a las tasas de error aceptables. También puede analizar los datos históricos para determinar la tasa de error aceptable para la carga de trabajo de la aplicación y, a continuación, ajustar el umbral en consecuencia. Los errores 4XX que se producen con frecuencia deben ser objeto de alarma. Sin embargo, si se establece un valor muy bajo para el umbral, la alarma puede ser demasiado sensible.

Período: 60

Puntos de datos para la alarma: 5

Períodos de evaluación: 5

Operador de comparación: GREATER_THAN_THRESHOLD

5xx

Dimensiones: ApiId, Stage

Descripción de la alarma: esta alarma ayuda a detectar una alta tasa de errores del lado del servidor. Esto puede indicar que hay algún problema en el backend de la API, en la red o en la integración entre la puerta de enlace de la API y la API del backend. Esta documentación puede ayudarle a solucionar la causa de los errores 5XX.

Finalidad: esta alarma puede detectar altas tasas de errores del lado del servidor en las solicitudes de la puerta de enlace de la API.

Estadística: Average

Umbral recomendado: 0,05

Justificación del umbral: el umbral sugerido detecta cuándo más del 5 % del total de las solicitudes reciben errores 5XX. Sin embargo, puede ajustar el umbral para adaptarlo al tráfico de las solicitudes y a las tasas de error aceptables. También, puede analizar los datos históricos para determinar cuál es la tasa de error aceptable para la carga de trabajo de la aplicación y, a continuación, ajustar el umbral en consecuencia. Los errores 5XX que se producen con frecuencia deben ser objeto de alarma. Sin embargo, si se establece un valor muy bajo para el umbral, la alarma puede ser demasiado sensible.

Período: 60

Puntos de datos para la alarma: 3

Períodos de evaluación: 3

Operador de comparación: GREATER_THAN_THRESHOLD

MessageCount

Dimensiones: ApiId, Stage

Descripción de la alarma: esta alarma ayuda a detectar un volumen de tráfico bajo para la etapa de la API de WebSocket. Esto puede indicar un problema cuando los clientes llaman a la API, como el uso de puntos de conexión incorrectos o problemas con el backend al enviar mensajes a los clientes. También, podría indicar un problema con la configuración o los permisos de la API, lo que haría que los clientes no puedan acceder a ella.

Finalidad: esta alarma puede detectar un volumen de tráfico bajo imprevisto para la etapa de la API de WebSocket. Recomendamos que cree esta alarma si la API recibe y envía un número predecible y constante de mensajes en condiciones normales. Si tiene habilitadas las métricas detalladas de CloudWatch y puede predecir el volumen de tráfico normal por ruta, es mejor crear alarmas alternativas a esta, a fin de tener una supervisión más detallada de las caídas en el volumen de tráfico en cada ruta. No recomendamos esta alarma para las API que no esperan un tráfico constante y uniforme.

Estadística: SampleCount

Umbral recomendado: depende de la situación

Justificación del umbral: establezca el valor del umbral en función del análisis de datos históricos para determinar cuál es el recuento de mensajes de referencia esperado para la API. Si se establece el umbral en un valor muy alto, es posible que la alarma sea demasiado sensible en los períodos de tráfico bajo normal y esperado. Por el contrario, si se establece en un valor muy bajo, la alarma podría pasar por alto descensos anómalos más pequeños en el volumen del tráfico.

Período: 60

Puntos de datos para la alarma: 10

Períodos de evaluación: 10

Operador de comparación: LESS_THAN_THRESHOLD

MessageCount

Dimensiones: ApiId, Stage, Route

Descripción de la alarma: esta alarma ayuda a detectar un volumen de tráfico bajo para la ruta de la API de WebSocket en la etapa. Esto puede indicar un problema con los clientes al llamar a la API, como el uso de puntos de conexión incorrectos, o problemas con el backend al enviar mensajes a los clientes. También, podría indicar un problema con la configuración o los permisos de la API, lo que haría que los clientes no puedan acceder a ella.

Finalidad: esta alarma puede detectar un volumen de tráfico bajo imprevisto para la ruta de la API de WebSocket en la etapa. Recomendamos que cree esta alarma si la API recibe y envía un número predecible y constante de mensajes en condiciones normales. No recomendamos esta alarma para las API que no esperan un tráfico constante y uniforme.

Estadística: SampleCount

Umbral recomendado: depende de la situación

Justificación del umbral: establezca el umbral en función del análisis de datos históricos para determinar cuál es el recuento de mensajes de referencia esperado para la API. Si se establece el umbral en un valor muy alto, es posible que la alarma sea demasiado sensible en los períodos de tráfico bajo normal y esperado. Por el contrario, si se establece en un valor muy bajo, la alarma podría pasar por alto descensos anómalos más pequeños en el volumen del tráfico.

Período: 60

Puntos de datos para la alarma: 10

Períodos de evaluación: 10

Operador de comparación: LESS_THAN_THRESHOLD

ClientError

Dimensiones: ApiId, Stage

Descripción de la alarma: esta alarma detecta una alta tasa de errores del cliente. Esto puede indicar un problema en los parámetros de autorización o del mensaje. También podría significar que se eliminó una ruta o que un cliente solicita una que no existe en la API. Considere la posibilidad de habilitar los Registros de CloudWatch y comprobar si hay algún error que pueda causar los errores 4XX. Además, considere la posibilidad de habilitar métricas detalladas de CloudWatch para ver estas métricas por ruta, lo que le ayudará a reducir la búsqueda del origen de los errores. Los errores también pueden deberse a que se supera la limitación configurada. Si las respuestas y los registros indican tasas altas e inesperadas de errores 429, siga esta guía para solucionar el problema.

Finalidad: esta alarma puede detectar altas tasas de errores del cliente en los mensajes de la puerta de enlace de la API de WebSocket.

Estadística: Average

Umbral recomendado: 0,05

Justificación del umbral: el umbral sugerido detecta cuándo más del 5 % del total de las solicitudes reciben errores 4XX. Puede ajustar el umbral para adaptarlo al tráfico de las solicitudes y a sus tasas de error aceptables. También puede analizar los datos históricos para determinar la tasa de error aceptable para la carga de trabajo de la aplicación y, a continuación, ajustar el umbral en consecuencia. Los errores 4XX que se producen con frecuencia deben ser objeto de alarma. Sin embargo, si se establece un valor muy bajo para el umbral, la alarma puede ser demasiado sensible.

Período: 60

Puntos de datos para la alarma: 5

Períodos de evaluación: 5

Operador de comparación: GREATER_THAN_THRESHOLD

ExecutionError

Dimensiones: ApiId, Stage

Descripción de la alarma: esta alarma ayuda a detectar una alta tasa de errores de ejecución. Esto puede deberse a los errores 5XX en la integración, problemas con los permisos u otros factores que impidan la invocación correcta de la integración, como la limitación o la eliminación de la integración. Considere la posibilidad de habilitar los Registros de CloudWatch para su API y comprobar los registros para ver el tipo y la causa de los errores. Además, considere la posibilidad de habilitar las métricas detalladas de CloudWatch para obtener una vista de esta métrica por ruta, lo que le ayudará a reducir la búsqueda del origen de los errores. Esta documentación también puede ayudarle a solucionar la causa de cualquier error de conexión.

Finalidad: esta alarma puede detectar altas tasas de errores de ejecución en los mensajes de la puerta de enlace de la API de WebSocket.

Estadística: Average

Umbral recomendado: 0,05

Justificación del umbral: el umbral sugerido detecta cuándo se producen errores de ejecución en más del 5 % del total de las solicitudes. Puede ajustar el umbral para adaptarlo al tráfico de las solicitudes, así como a sus tasas de error aceptables. Puede analizar los datos históricos para determinar la tasa de error aceptable para la carga de trabajo de la aplicación y, a continuación, ajustar el umbral en consecuencia. Los errores de ejecución que se producen con frecuencia deben ser objeto de alarma. Sin embargo, si se establece un valor muy bajo para el umbral, la alarma puede ser demasiado sensible.

Período: 60

Puntos de datos para la alarma: 3

Períodos de evaluación: 3

Operador de comparación: GREATER_THAN_THRESHOLD

Amazon EC2 Auto Scaling

GroupInServiceCapacity

Dimensiones: AutoScalingGroupName

Descripción de la alarma: esta alarma ayuda a detectar cuando la capacidad del grupo está por debajo de la capacidad deseada requerida para la carga de trabajo. Para solucionar el problema, compruebe si sus actividades de escalado fallaron en el lanzamiento y confirme que la configuración de capacidad deseada es la correcta.

Finalidad: esta alarma puede detectar una baja disponibilidad en su grupo de escalado automático debido a errores de lanzamiento o a lanzamientos suspendidos.

Estadística: Average

Umbral recomendado: depende de la situación

Justificación del umbral: el valor del umbral debe ser la capacidad mínima requerida para ejecutar la carga de trabajo. En la mayoría de los casos, puede configurarse para que coincida con la métrica GroupDesiredCapacity.

Período: 60

Puntos de datos para la alarma: 10

Períodos de evaluación: 10

Operador de comparación: LESS_THAN_THRESHOLD

Amazon CloudFront

5xxErrorRate

Dimensiones: DistributionId, Region=Global

Descripción de la alarma: esta alarma supervisa el porcentaje de respuestas de error 5XX del servidor de origen para ayudarle a detectar si el servicio CloudFront tiene problemas. Consulte Solucionar respuestas de error del origen para obtener información que le ayude a entender los problemas de su servidor. Además, active las métricas adicionales para obtener métricas de error detalladas.

Finalidad: esta alarma se utiliza para detectar problemas al atender las solicitudes del servidor de origen o problemas de comunicación entre CloudFront y el servidor de origen.

Estadística: Average

Umbral recomendado: depende de la situación

Justificación del umbral: el valor del umbral recomendado para esta alarma depende en gran medida de la tolerancia para las respuestas de 5XX. Puede analizar las tendencias y los datos históricos y, a continuación, establecer el umbral en consecuencia. Dado que los errores 5XX pueden deberse a problemas transitorios, le recomendamos que establezca el umbral en un valor superior a 0 para que la alarma no sea demasiado sensible.

Período: 60

Puntos de datos para la alarma: 5

Períodos de evaluación: 5

Operador de comparación: GREATER_THAN_THRESHOLD

OriginLatency

Dimensiones: DistributionId, Region=Global

Descripción de la alarma: la alarma ayuda a supervisar si el servidor de origen tarda demasiado en responder. Si el servidor tarda demasiado en responder, es posible que se agote el tiempo de espera. Consulte Buscar y corregir respuestas con retardo desde aplicaciones en su servidor de origen si experimenta valores altos de `OriginLatency` de forma constante.

Finalidad: esta alarma se utiliza para detectar problemas con un servidor de origen que tarda demasiado en responder.

Estadística: p90

Umbral recomendado: depende de la situación

Justificación del umbral: debe calcular el valor de alrededor del 80 % del tiempo de espera de la respuesta de origen y utilizar el resultado como valor para el umbral. Si esta métrica se acerca con frecuencia al valor de tiempo de espera de la respuesta de origen, es posible que comience a experimentar errores 504.

Período: 60

Puntos de datos para la alarma: 5

Períodos de evaluación: 5

Operador de comparación: GREATER_THAN_THRESHOLD

FunctionValidationErrors

Dimensiones: DistributionId, FunctionName, Region=Global

Descripción de la alarma: esta alarma ayuda a supervisar los errores de validación de las funciones de CloudFront para que pueda tomar medidas para resolverlos. Analice los registros de funciones de CloudWatch y observe el código de la función para encontrar y resolver la causa raíz del problema. Consulte Restricciones en funciones de la periferia para comprender los errores de configuración habituales de CloudFront Functions.

Finalidad: esta alarma se utiliza para detectar errores de validación en las funciones de CloudFront.

Estadística: Sum

Umbral recomendado: 0,0

Justificación del umbral: un valor superior a 0 indica un error de validación. Recomendamos establecer el umbral en 0 porque los errores de validación implican un problema cuando las funciones de CloudFront se devuelven a CloudFront. Por ejemplo, CloudFront necesita el encabezado Host HTTP para procesar una solicitud. No hay nada que impida a un usuario eliminar el encabezado Host del código de las funciones de CloudFront. Sin embargo, cuando CloudFront recibe la respuesta y falta el encabezado del Host, CloudFront devuelve un error de validación.

Período: 60

Puntos de datos para la alarma: 2

Períodos de evaluación: 2

Operador de comparación: GREATER_THAN_THRESHOLD

FunctionExecutionErrors

Dimensiones: DistributionId, FunctionName, Region=Global

Descripción de la alarma: esta alarma le ayuda a supervisar los errores de ejecución de las funciones de CloudFront para que pueda tomar medidas para resolverlos. Analice los registros de funciones de CloudWatch y observe el código de la función para encontrar y resolver la causa raíz del problema.

Finalidad: esta alarma se utiliza para detectar errores de ejecución de las funciones de CloudFront.

Estadística: Sum

Umbral recomendado: 0,0

Justificación del umbral: recomendamos establecer el umbral en 0 porque un error de ejecución indica un problema con el código que se produce en tiempo de ejecución.

Período: 60

Puntos de datos para la alarma: 5

Períodos de evaluación: 5

Operador de comparación: GREATER_THAN_THRESHOLD

FunctionThrottles

Dimensiones: DistributionId, FunctionName, Region=Global

Descripción de la alarma: esta alarma ayuda a supervisar si la función de CloudFront está limitada. Si su función está limitada, significa que está tardando demasiado en ejecutarse. Para evitar limitaciones de funciones, considere optimizar el código de la función.

Finalidad: esta alarma puede detectar cuando la función de CloudFront está restringida para que pueda reaccionar, resolver el problema y ofrecer una experiencia de cliente fluida.

Estadística: Sum

Umbral recomendado: 0,0

Justificación del umbral: recomendamos establecer el umbral en 0 para permitir una resolución más rápida de los limitadores de las funciones.

Período: 60

Puntos de datos para la alarma: 5

Períodos de evaluación: 5

Operador de comparación: GREATER_THAN_THRESHOLD

Amazon Cognito

SignUpThrottles

Dimensiones: UserPool, UserPoolClient

Descripción de la alarma: esta alarma supervisa el recuento de solicitudes limitadas. Si los usuarios se ven limitados con frecuencia, debe aumentar el límite mediante una solicitud de aumento en la cuota de servicio. Consulte Cuotas en Amazon Cognito para obtener información acerca de cómo solicitar un aumento de cuotas. Para tomar medidas de forma proactiva, considere la posibilidad de realizar un seguimiento de la cuota de uso.

Finalidad: esta alarma ayuda a supervisar si se producen solicitudes de registro limitadas. Esto puede ayudarle a saber cuándo tomar medidas para mitigar cualquier deterioro en la experiencia de registro. La limitación sostenida de las solicitudes es una experiencia negativa de registro de usuarios.

Estadística: Sum

Umbral recomendado: depende de la situación

Justificación del umbral: un grupo de usuarios bien aprovisionado no debería sufrir ningún tipo de limitación que afecte a varios puntos de datos. Por lo tanto, un umbral típico para una carga de trabajo esperada debería ser cero. En el caso de una carga de trabajo irregular con ráfagas frecuentes, puede analizar los datos históricos para determinar la limitación aceptable de la carga de trabajo de la aplicación y, a continuación, ajustar el umbral en consecuencia. Se debe volver a realizar una solicitud limitada para minimizar el impacto en la aplicación.

Período: 60

Puntos de datos para la alarma: 5

Períodos de evaluación: 5

Operador de comparación: GREATER_THAN_THRESHOLD

SignInThrottles

Dimensiones: UserPool, UserPoolClient

Descripción de la alarma: esta alarma supervisa el recuento de solicitudes de autenticación de usuarios limitadas. Si los usuarios se ven limitados con frecuencia, es posible que tenga que aumentar el límite mediante una solicitud de aumento de la cuota de servicio. Consulte Cuotas en

Amazon Cognito para obtener información acerca de cómo solicitar un aumento de cuotas. Para tomar medidas de forma proactiva, considere la posibilidad de realizar un seguimiento de la cuota de uso.

Finalidad: esta alarma ayuda a supervisar la aparición de solicitudes de inicio de sesión limitadas. Esto puede ayudarle a saber cuándo tomar medidas para mitigar cualquier deterioro en la experiencia de inicio de sesión. La limitación sostenida de las solicitudes es una experiencia de autenticación negativa para los usuarios.

Estadística: Sum

Umbral recomendado: depende de la situación

Justificación del umbral: un grupo de usuarios bien aprovisionado no debería sufrir ningún tipo de limitación que afecte a varios puntos de datos. Por lo tanto, un umbral típico para una carga de trabajo esperada debería ser cero. En el caso de una carga de trabajo irregular con ráfagas frecuentes, puede analizar los datos históricos para determinar la limitación aceptable de la carga de trabajo de la aplicación y, a continuación, ajustar el umbral en consecuencia. Se debe volver a realizar una solicitud limitada para minimizar el impacto en la aplicación.

Período: 60

Puntos de datos para la alarma: 5

Períodos de evaluación: 5

Operador de comparación: GREATER_THAN_THRESHOLD

TokenRefreshThrottles

Dimensiones: UserPool, UserPoolClient

Descripción de la alarma: puede establecer el valor del umbral para que se adapte al tráfico de la solicitud y para que coincida con la limitación aceptable para las solicitudes de actualización de los tokens. La limitación se utiliza para proteger el sistema de demasiadas solicitudes. Sin embargo, es importante supervisar si también tiene un aprovisionamiento insuficiente para tu tráfico normal. Puede analizar los datos históricos para determinar la limitación aceptable para la carga de trabajo de la aplicación y, a continuación, ajustar el umbral de la alarma para que sea superior al nivel de limitación aceptable. La aplicación o el servicio deben volver a intentar las solicitudes limitadas, ya que son transitorias. Por lo tanto, un valor muy bajo para el umbral puede provocar que la alarma sea sensible.

Finalidad: esta alarma ayuda a supervisar la aparición de solicitudes de actualización de tokens limitadas. Esto puede ayudarle a saber cuándo tomar medidas para mitigar cualquier posible problema, a fin de garantizar una experiencia de usuario fluida, y el buen estado y la fiabilidad de su sistema de autenticación. La limitación sostenida de las solicitudes es una experiencia de autenticación negativa para los usuarios.

Estadística: Sum

Umbral recomendado: depende de la situación

Justificación del umbral: el valor del umbral también se puede configurar o ajustar para adaptarlo al tráfico de la solicitud, así como a una limitación aceptable para las solicitudes de actualización de los tokens. La limitación se utiliza para proteger el sistema de demasiadas solicitudes. Sin embargo, también es importante supervisar si no está bien aprovisionado para el tráfico normal y comprobar si eso es lo que causa el impacto. Los datos históricos también se pueden analizar para determinar cuál es la limitación aceptable para la carga de trabajo de la aplicación, y el umbral se puede ajustar por encima del nivel de limitación aceptable habitual. La aplicación o el servicio deben volver a intentar las solicitudes limitadas, ya que son transitorias. Por lo tanto, un valor muy bajo para el umbral puede provocar que la alarma sea sensible.

Período: 60

Puntos de datos para la alarma: 5

Períodos de evaluación: 5

Operador de comparación: GREATER_THAN_THRESHOLD

FederationThrottles

Dimensiones: UserPool, UserPoolClient, IdentityProvider

Descripción de la alarma: esta alarma supervisa el recuento de solicitudes de federación de identidades limitadas. Si observa una limitación constante, esto podría indicar que necesita aumentar el límite mediante una solicitud de un aumento de la cuota de servicio. Consulte Cuotas en Amazon Cognito para obtener información acerca de cómo solicitar un aumento de cuotas.

Finalidad: esta alarma ayuda a supervisar la aparición de solicitudes limitadas de federación de identidades. Esto puede ayudarle a dar respuestas proactivas a los cuellos de botella de rendimiento o a los errores de configuración y garantizar una experiencia de autenticación fluida para sus usuarios. La limitación sostenida de las solicitudes es una experiencia de autenticación negativa para los usuarios.

Estadística: Sum

Umbral recomendado: depende de la situación

Justificación del umbral: puede establecer el umbral para que se adapte al tráfico de la solicitud y para que coincida con la limitación aceptable para las solicitudes de federación de identidades. La limitación se utiliza para proteger el sistema de demasiadas solicitudes. Sin embargo, es importante supervisar si también tiene un aprovisionamiento insuficiente para tu tráfico normal. Puede analizar los datos históricos para encontrar la limitación aceptable para la carga de trabajo de la aplicación y, a continuación, establecer el umbral en un valor superior a su nivel de limitación aceptable. La aplicación o el servicio deben volver a intentar las solicitudes limitadas, ya que son transitorias. Por lo tanto, un valor muy bajo para el umbral puede provocar que la alarma sea sensible.

Período: 60

Puntos de datos para la alarma: 5

Períodos de evaluación: 5

Operador de comparación: GREATER_THAN_THRESHOLD

Amazon DynamoDB

AccountProvisionedReadCapacityUtilization

Dimensiones: None

Descripción de la alarma: esta alarma detecta si la capacidad de lectura de la cuenta está cerca de alcanzar el límite aprovisionado. Si esto ocurre, puede aumentar la cuota de la cuenta para utilizar la capacidad de lectura. Puede ver sus cuotas actuales de unidades de capacidad de lectura y solicitar aumentos mediante Service Quotas.

Finalidad: la alarma puede detectar si la utilización de la capacidad de lectura de la cuenta se acerca a la utilización de la capacidad de lectura aprovisionada. Si la utilización alcanza su límite máximo, DynamoDB comienza a limitar las solicitudes de lectura.

Estadística: Maximum

Umbral recomendado: 80,0

Justificación del umbral: fije el umbral en el 80 %, de modo que se puedan tomar medidas (como aumentar los límites de la cuenta) antes de que alcance su capacidad máxima para evitar la limitación.

Período: 300

Puntos de datos para la alarma: 2

Períodos de evaluación: 2

Operador de comparación: GREATER_THAN_THRESHOLD

AccountProvisionedWriteCapacityUtilization

Dimensiones: None

Descripción de la alarma: esta alarma detecta si la capacidad de escritura de la cuenta está cerca de alcanzar el límite aprovisionado. Si esto ocurre, puede aumentar la cuota de la cuenta para utilizar la capacidad de escritura. Puede ver sus cuotas actuales de unidades de capacidad de escritura y solicitar aumentos mediante Service Quotas.

Finalidad: esta alarma puede detectar si la utilización de la capacidad de escritura de la cuenta se acerca a la utilización de la capacidad de escritura aprovisionada. Si la utilización alcanza su límite máximo, DynamoDB comienza a limitar las solicitudes de escritura.

Estadística: Maximum

Umbral recomendado: 80,0

Justificación del umbral: establezca el umbral en el 80 %, de modo que la acción (como aumentar los límites de la cuenta) se pueda tomar antes de que alcance su capacidad máxima para evitar la limitación.

Período: 300

Puntos de datos para la alarma: 2

Períodos de evaluación: 2

Operador de comparación: GREATER_THAN_THRESHOLD

AgeOfOldestUnreplicatedRecord

Dimensiones: TableName, DelegatedOperation

Descripción de la alarma: esta alarma detecta el retraso en la replicación a un flujo de datos de Kinesis. En funcionamiento normal, `AgeOfOldestUnreplicatedRecord` debe estar en el orden de los milisegundos. Este número crece según los intentos de replicación fallidos cuando se deben a elecciones de configuración controladas por el cliente. Los ejemplos de las configuraciones controladas por el cliente que provocan intentos de replicación fallidos son una capacidad de flujo de datos de Kinesis subaprovisionada que produce una limitación excesiva o una actualización manual de las políticas de acceso del flujo de datos de Kinesis que evita que DynamoDB agregue datos al flujo de datos. Para mantener esta métrica lo más baja posible, debe garantizar el aprovisionamiento correcto de la capacidad del flujo de datos de Kinesis y asegurarse de que los permisos de DynamoDB no se modifiquen.

Finalidad: esta alarma puede supervisar los intentos de replicación fallidos y el consiguiente retraso en la replicación en el flujo de datos de Kinesis.

Estadística: Maximum

Umbral recomendado: depende de la situación

Justificación del umbral: defina el umbral de acuerdo con el retraso de replicación deseado, medido en milisegundos. Este valor depende de los requisitos de la carga de trabajo y del rendimiento esperado.

Período: 300

Puntos de datos para la alarma: 3

Períodos de evaluación: 3

Operador de comparación: GREATER_THAN_THRESHOLD

FailedToReplicateRecordCount

Dimensiones: TableName, DelegatedOperation

Descripción de la alarma: esta alarma detecta el número de registros que DynamoDB no pudo replicar en el flujo de datos de Kinesis. Algunos elementos de más de 34 KB podrían expandirse de tamaño para cambiar los registros de datos que superan el límite de tamaño del elemento de 1 MB de Kinesis Data Streams. Esta expansión de tamaño se produce cuando estos elementos mayores a 34 KB incluyen un gran número de valores de atributos booleanos o vacíos. Los valores de atributos booleanos y vacíos se almacenan como 1 byte en DynamoDB, pero se expanden hasta 5 bytes cuando se serializan mediante JSON estándar para la replicación de Kinesis Data Streams. DynamoDB no puede replicar tales registros de cambios en el flujo de

datos de Kinesis. DynamoDB omite estos registros de datos de cambios y continúa replicando automáticamente los registros posteriores.

Finalidad: esta alarma puede supervisar el número de registros que DynamoDB no pudo replicar en el flujo de datos de Kinesis debido al límite de tamaño de los elementos de Kinesis Data Streams.

Estadística: Sum

Umbral recomendado: 0,0

Justificación del umbral: defina el umbral en 0 para detectar cualquier registro que DynamoDB no pueda replicar.

Período: 60

Puntos de datos para la alarma: 1

Períodos de evaluación: 1

Operador de comparación: GREATER_THAN_THRESHOLD

ReadThrottleEvents

Dimensiones: TableName

Descripción de la alarma: esta alarma detecta si hay un número elevado de solicitudes de lectura que se están limitando para la tabla de DynamoDB. Para solucionar el problema, consulte Solución de problemas de limitación en Amazon DynamoDB.

Finalidad: esta alarma puede detectar una limitación constante de las solicitudes de lectura a la tabla de DynamoDB. La limitación constante de las solicitudes de lectura puede afectar de forma negativa a las operaciones de lectura de la carga de trabajo y reducir la eficiencia general del sistema.

Estadística: Sum

Umbral recomendado: depende de la situación

Justificación del umbral: establezca el umbral de acuerdo con el tráfico de lectura esperado para la tabla de DynamoDB, teniendo en cuenta un nivel de limitación aceptable. Es importante supervisar si el aprovisionamiento es insuficiente y si no se está produciendo una limitación constante. También, puede analizar los datos históricos para encontrar el nivel de limitación

aceptable para la carga de trabajo de la aplicación y, a continuación, ajustar el umbral para que sea superior al nivel de limitación habitual. La aplicación o el servicio deben volver a intentar las solicitudes limitadas, ya que son transitorias. Por lo tanto, un umbral muy bajo puede provocar que la alarma sea demasiado sensible y provocar transiciones de estado no deseadas.

Período: 60

Puntos de datos para la alarma: 5

Períodos de evaluación: 5

Operador de comparación: GREATER_THAN_THRESHOLD

ReadThrottleEvents

Dimensiones: TableName, GlobalSecondaryIndexName

Descripción de la alarma: esta alarma detecta si se limita un número elevado de solicitudes de lectura en el índice secundario global de la tabla de DynamoDB. Para solucionar el problema, consulte Solución de problemas de limitación en Amazon DynamoDB.

Finalidad: la alarma puede detectar una limitación constante de las solicitudes de lectura del índice secundario global de la tabla de DynamoDB. La limitación constante de las solicitudes de lectura puede afectar de forma negativa a las operaciones de lectura de la carga de trabajo y reducir la eficiencia general del sistema.

Estadística: Sum

Umbral recomendado: depende de la situación

Justificación del umbral: establezca el umbral de acuerdo con el tráfico de lectura esperado para la tabla de DynamoDB, teniendo en cuenta un nivel de limitación aceptable. Es importante supervisar si tiene un aprovisionamiento insuficiente y si no se produce una limitación constante. También, puede analizar los datos históricos para encontrar un nivel de limitación aceptable para la carga de trabajo de la aplicación y, a continuación, ajustar el umbral para que sea superior al nivel de limitación aceptable habitual. La aplicación o el servicio deben volver a intentar las solicitudes limitadas, ya que son transitorias. Por lo tanto, un umbral muy bajo puede provocar que la alarma sea demasiado sensible y provocar transiciones de estado no deseadas.

Período: 60

Puntos de datos para la alarma: 5

Períodos de evaluación: 5

Operador de comparación: GREATER_THAN_THRESHOLD

ReplicationLatency

Dimensiones: TableName, ReceivingRegion

Descripción de la alarma: la alarma detecta si la réplica de una región de la tabla global va retrasada respecto a la región de origen. La latencia puede aumentar si una región de AWS se encuentra degradada y tiene una réplica de tabla en esa región. En este caso, puede redirigir temporalmente la actividad de lectura y escritura de la aplicación a otra región de AWS. Si utiliza tablas globales del 29.11.2017 (heredadas), debe comprobar que las unidades de capacidad de escritura (WCU) son idénticas para cada una de las tablas de réplica. También puede asegurarse de seguir las pautas de las Prácticas recomendadas y requisitos para la administración de tablas globales.

Finalidad: la alarma puede detectar si la tabla de réplica de una región se retrasa al replicar los cambios de otra región. Esto podría provocar que su réplica se diferencie de las demás réplicas. Resulta útil conocer la latencia de replicación de cada región de AWS y avisar si esa latencia de replicación aumenta de forma continua. La replicación de la tabla se aplica solo a las tablas globales.

Estadística: Average

Umbral recomendado: depende de la situación

Justificación del umbral: el valor del umbral recomendado para esta alarma depende en gran medida del caso de uso. Las latencias de replicación superiores a 3 minutos suelen ser motivo de investigación. Revise la importancia y los requisitos del retraso de la replicación, analice las tendencias históricas y, en base a eso, seleccione el umbral.

Período: 60

Puntos de datos para la alarma: 15

Períodos de evaluación: 15

Operador de comparación: GREATER_THAN_THRESHOLD

SuccessfulRequestLatency

Dimensiones: TableName, Operation

Descripción de la alarma: esta alarma detecta una latencia elevada para el funcionamiento de la tabla de DynamoDB (indicada por el valor de dimensión de la `Operation` de la alarma). Consulte este documento para solucionar los problemas de latencia en Amazon DynamoDB.

Finalidad: esta alarma puede detectar una latencia elevada para la operación de la tabla de DynamoDB. Una latencia más alta de las operaciones puede afectar de forma negativa a la eficiencia general del sistema.

Estadística: Average

Umbral recomendado: depende de la situación

Justificación del umbral: DynamoDB proporciona una latencia media de milisegundos de un solo dígito para operaciones únicas como GetItem y PutItem, entre otras. Sin embargo, puede establecer el umbral en función de la tolerancia aceptable para la latencia según el tipo de operación y la tabla implicadas en la carga de trabajo. Puede analizar los datos históricos de esta métrica para encontrar la latencia habitual de la operación de la tabla y, a continuación, establecer el umbral en un número que represente un retraso crítico de la operación.

Período: 60

Puntos de datos para la alarma: 10

Períodos de evaluación: 10

Operador de comparación: GREATER_THAN_THRESHOLD

SystemErrors

Dimensiones: TableName

Descripción de la alarma: esta alarma detecta un número elevado y sostenido de errores del sistema en las solicitudes de tablas de DynamoDB. Si sigue recibiendo errores 5XX, abra el Service Health Dashboard de AWS para comprobar si hay problemas operativos con el servicio. Puede utilizar esta alarma para recibir notificaciones en caso de que se produzca un problema de servicio interno prolongado por parte de DynamoDB y le ayudará a establecer una correlación con el problema al que se enfrenta la aplicación del cliente. Consulte Control de errores con DynamoDB para obtener más información.

Finalidad: esta alarma puede detectar errores de sistema persistentes en las solicitudes de tablas de DynamoDB. Los errores del sistema indican errores de servicio internos desde DynamoDB y ayudan a correlacionarlos con el problema que tiene el cliente.

Estadística: Sum

Umbral recomendado: depende de la situación

Justificación del umbral: establezca el umbral de acuerdo con el tráfico esperado, teniendo en cuenta un nivel aceptable de errores del sistema. También, puede analizar los datos históricos para encontrar el recuento de errores aceptable para la carga de trabajo de la aplicación y, a continuación, ajustar el umbral en consecuencia. La aplicación o el servicio deben volver a reintentar los errores del sistema, ya que son transitorios. Por lo tanto, un umbral muy bajo puede provocar que la alarma sea demasiado sensible y causar transiciones de estado no deseadas.

Período: 60

Puntos de datos para la alarma: 15

Períodos de evaluación: 15

Operador de comparación: GREATER_THAN_THRESHOLD

ThrottledPutRecordCount

Dimensiones: TableName, DelegatedOperation

Descripción de la alarma: esta alarma detecta los registros que el flujo de datos de Kinesis limita durante la replicación de la captura de datos de cambios en Kinesis. Esta limitación se debe a que la capacidad de flujo de datos de Kinesis es insuficiente. Si experimenta una limitación controlada excesiva y regular, es posible que tenga que aumentar el número de fragmentos del flujo de Kinesis proporcionalmente al rendimiento de escritura observado de la tabla. Para obtener más información sobre cómo determinar el tamaño de un flujo de datos de Kinesis, consulte Determinar el tamaño inicial de un flujo de datos de Kinesis.

Finalidad: esta alarma puede supervisar el número de registros que el flujo de datos de Kinesis limitó debido a la capacidad insuficiente del flujo de datos de Kinesis.

Estadística: Maximum

Umbral recomendado: depende de la situación

Justificación del umbral: es posible que experimente cierta limitación durante los picos de uso excepcionales, pero los registros limitados deben ser lo más bajos posible para evitar una latencia de replicación mayor (DynamoDB vuelve a intentar enviar los registros limitados al flujo de datos de Kinesis). Establezca el umbral en un número que pueda ayudarlo a identificar el exceso de

limitación normal. También, puede analizar los datos históricos de esta métrica para encontrar las tasas de limitación aceptables para la carga de trabajo de la aplicación. Ajuste el umbral a un valor que la aplicación pueda tolerar en función del caso de uso.

Período: 60

Puntos de datos para la alarma: 10

Períodos de evaluación: 10

Operador de comparación: GREATER_THAN_THRESHOLD

UserErrors

Dimensiones: None

Descripción de la alarma: esta alarma detecta un número elevado y sostenido de errores de usuario en las solicitudes de tablas de DynamoDB. Puede comprobar los registros de las aplicaciones del cliente durante el período de emisión para comprobar por qué las solicitudes no son válidas. Puede comprobar el código 400 de estado HTTP para ver el tipo de error que se produce y tomar las medidas correspondientes. Es posible que deba corregir la lógica de la aplicación para crear solicitudes válidas.

Finalidad: esta alarma puede detectar errores de usuario persistentes en las solicitudes de tablas de DynamoDB. Los errores de usuario en las operaciones solicitadas significan que el cliente produce solicitudes no válidas y falla.

Estadística: Sum

Umbral recomendado: depende de la situación

Justificación del umbral: defina el umbral en cero para detectar cualquier error del lado del cliente. O puede establecerlo en un valor más alto si quiere evitar que se active la alarma por un número de errores muy bajo. Decida en función del caso de uso y del tráfico de las solicitudes.

Período: 60

Puntos de datos para la alarma: 10

Períodos de evaluación: 10

Operador de comparación: GREATER_THAN_THRESHOLD

WriteThrottleEvents

Dimensiones: TableName

Descripción de la alarma: esta alarma detecta si se limita un número elevado de solicitudes de escritura en la tabla de DynamoDB. Consulte Solución de problemas de limitación en Amazon DynamoDB para solucionar este problema.

Finalidad: esta alarma puede detectar una limitación constante de las solicitudes de escritura en la tabla de DynamoDB. La limitación constante de las solicitudes de escritura puede afectar de forma negativa a la carga de trabajo de las operaciones de escritura y reducir la eficiencia general del sistema.

Estadística: Sum

Umbral recomendado: depende de la situación

Justificación del umbral: establezca el umbral de acuerdo con el tráfico de escritura esperado para la tabla de DynamoDB, teniendo en cuenta un nivel de limitación aceptable. Es importante supervisar si tiene un aprovisionamiento insuficiente y si no se produce una limitación constante. También, puede analizar los datos históricos para determinar el nivel de limitación aceptable para la carga de trabajo de la aplicación y, a continuación, ajustar el umbral a un valor superior al nivel de limitación aceptable habitual. La aplicación o el servicio deben volver a intentar las solicitudes limitadas, ya que son transitorias. Por lo tanto, un umbral muy bajo puede provocar que la alarma sea demasiado sensible y causar transiciones de estado no deseadas.

Período: 60

Puntos de datos para la alarma: 5

Períodos de evaluación: 5

Operador de comparación: GREATER_THAN_THRESHOLD

WriteThrottleEvents

Dimensiones: TableName, GlobalSecondaryIndexName

Descripción de la alarma: esta alarma detecta si se limita un número elevado de solicitudes de escritura para el índice secundario global de la tabla de DynamoDB. Consulte Solución de problemas de limitación en Amazon DynamoDB para solucionar este problema.

Finalidad: esta alarma puede detectar una limitación constante de las solicitudes de escritura para el índice secundario global de la tabla de DynamoDB. La limitación constante de las solicitudes de escritura puede afectar de forma negativa a la carga de trabajo de las operaciones de escritura y reducir la eficiencia general del sistema.

Estadística: Sum

Umbral recomendado: depende de la situación

Justificación del umbral: establezca el umbral de acuerdo con el tráfico de escritura esperado para la tabla de DynamoDB, teniendo en cuenta un nivel de limitación aceptable. Es importante supervisar si tiene un aprovisionamiento insuficiente y si no se produce una limitación constante. También, puede analizar los datos históricos para encontrar el nivel de limitación aceptable para la carga de trabajo de la aplicación y, a continuación, ajustar el umbral a un valor superior al nivel de limitación aceptable habitual. La aplicación o el servicio deben volver a intentar las solicitudes limitadas, ya que son transitorias. Por lo tanto, un valor muy bajo puede provocar que la alarma sea demasiado sensible y provocar transiciones de estado no deseadas.

Período: 60

Puntos de datos para la alarma: 5

Períodos de evaluación: 5

Operador de comparación: GREATER_THAN_THRESHOLD

Amazon EBS

VolumeStalledIOCheck

Dimensiones: VolumeId, InstanceId

Descripción de la alarma: esta alarma lo ayuda a supervisar el rendimiento de E/S de los volúmenes de Amazon EBS. Esta comprobación detecta problemas subyacentes en la infraestructura de Amazon EBS, como problemas de equipo o software en los subsistemas de almacenamiento de los volúmenes de Amazon EBS, problemas de equipo en el host físico que afectan a la accesibilidad de los volúmenes de Amazon EBS desde su instancia de Amazon EC2, y puede detectar problemas de conectividad entre la instancia y los volúmenes de Amazon EBS. Si la comprobación de E/S estancadas falla, puede esperar a que AWS resuelva el problema o tomar medidas, como reemplazar los volúmenes afectados o detener y reiniciar la instancia a la

que está asociado el volumen. En la mayoría de los casos, cuando se produce un error en esta métrica, Amazon EBS diagnosticará y recuperará automáticamente el volumen en cuestión de minutos.

Intención: esta alarma puede detectar el estado de sus volúmenes de Amazon EBS para determinar cuándo estos volúmenes están deteriorados y no pueden completar las operaciones de E/S.

Estadística: Maximum

Umbral recomendado: 1,0

Justificación del umbral: cuando falla una comprobación de estado, el valor de esta métrica es 1. El umbral se establece de modo que, siempre que la verificación de estado falle, la alarma esté en estado de ALARMA.

Período: 60

Puntos de datos para la alarma: 10

Períodos de evaluación: 10

Operador de comparación: GREATER_THAN_OR_EQUAL_TO_THRESHOLD

Amazon EC2

CPUUtilization

Dimensiones: InstanceId

Descripción de la alarma: esta alarma ayuda a supervisar el uso de la CPU de una instancia de EC2. En función de la aplicación, puede que los niveles de utilización siempre altos sean normales. Pero, si se degrada el rendimiento y la aplicación no está limitada por la E/S del disco, la memoria o los recursos de red, una CPU al máximo podría indicar un cuello de botella en los recursos o problemas de rendimiento de la aplicación. Un uso elevado de la CPU puede indicar que es necesario actualizar a una instancia con un uso más intensivo de la CPU. Si la supervisión detallada está habilitada, puede cambiar el período a 60 segundos en lugar de 300 segundos. Para obtener más información, consulte Activar o desactivar la supervisión detallada para las instancias.

Finalidad: esta alarma se utiliza para detectar un uso elevado de la CPU.

Estadística: Average

Umbral recomendado: 80,0

Justificación del umbral: en general, puede establecer el umbral de utilización de la CPU entre el 70 y el 80 %. Sin embargo, puede ajustar este valor en función del nivel de rendimiento y las características de la carga de trabajo aceptables. Para algunos sistemas, un uso elevado y constante de la CPU puede ser normal y no indicar un problema, mientras que para otros puede ser un motivo de preocupación. Analice los datos históricos de uso de la CPU para identificar el uso, determinar qué uso de la CPU es aceptable para su sistema y establecer el umbral correspondiente.

Período: 300

Puntos de datos para la alarma: 3

Períodos de evaluación: 3

Operador de comparación: GREATER_THAN_THRESHOLD

StatusCheckFailed

Dimensiones: InstanceId

Descripción de la alarma: esta alarma ayuda a supervisar las comprobaciones de estado del sistema y las comprobaciones de estado de las instancias. Si alguno de los tipos de verificación de estado falla, esta alarma debería estar en estado de ALARMA.

Finalidad: esta alarma se utiliza para detectar los problemas subyacentes con las instancias, incluidos los fallos en la comprobación del estado del sistema y los fallos en la comprobación del estado de las instancias.

Estadística: Maximum

Umbral recomendado: 1,0

Justificación del umbral: cuando falla una comprobación de estado, el valor de esta métrica es 1. El umbral se establece de modo que, siempre que la verificación de estado falle, la alarma esté en estado de ALARMA.

Período: 300

Puntos de datos para la alarma: 2

Períodos de evaluación: 2

Operador de comparación: GREATER_THAN_OR_EQUAL_TO_THRESHOLD

StatusCheckFailed_AttachedEBS

Dimensiones: InstanceId

Descripción de la alarma: esta alarma le ayuda a supervisar si se puede acceder a los volúmenes de Amazon EBS adjuntos a una instancia y completar operaciones de E/S. Esta comprobación de estado detecta los problemas subyacentes en el cómputo o en la infraestructura de Amazon EBS, como los siguientes:

- Problemas de hardware o software en los subsistemas de almacenamiento subyacentes a los volúmenes de Amazon EBS
- Problemas de hardware en el host físico que afectan a la accesibilidad de los volúmenes de Amazon EBS
- Problemas de conectividad entre la instancia y los volúmenes de Amazon EBS

Si la métrica de comprobación de estado de EBS adjunta falla, puede esperar a que Amazon resuelva el problema o tomar medidas, como reemplazar los volúmenes afectados o detener y reiniciar la instancia.

Intención: esta alarma se utiliza para detectar volúmenes de Amazon EBS inalcanzables adjuntos a una instancia. Esto puede provocar fallos en las operaciones de E/S.

Estadística: Maximum

Umbral recomendado: 1,0

Justificación del umbral: cuando falla una comprobación de estado, el valor de esta métrica es 1. El umbral se establece de modo que, siempre que la verificación de estado falle, la alarma esté en estado de ALARMA.

Período: 60

Puntos de datos para la alarma: 10

Períodos de evaluación: 10

Operador de comparación: GREATER_THAN_OR_EQUAL_TO_THRESHOLD

Amazon ElastiCache

CPUUtilization

Dimensiones: CacheClusterId, CacheNodeId

Descripción de la alarma: esta alarma ayuda a supervisar el uso de la CPU en toda la instancia de ElastiCache, incluidos los procesos del motor de base de datos y otros procesos que se ejecutan en la instancia. AWS ElastiCache admite dos tipos de motores: Memcached y Redis. Cuando se alcanza un uso elevado de la CPU en un nodo de Memcached, debería considerar la posibilidad de escalar verticalmente el tipo de instancia o agregar nuevos nodos de caché. En el caso de Redis, si la carga de trabajo principal son las solicitudes de lectura, debería considerar agregar más réplicas de lectura al clúster de caché. Si la carga de trabajo principal proviene de las solicitudes de escritura, debería considerar agregar más particiones para distribuir la carga de trabajo entre más nodos principales si ejecuta en modo agrupado o escalar verticalmente el tipo de instancia si ejecuta Redis en modo no agrupado.

Finalidad: esta alarma se utiliza para detectar un uso elevado de la CPU en los hosts de ElastiCache. Resulta útil obtener una visión amplia del uso de la CPU en toda la instancia, incluidos los procesos que no están relacionados con el motor.

Estadística: Average

Umbral recomendado: depende de la situación

Justificación del umbral: establezca el umbral en el porcentaje que refleje un nivel de uso crítico de la CPU para su aplicación. En el caso de Memcached, el motor puede utilizar hasta num_threads núcleos. En el caso de Redis, el motor funciona en gran medida con un solo subproceso, pero puede utilizar núcleos adicionales si están disponibles para acelerar la E/S. En la mayoría de los casos, puede establecer el umbral en alrededor del 90 % de la CPU disponible. Debido a que Redis usa un único subproceso, el valor del umbral real debe calcularse como una fracción de la capacidad total del nodo.

Período: 60

Puntos de datos para la alarma: 5

Períodos de evaluación: 5

Operador de comparación: GREATER_THAN_THRESHOLD

CurrConnections

Dimensiones: CacheClusterId, CacheNodeId

Descripción de la alarma: esta alarma detecta un número elevado de conexiones, lo que puede indicar problemas de rendimiento o una carga excesiva. Un aumento constante de `CurrConnections` podría provocar el agotamiento de las 65 000 conexiones disponibles. Puede indicar que las conexiones se cerraron de forma incorrecta en el lado de la aplicación y se dejaron establecidas en el lado del servidor. Debería considerar la posibilidad de utilizar la agrupación de conexiones o los tiempos de espera de conexión inactivos para limitar la cantidad de conexiones realizadas al clúster o, en el caso de Redis, considerar la posibilidad de ajustar tcp-keepalive en su clúster para detectar y eliminar posibles interconexiones inactivas.

Finalidad: la alarma ayuda a identificar los números elevados de conexiones que podrían afectar al rendimiento y la estabilidad del clúster de ElastiCache.

Estadística: Average

Umbral recomendado: depende de la situación

Justificación del umbral: el valor del umbral recomendado para esta alarma depende en gran medida del rango de conexiones aceptable para el clúster. Revise la capacidad y la carga de trabajo prevista del clúster de ElastiCache y analice los recuentos históricos de conexiones durante el uso habitual para establecer una referencia y, a continuación, seleccione el umbral correspondiente. Recuerde que cada nodo puede admitir hasta 65 000 conexiones simultáneas.

Período: 60

Puntos de datos para la alarma: 10

Períodos de evaluación: 10

Operador de comparación: GREATER_THAN_THRESHOLD

DatabaseMemoryUsagePercentage

Dimensiones: CacheClusterId

Descripción de la alarma: esta alarma le ayuda a supervisar el uso de la memoria del clúster. Cuando `DatabaseMemoryUsagePercentage` alcanza el 100 %, se activa la política maxmemory de Redis y es posible que se produzcan expulsiones en función de la política seleccionada. Si ningún objeto de la caché coincide con la política de expulsión, las operaciones

de escritura fallan. Algunas cargas de trabajo esperan o dependen de las expulsiones, pero, si no es así, tendrá que aumentar la capacidad de memoria del clúster. Puede escalar horizontalmente el clúster si agrega más nodos principales, o bien escalarlo con un nodo de mayor tamaño. Consulte Escalado de clústeres de ElastiCache for Redis para obtener más información.

Finalidad: esta alarma se utiliza para detectar un uso elevado de la memoria del clúster, de modo que pueda evitar errores al escribir en el clúster. Resulta útil saber cuándo necesitará escalar verticalmente el clúster si su aplicación no espera sufrir expulsiones.

Estadística: Average

Umbral recomendado: depende de la situación

Justificación del umbral: según los requisitos de memoria de la aplicación y la capacidad de memoria del clúster de ElastiCache, debe establecer el umbral en el porcentaje que refleje el nivel crítico de uso de memoria del clúster. Puede utilizar los datos históricos de uso de memoria como referencia para determinar el umbral de uso de memoria aceptable.

Período: 60

Puntos de datos para la alarma: 5

Períodos de evaluación: 5

Operador de comparación: GREATER_THAN_THRESHOLD

EngineCPUUtilization

Dimensiones: CacheClusterId

Descripción de la alarma: esta alarma ayuda a supervisar el uso de la CPU de un subproceso del motor Redis dentro de la instancia de ElastiCache. Los motivos más comunes por los que la CPU tiene un motor elevado son los comandos de ejecución prolongada que consumen mucha CPU, el elevado número de solicitudes, el aumento del número de nuevas solicitudes de conexión de clientes en un periodo corto y el elevado número de expulsiones cuando la memoria caché no tiene suficiente memoria para almacenar datos nuevos. Debería considerar el Escalado de clústeres de ElastiCache for Redis mediante la adición de más nodos o el escalado vertical del tipo de instancia.

Finalidad: esta alarma se utiliza para detectar un uso elevado de la CPU del subproceso del motor de Redis. Resulta útil si desea supervisar el uso de la CPU del propio motor de base de datos.

Estadística: Average

Umbral recomendado: 90,0

Justificación del umbral: establezca el umbral en un porcentaje que refleje el nivel crítico de uso de la CPU del motor para su aplicación. Puede comparar su clúster mediante su aplicación y la carga de trabajo esperada para correlacionar EngineCPUUtilization con el rendimiento como referencia y, a continuación, establecer el umbral en consecuencia. En la mayoría de los casos, puede establecer el umbral en alrededor del 90 % de la CPU disponible.

Período: 60

Puntos de datos para la alarma: 5

Períodos de evaluación: 5

Operador de comparación: GREATER_THAN_THRESHOLD

ReplicationLag

Dimensiones: CacheClusterId

Descripción de la alarma: esta alarma ayuda a supervisar el estado de la replicación del clúster de ElastiCache. Un retraso de replicación alto significa que el nodo principal o la réplica no pueden mantener el ritmo de la replicación. Si la actividad de escritura es demasiado alta, considere escalar horizontalmente el clúster con la adición de más nodos principales, o, bien, escalar verticalmente el clúster con un tipo de nodo de mayor tamaño. Consulte Escalado de clústeres de ElastiCache for Redis para obtener más información. Si sus réplicas de lectura están sobrecargadas por la cantidad de solicitudes de lectura, considere agregar más réplicas de lectura.

Finalidad: esta alarma se utiliza para detectar un retraso entre las actualizaciones de datos en el nodo principal y su sincronización con el nodo de réplica. Ayuda a garantizar la coherencia de datos de un nodo de clúster de réplica de lectura.

Estadística: Average

Umbral recomendado: depende de la situación

Justificación del umbral: establezca el umbral de acuerdo con los requisitos de su aplicación y el posible impacto del retraso en la replicación. Debe tener en cuenta las velocidades de escritura esperadas de la aplicación y las condiciones de la red para determinar el retraso de replicación aceptable.

Período: 60

Puntos de datos para la alarma: 15

Períodos de evaluación: 15

Operador de comparación: GREATER_THAN_THRESHOLD

Amazon EC2 (**`AWS/ElasticGPUs`**)

GPUConnectivityCheckFailed

Dimensiones: InstanceId, EGPUId

Descripción de la alarma: esta alarma ayuda a detectar fallos de conexión entre la instancia y el acelerador de Elastic Graphics. Elastic Graphics utiliza la red de la instancia para enviar comandos de OpenGL a una tarjeta gráfica conectada de forma remota. Además, se suele utilizar tecnología de acceso remoto para obtener acceso al escritorio que ejecuta una aplicación de OpenGL con un acelerador de Elastic Graphics. Es importante distinguir entre un problema de rendimiento relacionado con la presentación de OpenGL o con la tecnología de acceso remoto del escritorio. Para obtener más información sobre el problema, consulte Investigación de problemas de rendimiento de las aplicaciones.

Finalidad: esta alarma se utiliza para detectar problemas de conectividad desde la instancia al acelerador de Elastic Graphics.

Estadística: Maximum

Umbral recomendado: 0,0

Justificación del umbral: el valor del umbral de 1 indica que falló la conectividad.

Período: 300

Puntos de datos para la alarma: 3

Períodos de evaluación: 3

Operador de comparación: GREATER_THAN_THRESHOLD

GPUHealthCheckFailed

Dimensiones: InstanceId, EGPUId

Descripción de la alarma: esta alarma le ayuda a saber cuándo el estado del acelerador Elastic Graphics es incorrecto. Si el acelerador no está en buen estado, consulte los pasos para la solución de problemas en Resolver problemas de estado incorrecto.

Finalidad: esta alarma se utiliza para detectar si el acelerador de Elastic Graphics está en mal estado.

Estadística: Maximum

Umbral recomendado: 0,0

Justificación del umbral: el valor de umbral de 1 indica que falló una comprobación de estado.

Período: 300

Puntos de datos para la alarma: 3

Períodos de evaluación: 3

Operador de comparación: GREATER_THAN_THRESHOLD

Amazon ECS

CPUReservation

Dimensiones: ClusterName

Descripción de la alarma: esta alarma le ayuda a detectar una reserva de CPU elevada en el clúster ECS. Una reserva de CPU elevada puede indicar que el clúster se está quedando sin CPU registradas para la tarea. Para solucionar problemas, puede agregar más capacidad, escalar el clúster o configurar el escalado automático.

Finalidad: la alarma se utiliza para detectar si el número total de unidades de CPU reservadas para las tareas del clúster está por alcanzar el total de unidades de CPU registradas para el clúster. Esto le ayuda a saber cuándo escalar verticalmente el clúster. Alcanzar el total de unidades de CPU para el clúster puede provocar que se agote la CPU para las tareas. Si tiene activado el escalado gestionado de los proveedores de capacidad de EC2 o asoció Fargate a los proveedores de capacidad, no se recomienda esta alarma.

Estadística: Average

Umbral recomendado: 90,0

Justificación del umbral: establezca el umbral de reserva de CPU en 90 %. Como alternativa, puede elegir un valor inferior en función de las características del clúster.

Período: 60

Puntos de datos para la alarma: 5

Períodos de evaluación: 5

Operador de comparación: GREATER_THAN_THRESHOLD

CPUUtilization

Dimensiones: ClusterName, ServiceName

Descripción de la alarma: esta alarma le ayuda a detectar un uso elevado de la CPU del servicio ECS. Si no hay una implementación de ECS en curso, una utilización máxima de la CPU podría indicar un cuello de botella en los recursos o problemas de rendimiento de las aplicaciones. Para solucionar los problemas, puede aumentar el límite de la CPU.

Finalidad: esta alarma se utiliza para detectar un uso elevado de la CPU para el servicio de ECS. Un uso elevado y constante de la CPU puede indicar un cuello de botella en los recursos o problemas de rendimiento de las aplicaciones.

Estadística: Average

Umbral recomendado: 90,0

Justificación del umbral: las métricas del servicio para la utilización de la CPU pueden superar el 100 % de utilización. Sin embargo, recomendamos que supervise la métrica para comprobar si hay un uso elevado de la CPU para evitar que afecte a otros servicios. Establezca el umbral entre el 90 y el 95 %. Recomendamos que actualice las definiciones de las tareas para reflejar el uso real y evitar futuros problemas con otros servicios.

Período: 60

Puntos de datos para la alarma: 5

Períodos de evaluación: 5

Operador de comparación: GREATER_THAN_THRESHOLD

MemoryReservation

Dimensiones: ClusterName

Descripción de la alarma: esta alarma ayuda a detectar una reserva de memoria elevada en el clúster de ECS. Una reserva de memoria elevada puede indicar un cuello de botella de recursos para el clúster. Para solucionar el problema, analice el rendimiento de la tarea de servicio para ver si se puede optimizar el uso de la memoria de la tarea. Además, puede registrar más memoria o configurar el escalado automático.

Finalidad: la alarma se utiliza para detectar si el total de unidades de memoria reservadas para las tareas del clúster está cerca de alcanzar el total de unidades de memoria registradas para el clúster. Esto puede ayudarle a saber cuándo escalar verticalmente el clúster. Alcanzar el total de unidades de memoria para el clúster puede provocar que el clúster no pueda iniciar nuevas tareas. Si activó el escalado gestionado por los proveedores de capacidad de EC2 o asoció Fargate a los proveedores de capacidad, no se recomienda utilizar esta alarma.

Estadística: Average

Umbral recomendado: 90,0

Justificación del umbral: establezca el umbral de reserva de memoria en 90 %. Puede ajustarlo a un valor inferior en función de las características del clúster.

Período: 60

Puntos de datos para la alarma: 5

Períodos de evaluación: 5

Operador de comparación: GREATER_THAN_THRESHOLD

HTTPCode_Target_5XX_Count

Dimensiones: ClusterName, ServiceName

Descripción de la alarma: esta alarma ayuda a detectar un recuento elevado de errores del lado del servidor en el servicio de ECS. Esto puede indicar que hay errores que hacen que el servidor no pueda atender las solicitudes. Para solucionar el problema, consulte los registros de la aplicación.

Finalidad: esta alarma se utiliza para detectar un número elevado de errores del lado del servidor en el servicio de ECS.

Estadística: Sum

Umbral recomendado: depende de la situación

Justificación del umbral: calcule un valor de alrededor del 5 % del tráfico promedio y utilice este valor como punto de partida para el umbral. Puede encontrar el tráfico promedio con la métrica `RequestCount`. También puede analizar los datos históricos para determinar la tasa de error aceptable para la carga de trabajo de la aplicación y, a continuación, ajustar el umbral en consecuencia. Los errores 5XX que se producen con frecuencia deben ser objeto de alarma. Sin embargo, si se establece un valor muy bajo para el umbral, la alarma puede ser demasiado sensible.

Período: 60

Puntos de datos para la alarma: 5

Períodos de evaluación: 5

Operador de comparación: GREATER_THAN_THRESHOLD

TargetResponseTime

Dimensiones: ClusterName, ServiceName

Descripción de la alarma: esta alarma ayuda a detectar un tiempo de respuesta del destino elevado para las solicitudes de servicio de ECS. Esto puede indicar que hay problemas que hacen que el servicio no pueda atender las solicitudes a tiempo. Para solucionar el problema, compruebe la métrica CPUUtilization para ver si el servicio se está quedando sin CPU o compruebe el uso de la CPU de otros servicios posteriores de los que depende el servicio.

Finalidad: esta alarma se utiliza para detectar un tiempo de respuesta del destino elevado para las solicitudes de servicio de ECS.

Estadística: Average

Umbral recomendado: depende de la situación

Justificación del umbral: el valor del umbral recomendado para esta alarma depende en gran medida del caso de uso. Revise la criticidad y los requisitos del tiempo de respuesta del destino del servicio y analice el comportamiento histórico de esta métrica para determinar los niveles de umbral razonables.

Período: 60

Puntos de datos para la alarma: 5

Períodos de evaluación: 5

Operador de comparación: GREATER_THAN_THRESHOLD

Amazon ECS con Información de contenedores

EphemeralStorageUtilized

Dimensiones: ClusterName, ServiceName

Descripción de la alarma: esta alarma ayuda a detectar el alto nivel de almacenamiento efímero utilizado en el clúster de Fargate. Si el almacenamiento efímero es constantemente alto, puede comprobar el uso del almacenamiento efímero y aumentarlo.

Finalidad: esta alarma se utiliza para detectar un uso elevado de almacenamiento efímero en el clúster de Fargate. El uso constante de un alto nivel de almacenamiento efímero puede indicar que el disco está lleno y provocar una falla en el contenedor.

Estadística: Average

Umbral recomendado: depende de la situación

Justificación del umbral: establezca el umbral en aproximadamente el 90 % del tamaño del almacenamiento efímero. Puede ajustar este valor en función del uso aceptable del almacenamiento efímero del clúster de Fargate. Para algunos sistemas, puede ser normal utilizar un alto volumen constante de almacenamiento efímero, mientras que para otros puede provocar una falla en el contenedor.

Período: 60

Puntos de datos para la alarma: 5

Períodos de evaluación: 5

Operador de comparación: GREATER_THAN_THRESHOLD

RunningTaskCount

Dimensiones: ClusterName, ServiceName

Descripción de la alarma: esta alarma ayuda a detectar un número bajo de tareas en ejecución del servicio ECS. Si el recuento de tareas en ejecución es demasiado bajo, puede indicar que la aplicación no puede gestionar la carga de servicio y esto podría provocar problemas de rendimiento. Si no hay ninguna tarea en ejecución, es posible que el servicio Amazon ECS no esté disponible o que haya problemas de implementación.

Finalidad: esta alarma se utiliza para detectar si el número de tareas en ejecución es demasiado bajo. Un número bajo y constante de tareas en ejecución puede indicar problemas de rendimiento o implementación del servicio ECS.

Estadística: Average

Umbral recomendado: 0,0

Justificación del umbral: puede ajustar el umbral en función del recuento mínimo de tareas en ejecución del servicio ECS. Si el recuento de tareas en ejecución es 0, el servicio Amazon ECS no estará disponible.

Período: 60

Puntos de datos para la alarma: 5

Períodos de evaluación: 5

Operador de comparación: LESS_THAN_OR_EQUAL_TO_THRESHOLD

instance_filesystem_utilization

Dimensiones: InstanceId, ContainerInstanceId, ClusterName

Descripción de la alarma: esta alarma ayuda a detectar un uso elevado del sistema de archivos en el clúster de ECS. Si el uso del sistema de archivos es constantemente alto, verifique el uso del disco.

Finalidad: esta alarma se utiliza para detectar un uso elevado del sistema de archivos en el clúster de Amazon ECS. Un uso elevado y constante del sistema de archivos puede indicar un cuello de botella en los recursos o problemas de rendimiento de las aplicaciones y puede impedir la ejecución de nuevas tareas.

Estadística: Average

Umbral recomendado: 90,0

Justificación del umbral: puede establecer el umbral de utilización del sistema de archivos entre el 90 y el 95 %. Puede ajustar este valor en función del nivel de capacidad aceptable del sistema de archivos del clúster de Amazon ECS. Para algunos sistemas, una utilización elevada y constante del sistema de archivos puede ser normal y no indicar ningún problema, mientras que para otros puede ser motivo de preocupación y provocar problemas de rendimiento e impedir la ejecución de nuevas tareas.

Período: 60

Puntos de datos para la alarma: 5

Períodos de evaluación: 5

Operador de comparación: GREATER_THAN_THRESHOLD

Amazon EFS

PercentIOLimit

Dimensiones: FileSystemId

Descripción de la alarma: esta alarma ayuda a garantizar que la carga de trabajo se mantenga dentro del límite de E/S disponible para el sistema de archivos. Si la métrica alcanza su límite de E/S con frecuencia, considere la posibilidad de trasladar la aplicación a un sistema de archivos que utilice el rendimiento máximo de E/S como modo. Para solucionar problemas, compruebe los clientes que están conectados al sistema de archivos y las aplicaciones de los clientes que limitan el sistema de archivos.

Finalidad: esta alarma se utiliza para detectar qué tan cerca está un sistema de archivos para llegar al límite de E/S del modo de rendimiento de uso general. Un porcentaje de E/S elevado y constante puede ser un indicador de que el sistema de archivos no puede escalarse lo suficiente con respecto a las solicitudes de E/S y el sistema de archivos puede ser un cuello de botella de recursos para las aplicaciones que utilizan el sistema de archivos.

Estadística: Average

Umbral recomendado: 100,0

Justificación del umbral: cuando el sistema de archivos alcanza su límite de E/S, es posible que responda más lento a las solicitudes de lectura y escritura. Por lo tanto, se recomienda supervisar la métrica para evitar que afecte a las aplicaciones que utilizan el sistema de archivos. El umbral

se puede establecer en torno al 100 %. Sin embargo, este valor se puede ajustar a un valor inferior en función de las características del sistema de archivos.

Período: 60

Puntos de datos para la alarma: 15

Períodos de evaluación: 15

Operador de comparación: GREATER_THAN_OR_EQUAL_TO_THRESHOLD

BurstCreditBalance

Dimensiones: FileSystemId

Descripción de la alarma: esta alarma ayuda a garantizar que haya saldo de créditos de ráfaga disponible para el uso del sistema de archivos. Cuando no haya créditos de ráfaga disponibles, el acceso de las aplicaciones al sistema de archivos se verá limitado debido al bajo rendimiento. Si la métrica cae a 0 de forma constante, considere la posibilidad de cambiar del modo de rendimiento al modo de rendimiento elástico o aprovisionado.

Finalidad: esta alarma se utiliza para detectar un saldo de crédito de ráfaga bajo del sistema de archivos. Un saldo de créditos de ráfaga bajo y constante puede ser un indicador de la ralentización del rendimiento y del aumento de la latencia de E/S.

Estadística: Average

Umbral recomendado: 0,0

Justificación del umbral: cuando el sistema de archivos se queda sin créditos de ráfaga e incluso si la tasa de rendimiento de referencia es inferior, EFS proporciona un rendimiento medido de 1 MiBps a todos los sistemas de archivos. Sin embargo, se recomienda supervisar la métrica para comprobar si el saldo de créditos por ráfaga es bajo para evitar que el sistema de archivos actúe como un cuello de botella para los recursos de las aplicaciones. El umbral se puede establecer en torno a 0 bytes.

Período: 60

Puntos de datos para la alarma: 15

Períodos de evaluación: 15

Operador de comparación: LESS_THAN_OR_EQUAL_TO_THRESHOLD

Amazon EKS con Información de contenedores

node_cpu_utilization

Dimensiones: ClusterName

Descripción de la alarma: esta alarma ayuda a detectar un uso elevado de la CPU en los nodos de trabajo del clúster de EKS. Si la utilización es elevada de forma constante, podría indicar la necesidad de reemplazar los nodos de trabajo por instancias que tengan mayor CPU o la necesidad de escalar horizontalmente el sistema.

Finalidad: esta alarma ayuda a supervisar el uso de la CPU de los nodos de trabajo del clúster de EKS para que el rendimiento del sistema no se degrade.

Estadística: Maximum

Umbral recomendado: 80,0

Justificación del umbral: se recomienda establecer el umbral en un valor inferior o igual al 80 % para disponer del tiempo suficiente para depurar el problema antes de que el sistema empiece a verse afectado.

Período: 60

Puntos de datos para la alarma: 5

Períodos de evaluación: 5

Operador de comparación: GREATER_THAN_THRESHOLD

node_filesystem_utilization

Dimensiones: ClusterName

Descripción de la alarma: esta alarma ayuda a detectar un uso elevado del sistema de archivos en los nodos de trabajo del clúster de EKS. Si la utilización es elevada de forma constante, es posible que necesite actualizar los nodos de trabajo para que tengan un mayor volumen de disco o que necesite escalarlos horizontalmente.

Finalidad: esta alarma ayuda a supervisar la utilización del sistema de archivos de los nodos de trabajo del clúster de EKS. Si la utilización alcanza el 100 %, se pueden producir fallos en la aplicación, cuellos de botella de E/S del disco, la expulsión del pod o que el nodo deje de responder por completo.

Estadística: Maximum

Umbral recomendado: depende de la situación

Justificación del umbral: si hay suficiente presión en el disco (lo que significa que el disco se está llenando), los nodos se marcan como en mal estado y los pods se expulsan del nodo. Los pods de un nodo con presión de disco se expulsan cuando el sistema de archivos disponible es inferior a los umbrales de expulsión establecidos en kubelet. Establezca el umbral de alarma para tener tiempo suficiente para reaccionar antes de que el nodo sea expulsado del clúster.

Período: 60

Puntos de datos para la alarma: 5

Períodos de evaluación: 5

Operador de comparación: GREATER_THAN_THRESHOLD

node_memory_utilization

Dimensiones: ClusterName

Descripción de la alarma: esta alarma ayuda a detectar un uso elevado de la memoria en los nodos de trabajo del clúster de EKS. Si la utilización es elevada de forma constante, podría indicar la necesidad de aumentar el número de réplicas de los pods u optimizar la aplicación.

Finalidad: esta alarma ayuda a supervisar el uso de la memoria de los nodos de trabajo del clúster de EKS para que el rendimiento del sistema no se degrade.

Estadística: Maximum

Umbral recomendado: 80,0

Justificación del umbral: se recomienda establecer el umbral en un valor inferior o igual al 80 % para disponer de tiempo suficiente para depurar el problema antes de que el sistema empiece a verse afectado.

Período: 60

Puntos de datos para la alarma: 5

Períodos de evaluación: 5

Operador de comparación: GREATER_THAN_THRESHOLD

pod_cpu_utilization_over_pod_limit

Dimensiones: ClusterName, Namespace, Service

Descripción de la alarma: esta alarma ayuda a detectar un uso elevado de la CPU en los pods del clúster de EKS. Si la utilización es siempre alta, podría indicar la necesidad de aumentar el límite de la CPU del pod afectado.

Finalidad: esta alarma ayuda a supervisar el uso de la CPU de los pods que pertenecen a un servicio de Kubernetes en el clúster de EKS, de modo que se puede identificar con rapidez si el pod de un servicio consume más CPU de lo esperado.

Estadística: Maximum

Umbral recomendado: 80,0

Justificación del umbral: se recomienda establecer el umbral en un valor inferior o igual al 80 % para disponer de tiempo suficiente para depurar el problema antes de que el sistema empiece a verse afectado.

Período: 60

Puntos de datos para la alarma: 5

Períodos de evaluación: 5

Operador de comparación: GREATER_THAN_THRESHOLD

pod_memory_utilization_over_pod_limit

Dimensiones: ClusterName, Namespace, Service

Descripción de la alarma: esta alarma ayuda a detectar un uso elevado de la memoria en los pods del clúster de EKS. Si la utilización es siempre alta, podría indicar la necesidad de aumentar el límite de memoria del pod afectado.

Finalidad: esta alarma ayuda a supervisar el uso de la memoria de los pods del clúster de EKS para que el rendimiento del sistema no se degrade.

Estadística: Maximum

Umbral recomendado: 80,0

Justificación del umbral: se recomienda establecer el umbral en un valor inferior o igual al 80 % para disponer de tiempo suficiente para depurar el problema antes de que el sistema empiece a verse afectado.

Período: 60

Puntos de datos para la alarma: 5

Períodos de evaluación: 5

Operador de comparación: GREATER_THAN_THRESHOLD

Amazon Kinesis Data Streams

GetRecords.IteratorAgeMilliseconds

Dimensiones: StreamName

Descripción de la alarma: esta alarma puede detectar si la antigüedad máxima del iterador es demasiado alta. Para aplicaciones de procesamiento de datos en tiempo real, configure la retención de datos de acuerdo con la tolerancia al retraso. Esto proceso suele tomar solo algunos minutos. En el caso de las aplicaciones que procesan datos históricos, utilice esta métrica para supervisar la velocidad de recuperación. Una solución rápida para detener la pérdida de datos consiste en incrementar el período de retención mientras se soluciona el problema. También, puede aumentar el número de trabajadores que procesan los registros en su aplicación para consumidores. Las causas más comunes del incremento gradual en la antigüedad del iterador son la insuficiencia de recursos físicos o una lógica de procesamiento de registros que no se escaló con un aumento en el rendimiento del flujo. Consulte este enlace para obtener más detalles.

Intención: esta alarma se utiliza para detectar si los datos de su flujo van a caducar porque se conservaron durante demasiado tiempo o porque el procesamiento de los registros es demasiado lento. Esta alarma ayuda a evitar la pérdida de datos tras alcanzar el 100 % del tiempo de retención del flujo.

Estadística: Maximum

Umbral recomendado: depende de la situación

Justificación del umbral: el valor de umbral recomendado para esta alarma depende en gran medida del período de retención del flujo y de la tolerancia al retraso en el procesamiento de los

registros. Revise sus requisitos y analice las tendencias históricas y, a continuación, establezca el umbral en la cantidad de milisegundos que representa un retraso de procesamiento crítico. Si la antigüedad de un iterador supera el 50 % del periodo de retención (con un valor predeterminado de 24 horas, pero configurable hasta 365 días), existe el riesgo de pérdida de datos debido al vencimiento del registro. Puede supervisar la métrica para asegurarse de que ninguna de sus particiones nunca se acerque a este límite.

Período: 60

Puntos de datos para la alarma: 15

Períodos de evaluación: 15

Operador de comparación: GREATER_THAN_THRESHOLD

GetRecords.Success

Dimensiones: StreamName

Descripción de la alarma: esta métrica aumenta cada vez que los consumidores leen de forma correcta los datos del flujo. `GetRecords` no devuelve ningún dato cuando genera una excepción. La excepción más común es `ProvisionedThroughputExceededException`, debido a que la tasa de solicitudes del flujo es demasiado alta o a que el rendimiento disponible ya está ocupado para el segundo en cuestión. Reduzca la frecuencia o el tamaño de las solicitudes. Para obtener más información, consulte Límites en la Guía para desarrolladores de Amazon Kinesis Data Streams, y Reintentos de error y retroceso exponencial en AWS.

Finalidad: esta alarma puede detectar si los consumidores tienen problemas para recuperar los registros del flujo. Al configurar una alarma en esta métrica, puede detectar de forma proactiva cualquier problema relacionado con el consumo de datos, como el aumento de las tasas de error o la disminución del número de recuperaciones exitosas. Esto le permite tomar medidas oportunas para resolver posibles problemas y mantener una canalización de procesamiento de datos sin inconvenientes.

Estadística: Average

Umbral recomendado: depende de la situación

Justificación del umbral: en función de la importancia de recuperar los registros del flujo, establezca el umbral en basándose en la tolerancia de la aplicación a los registros fallidos. El umbral debe ser el porcentaje correspondiente de operaciones exitosas. Puede utilizar los datos métricos históricos de GetRecords como referencia para determinar la tasa de errores aceptable.

También debe tener en cuenta los reintentos al establecer el umbral, ya que los registros fallidos se pueden volver a intentar. Esto ayuda a evitar que los picos transitorios generen alertas innecesarias.

Período: 60

Puntos de datos para la alarma: 5

Períodos de evaluación: 5

Operador de comparación: LESS_THAN_THRESHOLD

PutRecord.Success

Dimensiones: StreamName

Descripción de la alarma: esta alarma detecta cuando el número de operaciones de `PutRecord` fallidas supera el umbral. Investigue los registros del productor de datos para encontrar las causas principales de las fallas. La razón más común es un rendimiento aprovisionado insuficiente en la partición que provocó el `ProvisionedThroughputExceededException`. Esto se debe a que la tasa de solicitudes para el flujo es demasiado alta o a que el rendimiento que se intentó incorporar en la partición es demasiado alto. Reduzca la frecuencia o el tamaño de las solicitudes. Para obtener más información, consulte Límites y Reintentos de error y retroceso exponencial en AWS.

Finalidad: esta alarma puede detectar fallas en la incorporación de registros en el flujo. Le ayuda a identificar problemas al escribir datos en el flujo. Al configurar una alarma en esta métrica, puede detectar de forma proactiva cualquier problema que puedan tener los productores a la hora de publicar datos en el flujo, como el aumento de las tasas de error o la disminución del número de registros que se publican con éxito. Esto le permite tomar medidas oportunas para abordar posibles problemas y mantener un proceso fiable de ingesta de datos.

Estadística: Average

Umbral recomendado: depende de la situación

Justificación del umbral: en función de la importancia del procesamiento y la ingesta de datos para su servicio, establezca el umbral en función de la tolerancia de su aplicación a los registros fallidos. El umbral debe ser el porcentaje correspondiente de operaciones exitosas. Puede utilizar los datos métricos históricos de PutRecord como referencia para determinar la tasa de fallos aceptable. También debe tener en cuenta los reintentos al establecer el umbral, ya que los registros fallidos se pueden volver a intentar.

Período: 60

Puntos de datos para la alarma: 5

Períodos de evaluación: 5

Operador de comparación: LESS_THAN_THRESHOLD

PutRecords.FailedRecords

Dimensiones: StreamName

Descripción de la alarma: esta alarma detecta cuando el número de `PutRecords` fallidos supera el umbral. Kinesis Data Streams intenta procesar todos los registros de cada solicitud de `PutRecords`, pero un único error en el registro no detiene el procesamiento de los registros posteriores. El motivo principal de estos errores es que se supera el rendimiento de un flujo o una partición individual. Las causas más comunes son los picos de tráfico y las latencias de la red, que hacen que los registros lleguen al flujo de manera desigual. Debe detectar los registros procesados de forma incorrecta y reintentarlos en una llamada posterior. Consulte Handling Failures When Using PutRecords para obtener más información.

Finalidad: esta alarma puede detectar errores constantes cuando se utiliza la operación por lotes para colocar registros en el flujo. Al configurar una alarma en esta métrica, puede detectar de forma proactiva un aumento en el número de registros fallidos, lo que le permitirá tomar medidas oportunas para abordar los problemas subyacentes y garantizar un proceso de ingesta de datos fiable y fluido.

Estadística: Sum

Umbral recomendado: depende de la situación

Justificación del umbral: defina el umbral en el número de registros fallidos que refleje la tolerancia de la aplicación frente a los registros fallidos. Puede utilizar los datos históricos como referencia para determinar el valor de errores aceptable. También debe tener en cuenta los reintentos al establecer el umbral, ya que los registros fallidos se pueden volver a intentar en llamadas posteriores a PutRecords.

Período: 60

Puntos de datos para la alarma: 5

Períodos de evaluación: 5

Operador de comparación: GREATER_THAN_THRESHOLD

ReadProvisionedThroughputExceeded

Dimensiones: StreamName

Descripción de la alarma: la alarma hace un seguimiento de la cantidad de registros que provocan una limitación de la capacidad de rendimiento de lectura. Si observa una limitación constante, debería considerar la posibilidad de agregar más particiones al flujo para aumentar el rendimiento de lectura aprovisionado. Si hay más de una aplicación para consumidores que se ejecuta en el flujo y estas comparten el límite `GetRecords`, recomendamos que registre las nuevas aplicaciones para consumidores mediante Enhanced Fan-Out. Si al agregar más particiones no se reduce el número de limitaciones, es posible que se trate de una partición “activa” de la que se está leyendo más que desde otras particiones. Habilite la supervisión mejorada, busque la partición “activa” y divídala.

Finalidad: esta alarma puede detectar si los consumidores se ven limitados al superar el rendimiento de lectura aprovisionado (determinado por el número de particiones de las que dispone). En ese caso, no podrá leer desde el flujo y este puede empezar a hacer copias de seguridad.

Estadística: Average

Umbral recomendado: depende de la situación

Justificación del umbral: por lo general, las solicitudes limitadas se pueden volver a intentar y, por lo tanto, establecer el umbral en cero hace que la alarma sea demasiado sensible. Sin embargo, una limitación constante puede afectar a la lectura desde el flujo y debería activar la alarma. Establezca el umbral en un porcentaje en función de las solicitudes limitadas de la aplicación y de las configuraciones de reintento.

Período: 60

Puntos de datos para la alarma: 5

Períodos de evaluación: 5

Operador de comparación: GREATER_THAN_THRESHOLD

SubscribeToShardEvent.MillisBehindLatest

Dimensiones: StreamName, ConsumerName

Descripción de la alarma: esta alarma detecta cuando el retraso en el procesamiento de los registros en la aplicación sobrepasa el umbral. Los problemas transitorios, como los fallos en el funcionamiento de la API en una aplicación posterior, pueden provocar un aumento repentino en la métrica. Si estos fallos se producen de forma constante, debería investigar. Una causa común es que el consumidor no procesa los registros con la suficiente rapidez debido a la insuficiencia de los recursos físicos o de la lógica de procesamiento de registros, que no se escala con el aumento del rendimiento del flujo. El bloqueo de las llamadas en una ruta crítica suele ser la causa de la ralentización en el procesamiento de registros. Puede aumentar el paralelismo si incrementa el número de particiones. También, debe confirmar que los nodos de procesamiento subyacentes tengan recursos físicos suficientes durante los picos de demanda.

Finalidad: esta alarma puede detectar un retraso en la suscripción al evento de una partición del flujo. Esto indica un retraso en el procesamiento y puede ayudar a identificar posibles problemas con el rendimiento de la aplicación para consumidores o con el estado general del flujo. Cuando el retraso en el procesamiento es significativo, debe investigar y abordar cualquier cuello de botella o ineficiencia en las aplicaciones del consumidor para garantizar el procesamiento de los datos en tiempo real y minimizar la acumulación de datos.

Estadística: Average

Umbral recomendado: depende de la situación

Justificación del umbral: el valor del umbral recomendado para esta alarma depende en gran medida del retraso que pueda tolerar su aplicación. Revise los requisitos de su solicitud y analice las tendencias históricas y, a continuación, seleccione el umbral correspondiente. Cuando la llamada SubscribeToShard se realiza de forma correcta, el consumidor recibirá eventos de SubscribeToShardEvent a través de la conexión persistente durante un máximo de 5 minutos, tras lo cual tendrá que volver a llamar a SubscribeToShard para renovar la suscripción si quiere continuar con la recepción de registros.

Período: 60

Puntos de datos para la alarma: 5

Períodos de evaluación: 5

Operador de comparación: GREATER_THAN_THRESHOLD

WriteProvisionedThroughputExceeded

Dimensiones: StreamName

Descripción de la alarma: esta alarma detecta cuando el número de registros que dan lugar a una limitación de la capacidad de rendimiento de escritura alcanza el umbral. Cuando los productores superan el rendimiento de escritura aprovisionado (determinado por el número de particiones que posee), se verán limitados y no podrán incluir los registros en el flujo. Para evitar una limitación constante, debería considerar la posibilidad de agregar particiones al flujo. Esto aumenta el rendimiento de escritura aprovisionado y evita futuras limitaciones. También debe tener en cuenta la elección de la clave de partición al incorporar los registros. Se prefiere la clave de partición aleatoria porque distribuye los registros de manera uniforme entre las particiones del flujo, siempre que sea posible.

Finalidad: esta alarma puede detectar si sus productores son rechazados a la hora de escribir registros debido a la limitación del flujo o la partición. Si el flujo está en modo aprovisionado, configurar esta alarma le ayudará a tomar medidas de forma proactiva cuando el flujo de datos alcance sus límites, lo que le permitirá optimizar la capacidad aprovisionada o tomar las medidas de escalado adecuadas para evitar la pérdida de datos y mantener un procesamiento de datos fluido.

Estadística: Average

Umbral recomendado: depende de la situación

Justificación del umbral: por lo general, las solicitudes limitadas se pueden volver a intentar, por lo que, si se establece el umbral en cero, la alarma será demasiado sensible. Sin embargo, una limitación constante puede afectar a la escritura en el flujo, por lo que debe configurar el umbral de alarma para detectar este problema. Establezca el umbral en un porcentaje en función de las solicitudes limitadas de la aplicación y de las configuraciones de reintento.

Período: 60

Puntos de datos para la alarma: 5

Períodos de evaluación: 5

Operador de comparación: GREATER_THAN_THRESHOLD

Lambda

ClaimedAccountConcurrency

Dimensiones: None

Descripción de la alarma: esta alarma ayuda a supervisar si la simultaneidad de la función de Lambda se acerca al límite de simultaneidad a nivel de la región de su cuenta. Una función comienza a limitarse si alcanza el límite de simultaneidad. Puede realizar las siguientes acciones para evitar la limitación.

1. Solicitar un aumento de simultaneidad en esta región.
2. Identifique y reduzca cualquier simultaneidad reservada o aprovisionada que no se utilice.
3. Identificar los problemas de funcionamiento en las funciones para mejorar la velocidad de procesamiento y, por lo tanto, mejorar el rendimiento.
4. Aumentar el tamaño del lote de las funciones, de modo que cada invocación de la función procese más mensajes.

Finalidad: esta alarma puede detectar de forma proactiva si la simultaneidad de su función de Lambda se acerca a la cuota de simultaneidad de la cuenta a nivel regional, para que pueda actuar en consecuencia. Una función se limita si `ClaimedAccountConcurrency` alcanza la cuota de simultaneidad de la cuenta a nivel de la región. Si utiliza la simultaneidad reservada (RC) o la simultaneidad aprovisionada (PC), esta alarma le ofrece más visibilidad del uso de la simultaneidad que una alarma activada en `ConcurrentExecutions`.

Estadística: Maximum

Umbral recomendado: depende de la situación

Justificación del umbral: debe calcular el valor de alrededor del 90 % de la cuota de simultaneidad configurada para la cuenta en la región y utilizar el resultado como valor para el umbral. De forma predeterminada, la cuenta tiene una cuota de simultaneidad de 1000 en todas las funciones en una región. Sin embargo, debe comprobar la cuota de su cuenta desde el panel de control de Service Quotas.

Período: 60

Puntos de datos para la alarma: 10

Períodos de evaluación: 10

Operador de comparación: GREATER_THAN_THRESHOLD

Errores

Dimensiones: FunctionName

Descripción de la alarma: esta alarma detecta un alto número de errores. Los errores incluyen las excepciones lanzadas por el código y las excepciones lanzadas por el tiempo de ejecución de Lambda. Puede consultar los registros relacionados con la función para diagnosticar el problema.

Finalidad: la alarma ayuda a detectar un alto número de errores en las invocaciones de funciones.

Estadística: Sum

Umbral recomendado: depende de la situación

Justificación del umbral: establezca el umbral en un número mayor a cero. El valor exacto puede depender de la tolerancia a los errores de la aplicación. Es fundamental comprender la importancia de las invocaciones que maneja la función. Para algunas aplicaciones, cualquier error puede ser inaceptable, mientras que otras aplicaciones permiten un cierto margen de error.

Período: 60

Puntos de datos para la alarma: 3

Períodos de evaluación: 3

Operador de comparación: GREATER_THAN_THRESHOLD

Limitaciones

Dimensiones: FunctionName

Descripción de la alarma: esta alarma detecta un número elevado de solicitudes de invocación limitadas. La limitación ocurre cuando no hay ninguna simultaneidad disponible para escalar verticalmente. Existen varios enfoques para resolver este problema. 1) Solicitar un aumento de simultaneidad de AWS Support en esta región. 2) Identificar los problemas de rendimiento en la función para mejorar la velocidad de procesamiento y, por lo tanto, mejorar el rendimiento. 3) Aumentar el tamaño del lote de la función, de modo que cada invocación de la función procese más mensajes.

Finalidad: la alarma ayuda a detectar un número elevado de solicitudes de invocación limitadas para una función de Lambda. Es importante saber si las solicitudes se rechazan con frecuencia debido a la limitación y si necesita mejorar el rendimiento de la función de Lambda o aumentar la capacidad de simultaneidad para evitar una limitación constante.

Estadística: Sum

Umbral recomendado: depende de la situación

Justificación del umbral: establezca el umbral en un número mayor a cero. El valor exacto del umbral puede depender de la tolerancia de la aplicación. Establezca el umbral de acuerdo con los requisitos de uso y escalado de la función.

Período: 60

Puntos de datos para la alarma: 5

Períodos de evaluación: 5

Operador de comparación: GREATER_THAN_OR_EQUAL_TO_THRESHOLD

Duration (Duración)

Dimensiones: FunctionName

Descripción de la alarma: esta alarma detecta tiempos de duración prolongados para procesar un evento mediante una función de Lambda. Las duraciones prolongadas pueden deberse a cambios en el código de la función, que hacen que la función tarde más en ejecutarse, o a que las dependencias de la función demoren más.

Intención: esta alarma puede detectar una duración prolongada de ejecución de una función de Lambda. Una duración prolongada del tiempo de ejecución indica que la invocación de una función tarda más tiempo y, también, puede afectar a la capacidad de simultaneidad de la invocación si Lambda gestiona un número mayor de eventos. Es fundamental saber si la función de Lambda tarda con frecuencia más tiempo de ejecución de lo esperado.

Estadística: p90

Umbral recomendado: depende de la situación

Justificación del umbral: el umbral de la duración depende de la aplicación y de las cargas de trabajo, así como de los requisitos de rendimiento. Para los requisitos de alto rendimiento, establezca el umbral en un tiempo más corto para comprobar si la función cumple con las expectativas. También, puede analizar los datos históricos de las métricas de duración para comprobar si el tiempo empleado coincide con las expectativas de rendimiento de la función y, a continuación, establecer el umbral en un tiempo superior al promedio histórico. Asegúrese de establecer el umbral por debajo del tiempo de espera de la función configurada.

Período: 60

Puntos de datos para la alarma: 15

Períodos de evaluación: 15

Operador de comparación: GREATER_THAN_THRESHOLD

ConcurrentExecutions

Dimensiones: FunctionName

Descripción de la alarma: esta alarma ayuda a supervisar si la simultaneidad de la función se acerca al límite de simultaneidad a nivel de la región de su cuenta. Una función comienza a limitarse si alcanza el límite de simultaneidad. Puede realizar las siguientes acciones para evitar la limitación.

1. Solicitar un aumento de simultaneidad en esta región.
2. Identificar los problemas de funcionamiento en las funciones para mejorar la velocidad de procesamiento y, por lo tanto, mejorar el rendimiento.
3. Aumentar el tamaño del lote de las funciones, de modo que cada invocación de la función procese más mensajes.

Para obtener una mejor visibilidad de la simultaneidad reservada y del uso de la simultaneidad aprovisionada, configura una alarma en la nueva métrica de `ClaimedAccountConcurrency`.

Finalidad: esta alarma puede detectar de forma proactiva si la simultaneidad de la función se acerca a la cuota de simultaneidad de la cuenta a nivel regional, para que pueda actuar en consecuencia. Una función se limita si alcanza la cuota de simultaneidad de la cuenta a nivel de la región.

Estadística: Maximum

Umbral recomendado: depende de la situación

Justificación del umbral: establezca el umbral en alrededor del 90 % de la cuota de simultaneidad establecida para la cuenta en la región. De forma predeterminada, la cuenta tiene una cuota de simultaneidad de 1000 en todas las funciones en una región. Sin embargo, puede comprobar la cuota de tu cuenta, ya que se puede aumentar si se pone en contacto con el servicio de asistencia de AWS.

Período: 60

Puntos de datos para la alarma: 10

Períodos de evaluación: 10

Operador de comparación: GREATER_THAN_THRESHOLD

Lambda Insights

Recomendamos configurar alarmas de las prácticas recomendadas para las siguientes métricas de Lambda Insights.

memory_utilization

Dimensiones: function_name

Descripción de la alarma: esta alarma se utiliza para detectar si la utilización de la memoria de una función de lambda se acerca al límite configurado. Para solucionar problemas, puede intentar 1) Optimizar su código. 2) Calcular de forma correcta su asignación de memoria al estimar con precisión los requisitos de memoria. Puede consultar Lambda Power Tuning para más información. 3) Utilizar la agrupación de conexiones. Consulte Using Amazon RDS Proxy with Lambda para ver la agrupación de conexiones para la base de datos de RDS. 4) También, puede considerar diseñar sus funciones para evitar almacenar grandes cantidades de datos en la memoria entre las invocaciones.

Finalidad: esta alarma sirve para detectar si la utilización de memoria para la función de Lambda se acerca al límite configurado.

Estadística: Average

Umbral recomendado: 90,0

Justificación del umbral: establezca el umbral en el 90 % para recibir una alerta cuando la utilización de la memoria supere el 90 % de la memoria asignada. Puede ajustarlo a un valor inferior si le preocupa la carga de trabajo relacionada para el uso de la memoria. También puede comprobar los datos históricos de esta métrica y establecer el umbral en consecuencia.

Período: 60

Puntos de datos para la alarma: 10

Períodos de evaluación: 10

Operador de comparación: GREATER_THAN_THRESHOLD

Amazon VPC (**`AWS/NATGateway`**)

ErrorPortAllocation

Dimensiones: NatGatewayId

Descripción de la alarma: esta alarma ayuda a detectar cuándo la puerta de enlace NAT no puede asignar puertos a nuevas conexiones. Para resolver este problema, consulte Resolve port allocation errors on NAT Gateway.

Finalidad: esta alarma se utiliza para detectar si la puerta de enlace NAT no pudo asignar un puerto de origen.

Estadística: Sum

Umbral recomendado: 0,0

Justificación del umbral: si el valor de ErrorPortAllocation es superior a cero, significa que hay demasiadas conexiones simultáneas a un único destino popular abiertas a través de la puerta de enlace NAT.

Período: 60

Puntos de datos para la alarma: 15

Períodos de evaluación: 15

Operador de comparación: GREATER_THAN_THRESHOLD

PacketsDropCount

Dimensiones: NatGatewayId

Descripción de la alarma: esta alarma ayuda a detectar cuándo la puerta de enlace NAT descarta paquetes. Esto puede deberse a un problema con la puerta de enlace NAT, así que consulte el panel de estado del servicio de AWS para ver el estado de la puerta de enlace NAT de AWS en su región. Esto puede ayudarlo a correlacionar el problema de red relacionado con el tráfico mediante la puerta de enlace NAT.

Finalidad: esta alarma se utiliza para detectar si la puerta de enlace NAT descarta paquetes.

Estadística: Sum

Umbral recomendado: depende de la situación

Justificación del umbral: debe calcular el valor del 0,01 por ciento del tráfico total en la puerta de enlace NAT y utilizar ese resultado como valor del umbral. Utilice los datos históricos del tráfico en la puerta de enlace NAT para determinar el umbral.

Período: 60

Puntos de datos para la alarma: 5

Períodos de evaluación: 5

Operador de comparación: GREATER_THAN_THRESHOLD

Enlace privado de AWS (**`AWS/PrivateLinkEndpoints`**)

PacketsDropped

Dimensiones: VPC Id, VPC Endpoint Id, Endpoint Type, Subnet Id, Service Name

Descripción de la alarma: esta alarma ayuda a detectar si el punto de conexión o el servicio del punto de conexión no funcionan de forma correcta, ya que supervisa la cantidad de paquetes descartados por el punto de conexión. Tenga en cuenta que se descartan los paquetes con un tamaño superior a 8500 bytes que llegan al punto de conexión de VPC. Para solucionar problemas, consulte ¿Cómo soluciono los problemas de conectividad entre un punto de conexión de Amazon VPC de interfaz y un servicio de punto de conexión?.

Finalidad: esta alarma se utiliza para detectar si el punto de conexión o el servicio del punto de conexión no están en buen estado.

Estadística: Sum

Umbral recomendado: depende de la situación

Justificación del umbral: establezca el umbral según el caso de uso. Si quiere saber si el punto de conexión o el servicio del punto de conexión están en mal estado, debe establecer un umbral bajo para poder solucionar el problema antes de que se produzca una pérdida importante de datos. Puede utilizar los datos históricos para comprender la tolerancia ante el descarte de paquetes y establecer el umbral en consecuencia.

Período: 60

Puntos de datos para la alarma: 5

Períodos de evaluación: 5

Operador de comparación: GREATER_THAN_THRESHOLD

Enlace privado de AWS (**AWS/PrivateLinkServices**)

RstPacketsSent

Dimensiones: Service Id, Load Balancer Arn, Az

Descripción de la alarma: esta alarma ayuda a detectar los destinos de un servicio de punto de conexión en mal estado en función de la cantidad de paquetes de restablecimiento que se envían a los puntos finales. Al depurar los errores de conexión con un consumidor de su servicio, puede validar si el servicio está restableciendo las conexiones con la métrica RstPacketsSent o si hay algún otro error en la ruta de la red.

Finalidad: esta alarma se utiliza para detectar si los destinos de un servicio de punto de conexión están en mal estado.

Estadística: Sum

Umbral recomendado: depende de la situación

Justificación del umbral: el umbral depende del caso de uso. Si su caso de uso puede tolerar que los destinos no estén en buen estado, puede establecer el umbral en un nivel alto. Si el caso de uso no tolera los destinos en mal estado, puede establecer el umbral en un nivel muy bajo.

Período: 60

Puntos de datos para la alarma: 5

Períodos de evaluación: 5

Operador de comparación: GREATER_THAN_THRESHOLD

Amazon RDS

CPUUtilization

Dimensiones: DBInstanceIdentifier

Descripción de la alarma: esta alarma ayuda a supervisar un uso elevado y constante de la CPU. La utilización de la CPU mide el tiempo de inactividad. Considere la posibilidad de utilizar Supervisión mejorada o Información del rendimiento para revisar qué tiempo de espera consume la mayor parte del tiempo de la CPU (`guest`, `irq`, `wait`, `nice`, etc.) en MariaDB, MySQL, Oracle y PostgreSQL. A continuación, evalúe qué consultas consumen la mayor cantidad de CPU. Si no puede ajustar la carga de trabajo, considere la posibilidad de cambiar a una clase de instancia de base de datos más grande.

Finalidad: esta alarma se utiliza para detectar un uso elevado y constante de la CPU con el fin de evitar tiempos de respuesta y tiempos de espera muy elevados. Si desea comprobar la microrráfaga de uso de la CPU, puede configurar un tiempo de evaluación de la alarma más bajo.

Estadística: Average

Umbral recomendado: 90,0

Justificación del umbral: es posible que los picos aleatorios en el consumo de CPU no obstaculicen el rendimiento de la base de datos, pero un nivel prolongado de CPU puede dificultar las próximas solicitudes de la base de datos. Según la carga de trabajo general de la base de datos, un nivel elevado de CPU en la instancia de RDS/Aurora puede reducir el rendimiento general.

Período: 60

Puntos de datos para la alarma: 5

Períodos de evaluación: 5

Operador de comparación: GREATER_THAN_THRESHOLD

DatabaseConnections

Dimensiones: DBInstanceIdentifier

Descripción de la alarma: esta alarma detecta un número elevado de conexiones. Revise las conexiones existentes y cancele las que estén en estado de “reposo” o que estén cerradas incorrectamente. Considere la posibilidad de utilizar la agrupación de conexiones para limitar el número de conexiones nuevas. También puede aumentar el tamaño de la instancia de base de datos para usar una clase con más memoria y, por lo tanto, un valor predeterminado más alto para “max_connections” o aumentar el valor “max_connections” en RDS y Aurora MySQL y PostgreSQL para la clase actual si puede soportar la carga de trabajo.

Finalidad: esta alarma se utiliza para evitar el rechazo de conexiones cuando se alcanza el número máximo de conexiones de base de datos. No se recomienda esta alarma si cambia con frecuencia la clase de instancia de base de datos, ya que al hacerlo se modifican la memoria y el número máximo predeterminado de conexiones.

Estadística: Average

Umbral recomendado: depende de la situación

Justificación del umbral: el número de conexiones permitidas depende del tamaño de la clase de instancia de base de datos y de los parámetros específicos del motor de base de datos relacionados con los procesos o las conexiones. Debe calcular un valor entre el 90 y el 95 % del número máximo de conexiones de la base de datos y utilizar ese resultado como valor del umbral.

Período: 60

Puntos de datos para la alarma: 5

Períodos de evaluación: 5

Operador de comparación: GREATER_THAN_THRESHOLD

EBSByteBalance%

Dimensiones: DBInstanceIdentifier

Descripción de la alarma: esta alarma ayuda a supervisar un bajo porcentaje de los créditos de rendimiento restantes. Para solucionar problemas, consulte problemas de latencia en RDS.

Finalidad: esta alarma se utiliza para detectar un bajo porcentaje de créditos de rendimiento que quedan en el bucket de ráfaga. Un porcentaje de balance de bytes bajo puede provocar problemas de cuello de botella en el rendimiento. Esta alarma no se recomienda para las instancias de Aurora PostgreSQL.

Estadística: Average

Umbral recomendado: 10,0

Justificación del umbral: un saldo de créditos de rendimiento inferior al 10 % se considera deficiente y se debe establecer el umbral en consecuencia. También puede establecer un umbral inferior si la aplicación puede tolerar un rendimiento inferior para la carga de trabajo.

Período: 60

Puntos de datos para la alarma: 3

Períodos de evaluación: 3

Operador de comparación: LESS_THAN_THRESHOLD

EBSIOBalance%

Dimensiones: DBInstanceIdentifier

Descripción de la alarma: esta alarma ayuda a supervisar el bajo porcentaje de créditos de IOPS restantes. Para obtener ayuda con la solución de problemas de latencia, consulte problemas de latencia en RDS.

Finalidad: esta alarma se utiliza para detectar un bajo porcentaje de créditos de E/S que quedan en el bucket de ráfaga. Un porcentaje de saldo de IOPS bajo puede provocar problemas de cuello de botella de IOPS. Esta alarma no se recomienda para las instancias de Aurora.

Estadística: Average

Umbral recomendado: 10,0

Justificación del umbral: un saldo de créditos de IOPS inferior al 10 % se considera deficiente y puede establecer el umbral en consecuencia. También puede establecer un umbral más bajo si la aplicación puede tolerar un menor número de IOPS para la carga de trabajo.

Período: 60

Puntos de datos para la alarma: 3

Períodos de evaluación: 3

Operador de comparación: LESS_THAN_THRESHOLD

FreeableMemory

Dimensiones: DBInstanceIdentifier

Descripción de la alarma: esta alarma ayuda a supervisar la poca memoria que se puede liberar, lo que puede indicar que hay un pico en las conexiones de la base de datos o que la instancia está sometida a una gran presión de memoria. Compruebe la presión de memoria supervisando las métricas de CloudWatch para `SwapUsage` además de `FreeableMemory`. Si el consumo de memoria de instancia es con frecuencia demasiado alto, significa que debe verificar la carga de trabajo o actualizar la clase de instancia. Para una instancia de base de datos de lectura

de Aurora, considere añadir instancias de base de datos de lectura al clúster. Para obtener información acerca de cómo solucionar problemas de Aurora, consulte problemas de memoria que se puede liberar.

Finalidad: esta alarma se utiliza para evitar que se agote la memoria, lo que puede provocar el rechazo de las conexiones.

Estadística: Average

Umbral recomendado: depende de la situación

Justificación del umbral: según la carga de trabajo y la clase de instancia, pueden ser adecuados diferentes valores para el umbral. Lo ideal es que la memoria disponible no sea inferior al 25 % de la memoria total durante períodos prolongados. Para Aurora puede establecer un umbral cercano al 5 %, ya que la métrica que se aproxima a un valor de 0 significa que la instancia de base de datos ha escalado verticalmente todo lo que puede. Puede analizar el comportamiento histórico de esta métrica para determinar los niveles de umbral razonables.

Período: 60

Puntos de datos para la alarma: 15

Períodos de evaluación: 15

Operador de comparación: LESS_THAN_THRESHOLD

FreeLocalStorage

Dimensiones: DBInstanceIdentifier

Descripción de la alarma: esta alarma ayuda a supervisar el bajo nivel de almacenamiento local libre. La edición Aurora compatible con PostgreSQL utiliza el almacenamiento local para almacenar los registros de errores y los archivos temporales. Aurora MySQL utiliza el almacenamiento local para almacenar registros de errores, registros generales, registros de consultas lentas, registros de auditoría y tablas temporales que no son de InnoDB. Estos volúmenes de almacenamiento local están respaldados por Amazon EBS Store y pueden ampliarse utilizando una clase de instancia de base de datos mayor. Para solucionar problemas, compruebe Aurora compatible con PostgreSQL y compatible con MySQL.

Finalidad: esta alarma se utiliza para detectar qué tan cerca está la instancia de base de datos Aurora de alcanzar el límite de almacenamiento local si no se utiliza Aurora sin servidor v2 o una versión superior. El almacenamiento local puede alcanzar su capacidad máxima si se almacenan

datos no persistentes, como archivos de registro y tablas temporales, en el almacenamiento local. Esta alarma puede evitar un error de falta de espacio que se produce cuando la instancia de base de datos se queda sin almacenamiento local.

Estadística: Average

Umbral recomendado: depende de la situación

Justificación del umbral: debe calcular entre el 10 y el 20 % de la cantidad de almacenamiento disponible en función de la velocidad y la tendencia del volumen de uso y, a continuación, utilizar ese resultado como valor límite para tomar medidas proactivas antes de que el volumen alcance su límite.

Período: 60

Puntos de datos para la alarma: 5

Períodos de evaluación: 5

Operador de comparación: LESS_THAN_THRESHOLD

FreeStorageSpace

Dimensiones: DBInstanceIdentifier

Descripción de la alarma: esta alarma comprueba si hay poco espacio de almacenamiento disponible. Considere la posibilidad de ampliar el almacenamiento de la base de datos si se acerca con frecuencia a los límites de capacidad de almacenamiento. Incluya búfer para asumir incrementos imprevistos de la demanda de las aplicaciones. Como alternativa, considere habilitar el escalado automático del almacenamiento de RDS. Además, considere la posibilidad de liberar más espacio eliminando los datos y registros no utilizados u obsoletos. Para obtener más información, consulte el documento sobre el agotamiento de almacenamiento de RDS y el documento sobre problemas de almacenamiento de PostgreSQL.

Finalidad: esta alarma ayuda a evitar problemas de almacenamiento lleno. Esto puede evitar el tiempo de inactividad que ocurre cuando la instancia de la base de datos se queda sin almacenamiento. No recomendamos usar esta alarma si tiene activado el escalado automático de almacenamiento o si cambia con frecuencia la capacidad de almacenamiento de la instancia de base de datos.

Estadística: Minimum

Umbral recomendado: depende de la situación

Justificación del umbral: el valor del umbral dependerá del espacio de almacenamiento actualmente asignado. Por lo general, debe calcular el valor del 10 por ciento del espacio de almacenamiento asignado y utilizar ese resultado como valor del umbral.

Período: 60

Puntos de datos para la alarma: 5

Períodos de evaluación: 5

Operador de comparación: LESS_THAN_THRESHOLD

MaximumUsedTransactionIDs

Dimensiones: DBInstanceIdentifier

Descripción de la alarma: esta alarma ayuda a evitar que los ID de transacción se distorsionen en PostgreSQL. Consulte los pasos de solución de problemas de este blog para investigar y resolver el problema. También puede consultar este blog para familiarizarse aún más con los conceptos de aspiración automática, los problemas más comunes y las prácticas recomendadas.

Finalidad: esta alarma se utiliza para evitar que los ID de transacción se distorsionen en PostgreSQL.

Estadística: Average

Umbral recomendado: 1,0E9

Justificación del umbral: si se establece este umbral en mil millones, tendrá tiempo para investigar el problema. El valor predeterminado de autovacuum_freeze_max_age es de 200 millones. Si la edad de la transacción más antigua es de mil millones, la aspiración automática tendrá problemas para mantener este umbral por debajo del objetivo de 200 millones de ID de transacción.

Período: 60

Puntos de datos para la alarma: 1

Períodos de evaluación: 1

Operador de comparación: GREATER_THAN_THRESHOLD

ReadLatency

Dimensiones: DBInstanceIdentifier

Descripción de la alarma: esta alarma ayuda a supervisar la alta latencia de lectura. Si la latencia de almacenamiento es alta, se debe a que la carga de trabajo supera los límites de los recursos. Puede revisar el uso de E/S en relación con la configuración de la instancia y del almacenamiento asignado. Consulte la sección de solución de problemas de latencia de los volúmenes de Amazon EBS causada por un cuello de botella de IOPS. Para Aurora, puede cambiar a una clase de instancia que tenga una configuración de almacenamiento optimizado para E/S. Consulte Planificación de E/S en Aurora para obtener orientación.

Finalidad: esta alarma se utiliza para detectar una latencia de lectura alta. Los discos de las bases de datos suelen tener una latencia de lectura/escritura baja, pero pueden tener problemas que provoquen operaciones de alta latencia.

Estadística: p90

Umbral recomendado: depende de la situación

Justificación del umbral: el valor del umbral recomendado para esta alarma depende en gran medida del caso de uso. Es probable que las latencias de lectura superiores a 20 milisegundos sean motivo de investigación. También puede establecer un umbral más alto si la aplicación puede tener una latencia mayor para las operaciones de lectura. Revise la criticidad y los requisitos de latencia de lectura y analice el comportamiento histórico de esta métrica para determinar los niveles de umbral razonables.

Período: 60

Puntos de datos para la alarma: 5

Períodos de evaluación: 5

Operador de comparación: GREATER_THAN_THRESHOLD

ReplicaLag

Dimensiones: DBInstanceIdentifier

Descripción de la alarma: esta alarma ayuda a comprender la cantidad de segundos que una réplica está atrasada respecto de la instancia principal. Una réplica de lectura de PostgreSQL registra un retraso de replicación de hasta cinco minutos si no se están produciendo transacciones de usuario en la instancia de base de datos de origen. Cuando la métrica ReplicaLag llegue a 0, la réplica estará funcionando al mismo ritmo que la instancia de base de datos principal. Si la métrica ReplicaLag devuelve -1, la replicación actualmente

no está activa. Para obtener orientación relacionada con PostgreSQL en RDS, consulte las prácticas recomendadas de replicación y, para solucionar problemas de `ReplicaLag` y errores relacionados, consulte solución de problemas de ReplicaLag.

Finalidad: esta alarma puede detectar el retraso en la réplica, lo que refleja la pérdida de datos que podría producirse en caso de que hubiera un error en la instancia principal. Si la réplica se queda muy por detrás de la instancia principal y esta falla, la réplica no tendrá datos que estaban en la instancia principal.

Estadística: Maximum

Umbral recomendado: 60,0

Justificación del umbral: normalmente, el retraso aceptable depende de la aplicación. Se recomienda no más de 60 segundos.

Período: 60

Puntos de datos para la alarma: 10

Períodos de evaluación: 10

Operador de comparación: GREATER_THAN_THRESHOLD

WriteLatency

Dimensiones: DBInstanceIdentifier

Descripción de la alarma: esta alarma ayuda a supervisar la alta latencia de escritura. Si la latencia de almacenamiento es alta, se debe a que la carga de trabajo supera los límites de los recursos. Puede revisar el uso de E/S en relación con la configuración de la instancia y del almacenamiento asignado. Consulte la sección de solución de problemas de latencia de los volúmenes de Amazon EBS causada por un cuello de botella de IOPS. Para Aurora, puede cambiar a una clase de instancia que tenga una configuración de almacenamiento optimizado para E/S. Consulte Planificación de E/S en Aurora para obtener orientación.

Finalidad: esta alarma se utiliza para detectar una latencia de lectura alta. Si bien los discos de las bases de datos suelen tener una latencia de lectura/escritura baja, pueden experimentar problemas que provoquen operaciones de alta latencia. Supervisar esto garantizará que la latencia del disco sea tan baja como se esperaba.

Estadística: p90

Umbral recomendado: depende de la situación

Justificación del umbral: el valor del umbral recomendado para esta alarma depende en gran medida del caso de uso. Es probable que las latencias de escritura superiores a 20 milisegundos sean motivo de investigación. También puede establecer un umbral más alto si la aplicación puede tener una latencia mayor para las operaciones de escritura. Revise la criticidad y los requisitos de latencia de escritura y analice el comportamiento histórico de esta métrica para determinar los niveles de umbral razonables.

Período: 60

Puntos de datos para la alarma: 5

Períodos de evaluación: 5

Operador de comparación: GREATER_THAN_THRESHOLD

DBLoad

Dimensiones: DBInstanceIdentifier

Descripción de la alarma: esta alarma ayuda a supervisar una carga de base de datos alta. Si el número de procesos supera el número de vCPU, los procesos comienzan a ponerse en cola. Cuando aumentan las colas, el rendimiento se ve afectado. Si la carga de base de datos suele estar por encima del máximo de la CPU virtual y el estado de espera principal es CPU, la CPU del sistema está sobrecargada. En este caso, puede supervisar `CPUUtilization`, `DBLoadCPU` y tareas en la cola en Información del rendimiento/Supervisión mejorada. Quizá desee limitar las conexiones con la instancia, ajustar las consultas SQL con una carga de CPU alta o pensar en la posibilidad de usar una clase de instancia de mayor tamaño. Si hay instancias altas y uniformes en cualquier estado de espera, eso indica que es posible que haya problemas de contención de recursos o cuellos de botella que hay que resolver.

Finalidad: esta alarma se utiliza para detectar una carga elevada de base de datos. Una carga elevada de la base de datos puede provocar problemas de rendimiento en la instancia de base de datos. Esta alarma no se aplica a las instancias de base de datos sin servidor.

Estadística: Average

Umbral recomendado: depende de la situación

Justificación del umbral: El valor máximo de la CPU virtual se determina por el número de núcleos de vCPU (CPU virtual) de la instancia de base de datos. En función de la vCPU máxima, pueden

ser adecuados diferentes valores para el umbral. Lo ideal es que la carga de la base de datos no supere la línea de vCPU.

Período: 60

Puntos de datos para la alarma: 15

Períodos de evaluación: 15

Operador de comparación: GREATER_THAN_THRESHOLD

AuroraVolumeBytesLeftTotal

Dimensiones: DBClusterIdentifier

Descripción de la alarma: esta alarma ayuda a supervisar el bajo volumen total restante. Cuando el volumen total restante alcanza el límite de tamaño, el clúster informa un error de falta de espacio. El almacenamiento Aurora escala automáticamente con los datos del volumen del clúster y se expande hasta 128 TiB o 64 TiB, según la versión del motor de base de datos. Considere la posibilidad de reducir almacenamiento eliminando las tablas y bases de datos que ya no necesite. Para obtener más información, consulte escalado del almacenamiento.

Finalidad: esta alarma se utiliza para detectar qué tan cerca está el clúster de Aurora del límite de tamaño del volumen. Esta alarma puede evitar un error de falta de espacio que se produce cuando el clúster se queda sin espacio. Esta alarma se recomienda solo para Aurora MySQL.

Estadística: Average

Umbral recomendado: depende de la situación

Justificación del umbral: debe calcular entre el 10 y el 20 % del límite de tamaño real en función de la velocidad y la tendencia del aumento del volumen de uso y, a continuación, utilizar ese resultado como valor del umbral para tomar medidas proactivas antes de que el volumen alcance su límite.

Período: 60

Puntos de datos para la alarma: 5

Períodos de evaluación: 5

Operador de comparación: LESS_THAN_THRESHOLD

AuroraBinlogReplicaLag

Dimensiones: DBClusterIdentifier, Rol=ESCRITURA

Descripción de la alarma: esta alarma ayuda a supervisar el estado de error de la replicación de la instancia de escritura de Aurora. Para obtener más información, consulte la reproducción de clústeres de base de datos de Aurora MySQL entre distintas regiones de AWS. Para solucionar problemas, consulte Problemas de replicación de Aurora MySQL.

Finalidad: esta alarma se utiliza para detectar si la instancia de escritura se encuentra en un estado de error y no puede replicar el origen. Esta alarma se recomienda solo para Aurora MySQL.

Estadística: Average

Umbral recomendado: -1,0

Justificación del umbral: se recomienda utilizar -1 como valor del umbral, ya que Aurora MySQL publica este valor si la réplica presenta un estado de error.

Período: 60

Puntos de datos para la alarma: 2

Períodos de evaluación: 2

Operador de comparación: LESS_THAN_OR_EQUAL_TO_THRESHOLD

BlockedTransactions

Dimensiones: DBInstanceIdentifier

Descripción de la alarma: esta alarma ayuda a supervisar un número elevado de transacciones bloqueadas en una instancia de base de datos Aurora. Las transacciones bloqueadas pueden terminar en una reversión o en una confirmación. La alta simultaneidad, las transacciones inactivas o las transacciones de larga duración pueden provocar el bloqueo de las transacciones. Para solucionar problemas, consulte la documentación de Aurora MySQL.

Finalidad: esta alarma se utiliza para detectar un número elevado de transacciones bloqueadas en una instancia de base de datos de Aurora a fin de evitar la reversión de las transacciones y el deterioro del rendimiento.

Estadística: Average

Umbral recomendado: depende de la situación

Justificación del umbral: debe calcular el 5 % de todas las transacciones de la instancia utilizando la métrica `ActiveTransactions` y utilizar ese resultado como valor del umbral. También puede revisar la criticidad y los requisitos de las transacciones bloqueadas y analizar el comportamiento histórico de esta métrica para determinar los niveles de umbral razonables.

Período: 60

Puntos de datos para la alarma: 5

Períodos de evaluación: 5

Operador de comparación: GREATER_THAN_THRESHOLD

BufferCacheHitRatio

Dimensiones: DBInstanceIdentifier

Descripción de la alarma: esta alarma ayuda a supervisar una tasa de aciertos de caché baja y constante del clúster Aurora. Si la tasa de aciertos es baja, eso indica que las consultas de esta instancia de base de datos van al disco con frecuencia. Para solucionar problemas, investigue la carga de trabajo para ver qué consultas están causando este comportamiento y consulte el documento recomendaciones sobre RAM de instancias de base de datos.

Finalidad: esta alarma se utiliza para detectar una tasa de aciertos de caché baja y constante a fin de evitar una disminución sostenida del rendimiento en la instancia de Aurora.

Estadística: Average

Umbral recomendado: 80,0

Justificación del umbral: puede establecer el umbral para la tasa de aciertos de la caché del búfer en un 80 %. Sin embargo, puede ajustar este valor en función del nivel de rendimiento y las características de la carga de trabajo aceptables.

Período: 60

Puntos de datos para la alarma: 10

Períodos de evaluación: 10

Operador de comparación: LESS_THAN_THRESHOLD

EngineUptime

Dimensiones: DBClusterIdentifier, Rol=ESCRITURA

Descripción de la alarma: esta alarma ayuda a supervisar el bajo tiempo de inactividad de la instancia de base de datos de escritura. La instancia de base de datos de escritura puede dejar de funcionar debido a un reinicio, mantenimiento, actualización o conmutación por error. Cuando el tiempo de actividad alcanza 0 debido a una conmutación por error en el clúster y el clúster tiene una o más réplicas de Aurora, entonces la réplica asciende a la instancia de escritura primaria durante un evento de error. Para aumentar la disponibilidad del clúster de base de datos, considere crear al menos una o varias réplicas de Aurora en dos o más zonas de disponibilidad diferentes. Para obtener más información, consulte factores que influyen en el tiempo de inactividad de Aurora.

Finalidad: esta alarma se utiliza para detectar si la instancia de base de datos de escritura de Aurora está inactiva. Esto puede evitar un fallo prolongado en la instancia de escritura que se produzca debido a un bloqueo o una conmutación por error.

Estadística: Average

Umbral recomendado: 0,0

Justificación del umbral: un evento de error provoca una interrupción breve durante la cual las operaciones de lectura y escritura generan errores con una excepción. Sin embargo, el servicio se suele restaurar en menos de 60 segundos y, en muchos casos, en menos de 30 segundos.

Período: 60

Puntos de datos para la alarma: 2

Períodos de evaluación: 2

Operador de comparación: LESS_THAN_OR_EQUAL_TO_THRESHOLD

RollbackSegmentHistoryListLength

Dimensiones: DBInstanceIdentifier

Descripción de la alarma: esta alarma ayuda a supervisar una longitud constante del historial de segmentos de reversión alta de una instancia de Aurora. Una gran longitud de la lista del historial de InnoDB indica que hay un gran número de versiones de filas antiguas, y las consultas y los cierres de bases de datos se han ralentizado. Para obtener más información y solucionar

problemas, consulte la documentación sobre el aumento significativo de la longitud de la lista del historial de InnoDB.

Finalidad: esta alarma se utiliza para detectar una longitud constante del historial de segmentos de reversión elevada. Esto puede ayudarle a evitar un deterioro sostenido del rendimiento y un uso elevado de la CPU en la instancia de Aurora. Esta alarma solo está disponible para Amazon MySQL.

Estadística: Average

Umbral recomendado: 1000000,0

Justificación del umbral: si establece este umbral en 1 millón, tendrá tiempo para investigar el problema. Sin embargo, puede ajustar este valor en función del nivel de rendimiento y las características de la carga de trabajo aceptables.

Período: 60

Puntos de datos para la alarma: 5

Períodos de evaluación: 5

Operador de comparación: GREATER_THAN_THRESHOLD

StorageNetworkThroughput

Dimensiones: DBClusterIdentifier, Rol=ESCRITURA

Descripción de la alarma: esta alarma ayuda a supervisar el alto rendimiento de la red de almacenamiento. Si el rendimiento de la red de almacenamiento supera el ancho de banda total de la red de la instancia EC2, se puede producir una latencia de lectura y escritura elevada, lo que puede reducir el rendimiento. Puede comprobar el tipo de instancia EC2 desde la consola de AWS. Para solucionar problemas, compruebe cualquier cambio en las latencias de escritura/lectura y evalúe si también ha activado una alarma en esta métrica. Si ese es el caso, evalúe el patrón de la carga de trabajo durante las horas en que se activó la alarma. Esto puede ayudar a identificar si puede optimizar la carga de trabajo para reducir la cantidad total de tráfico de red. Si esto no es posible, puede que tenga que considerar la posibilidad de escalar la instancia.

Finalidad: esta alarma se utiliza para detectar un alto rendimiento de la red de almacenamiento. La detección de un alto rendimiento puede evitar la caída de paquetes de red y el deterioro del rendimiento.

Estadística: Average

Umbral recomendado: depende de la situación

Justificación del umbral: debe calcular entre el 80 y el 90 % del ancho de banda total de la red del tipo de instancia EC2 y, a continuación, utilizar ese resultado como valor del umbral para tomar medidas de forma proactiva antes de que los paquetes de red se vean afectados. También puede revisar la criticidad y los requisitos del rendimiento de la red de almacenamiento y analizar el comportamiento histórico de esta métrica para determinar los niveles de umbral razonables.

Período: 60

Puntos de datos para la alarma: 5

Períodos de evaluación: 5

Operador de comparación: GREATER_THAN_THRESHOLD

Amazon Route 53 Public Data Plane

HealthCheckStatus

Dimensiones: HealthCheckId

Descripción de la alarma: esta alarma ayuda a detectar puntos de conexión en mal estado según los comprobadores de estado. Para entender el motivo de un fallo que provoca el mal estado, utilice la pestaña Comprobadores de estado de la consola de Comprobación de estado de Route 53 para ver el estado de cada región, así como el último error de la comprobación de estado. La pestaña de estado también muestra el motivo por el que se informa que el punto de conexión está en mal estado. Consulte los pasos para la solución de problemas.

Finalidad: esta alarma utiliza los comprobadores de estado de Route 53 para detectar puntos de conexión en mal estado.

Estadística: Average

Umbral recomendado: 1,0

Justificación del umbral: el estado del punto de conexión se informa como 1 cuando está en buen estado. Cualquier valor menor a 1 se considera en mal estado.

Período: 60

Puntos de datos para la alarma: 3

Períodos de evaluación: 3

Operador de comparación: LESS_THAN_THRESHOLD

Amazon S3

4XXErrors

Dimensiones: BucketName, FilterId

Descripción de la alarma: esta alarma nos ayuda a informar del número total de códigos de estado de error 4XX que se crean en respuesta a las solicitudes de los clientes. Por ejemplo, los códigos de error 403 pueden indicar una política de IAM incorrecta y los códigos de error 404 pueden indicar un mal comportamiento de la aplicación cliente. La habilitación del registro de acceso al servidor S3 de forma temporal, podría ayudarlo a identificar el origen del problema mediante los campos de estado HTTP y de código de error. Para obtener más información sobre el código de error, consulte Error Responses.

Finalidad: esta alarma se utiliza para crear una referencia para las tasas típicas de errores 4XX, de forma que pueda detectar cualquier anomalía que pudiera indicar un problema de configuración.

Estadística: Average

Umbral recomendado: 0,05

Justificación del umbral: el umbral recomendado es detectar si más del 5 % del total de las solicitudes reciben errores 4XX. Los errores 4XX que se producen con frecuencia deberían disparar las alarmas. Sin embargo, si se establece un valor muy bajo para el umbral, la alarma puede resultar demasiado sensible. También puede ajustar el umbral para adaptarlo a la carga de las solicitudes, teniendo en cuenta un nivel aceptable de errores 4XX. También puede analizar los datos históricos para encontrar la tasa de error aceptable para la carga de trabajo de la aplicación y, a continuación, ajustar el umbral en consecuencia.

Período: 60

Puntos de datos para la alarma: 15

Períodos de evaluación: 15

Operador de comparación: GREATER_THAN_THRESHOLD

5xxErrors

Dimensiones: BucketName, FilterId

Descripción de la alarma: esta alarma ayuda a detectar una gran cantidad de errores por parte del servidor. Estos errores indican que un cliente realizó una solicitud que el servidor no pudo completar. Esto puede ayudarlo a correlacionar el problema al que se enfrenta su aplicación debido a S3. Para obtener más información que lo ayude a gestionar o reducir los errores de manera eficiente, consulte Optimización de los patrones de diseño de rendimiento. Los errores también pueden deberse a un problema con S3. Compruebe el panel de estado del servicio de AWS para conocer el estado de Amazon S3 en su región.

Finalidad: esta alarma puede ayudar a detectar si la aplicación tiene problemas debido a errores 5XX.

Estadística: Average

Umbral recomendado: 0,05

Justificación del umbral: se recomienda establecer el umbral para detectar si más del 5 % del total de las solicitudes reciben errores 5XX. Sin embargo, puede ajustar el umbral para adaptarlo al tráfico de las solicitudes y a las tasas de error aceptables. También puede analizar los datos históricos para ver cuál es la tasa de error aceptable para la carga de trabajo de la aplicación y ajustar el umbral en consecuencia.

Período: 60

Puntos de datos para la alarma: 15

Períodos de evaluación: 15

Operador de comparación: GREATER_THAN_THRESHOLD

OperationsFailedReplication

Dimensiones: SourceBucket, DestinationBucket, RuleId

Descripción de la alarma: esta alarma ayuda a comprender un error de replicación. Esta métrica rastrea el estado de los nuevos objetos replicados mediante CRR de S3 o SRR de S3, y también rastrea los objetos existentes replicados mediante la replicación por lotes de S3. Consulte Solución de problemas de replicación para obtener más información.

Finalidad: esta alarma se utiliza para detectar si falló una operación de replicación.

Estadística: Maximum

Umbral recomendado: 0,0

Justificación del umbral: esta métrica emite un valor de 0 si las operaciones se realizan de forma correcta y no emite ningún valor si no se realizaron operaciones de replicación para el minuto. Cuando la métrica emite un valor superior a 0, significa que la operación de replicación no se realizó de forma correcta.

Período: 60

Puntos de datos para la alarma: 5

Períodos de evaluación: 5

Operador de comparación: GREATER_THAN_THRESHOLD

`S3ObjectLambda`

4xxErrors

Dimensiones: AccessPointName, DataSourceARN

Descripción de la alarma: esta alarma nos ayuda a informar del número total de códigos de estado de errores 4XX que se crean en respuesta a las solicitudes de los clientes. La habilitación del registro de acceso al servidor S3 de forma temporal, podría ayudarlo a identificar el origen del problema mediante los campos de estado HTTP y de código de error.

Finalidad: esta alarma se utiliza para crear una referencia para las tasas típicas de errores 4XX, de forma que pueda detectar cualquier anomalía que pudiera indicar un problema de configuración.

Estadística: Average

Umbral recomendado: 0,05

Justificación del umbral: se recomienda establecer el umbral para detectar si más del 5 % del total de las solicitudes reciben errores 4XX. Los errores 4XX que se producen con frecuencia deberían disparar las alarmas. Sin embargo, si se establece un valor muy bajo para el umbral, la alarma puede resultar demasiado sensible. También puede ajustar el umbral para adaptarlo a la carga de

las solicitudes, teniendo en cuenta un nivel aceptable de errores 4XX. También puede analizar los datos históricos para encontrar la tasa de error aceptable para la carga de trabajo de la aplicación y, a continuación, ajustar el umbral en consecuencia.

Período: 60

Puntos de datos para la alarma: 15

Períodos de evaluación: 15

Operador de comparación: GREATER_THAN_THRESHOLD

5xxErrors

Dimensiones: AccessPointName, DataSourceARN

Descripción de la alarma :esta alarma ayuda a detectar un gran número de errores del lado del servidor. Estos errores indican que un cliente realizó una solicitud que el servidor no pudo completar. Estos errores pueden deberse a un problema con S3. Compruebe el panel de estado del servicio de AWS para conocer el estado de Amazon S3 en su región. Esto puede ayudarlo a correlacionar el problema al que se enfrenta su aplicación debido a S3. Para obtener información que lo ayude a gestionar o reducir estos errores de forma eficiente, consulte Optimización de los patrones de diseño de rendimiento.

Finalidad: esta alarma puede ayudar a detectar si la aplicación tiene problemas debido a errores 5XX.

Estadística: Average

Umbral recomendado: 0,05

Justificación del umbral:recomendamos establecer el umbral para detectar si más del 5 % del total de las solicitudes reciben errores 5XX. Sin embargo, puede ajustar el umbral para adaptarlo al tráfico de las solicitudes y a las tasas de error aceptables. También puede analizar los datos históricos para ver cuál es la tasa de error aceptable para la carga de trabajo de la aplicación y ajustar el umbral en consecuencia.

Período: 60

Puntos de datos para la alarma: 15

Períodos de evaluación: 15

Operador de comparación: GREATER_THAN_THRESHOLD

LambdaResponse4xx

Dimensiones: AccessPointName, DataSourceARN

Descripción de la alarma: esta alarma ayuda a detectar y diagnosticar errores (del tipo 500) en las llamadas a S3 Object Lambda. Estos errores pueden deberse a errores o problemas en la configuración en la función de Lambda encargada de responder a sus solicitudes. Investigar los flujos de registro de CloudWatch de la función de Lambda asociada al punto de acceso de Object Lambda puede ayudarle a determinar el origen del problema en función de la respuesta de S3 Object Lambda.

Finalidad: esta alarma se utiliza para detectar errores 4XX del cliente en las llamadas a WriteGetObjectResponse.

Estadística: Average

Umbral recomendado: 0,05

Justificación del umbral: se recomienda establecer el umbral para detectar si más del 5 % del total de las solicitudes reciben errores 4XX. Los errores 4XX que se producen con frecuencia deberían disparar las alarmas. Sin embargo, si se establece un valor muy bajo para el umbral, la alarma puede resultar demasiado sensible. También puede ajustar el umbral para adaptarlo a la carga de las solicitudes, teniendo en cuenta un nivel aceptable de errores 4XX. También puede analizar los datos históricos para encontrar la tasa de error aceptable para la carga de trabajo de la aplicación y, a continuación, ajustar el umbral en consecuencia.

Período: 60

Puntos de datos para la alarma: 15

Períodos de evaluación: 15

Operador de comparación: GREATER_THAN_THRESHOLD

Amazon SNS

NumberOfMessagesPublished

Dimensiones: TopicName

Descripción de la alarma: esta alarma puede detectar cuando el número de mensajes SNS publicados es demasiado bajo. Para solucionar problemas, compruebe por qué los publicadores envían menos tráfico.

Finalidad: esta alarma ayuda a supervisar y detectar de forma proactiva descensos significativos en la publicación de notificaciones. Esto le ayuda a identificar posibles problemas con su aplicación o sus procesos empresariales, de modo que pueda tomar las medidas adecuadas para mantener el flujo de notificaciones esperado. Debe crear esta alarma si espera que su sistema tenga un tráfico mínimo que atender.

Estadística: Sum

Umbral recomendado: depende de la situación

Justificación del umbral: el número de mensajes publicados debe coincidir con el número esperado de mensajes publicados para su aplicación. También puede analizar los datos históricos, las tendencias y el tráfico para encontrar el umbral correcto.

Período: 60

Puntos de datos para la alarma: 5

Períodos de evaluación: 5

Operador de comparación: LESS_THAN_THRESHOLD

NumberOfNotificationsDelivered

Dimensiones: TopicName

Descripción de la alarma: esta alarma puede detectar cuando el número de mensajes SNS entregados es demasiado bajo. Esto puede deberse a la cancelación accidental de la suscripción de un punto de conexión o a un evento de SNS que provoque un retraso en los mensajes.

Finalidad: esta alarma ayuda a detectar una caída en el volumen de mensajes enviados. Debe crear esta alarma si espera que su sistema tenga un tráfico mínimo que atender.

Estadística: Sum

Umbral recomendado: depende de la situación

Justificación del umbral: el número de mensajes entregados debe coincidir con el número esperado de mensajes producidos y el número de consumidores. También puede analizar los datos históricos, las tendencias y el tráfico para encontrar el umbral correcto.

Período: 60

Puntos de datos para la alarma: 5

Períodos de evaluación: 5

Operador de comparación: LESS_THAN_THRESHOLD

NumberOfNotificationsFailed

Dimensiones: TopicName

Descripción de la alarma: esta alarma puede detectar cuando el número de mensajes SNS fallidos es demasiado alto. Para solucionar problemas con las notificaciones fallidas, habilite el registro en los Registros de CloudWatch. Revisar los registros puede ayudarle a determinar qué suscriptores fallan, así como los códigos de estado que devuelven.

Finalidad: esta alarma ayuda a detectar de forma proactiva los problemas relacionados con la entrega de las notificaciones y a tomar las medidas adecuadas para solucionarlos.

Estadística: Sum

Umbral recomendado: depende de la situación

Justificación del umbral: el valor del umbral recomendado para esta alarma depende en gran medida del impacto de las notificaciones fallidas. Revise los acuerdos de nivel de servicio (SLA) proporcionados a sus usuarios finales, la tolerancia a los errores, la importancia de las notificaciones y analice los datos históricos y, a continuación, seleccione el umbral correspondiente. El número de notificaciones fallidas debe ser 0 para los temas que solo tienen suscripciones a SQS, Lambda o Firehose.

Período: 60

Puntos de datos para la alarma: 5

Períodos de evaluación: 5

Operador de comparación: GREATER_THAN_THRESHOLD

NumberOfNotificationsFilteredOut-InvalidAttributes

Dimensiones: TopicName

Descripción de la alarma: esta alarma ayuda a supervisar y resolver posibles problemas con el publicador o los suscriptores. Compruebe si un publicador publica mensajes con atributos no válidos o si se aplica un filtro inadecuado a un suscriptor. También puede analizar los Registros de CloudWatch para ayudar a encontrar la causa raíz del problema.

Finalidad: la alarma se utiliza para detectar si los mensajes publicados no son válidos o si se han aplicado filtros inadecuados a un suscriptor.

Estadística: Sum

Umbral recomendado: 0,0

Justificación del umbral: los atributos no válidos casi siempre son un error del publicador. Se recomienda establecer el umbral en 0, ya que en un sistema en buen estado no deberían esperarse atributos no válidos.

Período: 60

Puntos de datos para la alarma: 5

Períodos de evaluación: 5

Operador de comparación: GREATER_THAN_THRESHOLD

NumberOfNotificationsFilteredOut-InvalidMessageBody

Dimensiones: TopicName

Descripción de la alarma: esta alarma ayuda a supervisar y resolver posibles problemas con el publicador o los suscriptores. Compruebe si un editor publica mensajes con cuerpos de mensaje no válidos o si se aplica un filtro inadecuado a un suscriptor. También puede analizar los Registros de CloudWatch para ayudar a encontrar la causa raíz del problema.

Finalidad: la alarma se utiliza para detectar si los mensajes publicados no son válidos o si se han aplicado filtros inadecuados a un suscriptor.

Estadística: Sum

Umbral recomendado: 0,0

Justificación del umbral: los cuerpos de los mensajes no válidos casi siempre son un error del publicador. Recomendamos establecer el umbral en 0, ya que los cuerpos de los mensajes deberían ser válidos en un sistema en buen estado.

Período: 60

Puntos de datos para la alarma: 5

Períodos de evaluación: 5

Operador de comparación: GREATER_THAN_THRESHOLD

NumberOfNotificationsRedrivenToDlq

Dimensiones: TopicName

Descripción de la alarma: esta alarma ayuda a supervisar la cantidad de mensajes que se mueven a una cola de mensajes fallidos.

Finalidad: la alarma se utiliza para detectar mensajes que pasaron a una cola de mensajes fallidos. Recomendamos que cree esta alarma cuando SNS esté acoplado a SQS, Lambda o Firehose.

Estadística: Sum

Umbral recomendado: 0,0

Justificación del umbral: en un sistema en buen estado de cualquier tipo de suscriptor, los mensajes no deben moverse a la cola de mensajes fallidos. Recomendamos que reciba una notificación en caso de que algún mensaje llegue a la cola, de modo que pueda identificar y abordar la causa raíz y, si es posible, redireccionar los mensajes en la cola de mensajes fallidos para evitar la pérdida de datos.

Período: 60

Puntos de datos para la alarma: 5

Períodos de evaluación: 5

Operador de comparación: GREATER_THAN_THRESHOLD

NumberOfNotificationsFailedToRedriveToDlq

Dimensiones: TopicName

Descripción de la alarma: esta alarma ayuda a supervisar los mensajes que no se pudieron mover a una cola de mensajes fallidos. Compruebe si existe una cola de mensajes fallidos y si está configurada como corresponde. Compruebe también que SNS tenga los permisos para acceder

a la cola de mensajes fallidos. Consulte la documentación sobre cola de mensajes fallidos para obtener más información.

Finalidad: la alarma se utiliza para detectar los mensajes que no se pudieron mover a una cola de mensajes fallidos.

Estadística: Sum

Umbral recomendado: 0,0

Justificación del umbral: casi siempre se trata de un error si los mensajes no se pueden mover a la cola de mensajes fallidos. El umbral recomendado es 0, lo que significa que todos los mensajes que no se procesen de forma correcta deben ser capaces de llegar a la cola de mensajes fallidos, una vez configurada.

Período: 60

Puntos de datos para la alarma: 5

Períodos de evaluación: 5

Operador de comparación: GREATER_THAN_THRESHOLD

SMSMonthToDateSpentUSD

Dimensiones: TopicName

Descripción de la alarma: la alarma ayuda a controlar si tiene una cuota suficiente en su cuenta para que SNS pueda entregar los mensajes. Si alcanza su cuota, SNS no podrá entregar los mensajes SMS. Para obtener más información acerca de la configuración de su cuota de gasto mensual de SMS o acerca de cómo solicitar un aumento de la cuota de gasto con AWS, consulte Configuración de las preferencias de mensajería SMS.

Finalidad: esta alarma se utiliza para detectar si tiene una cuota suficiente en su cuenta para que los mensajes SMS se entreguen de forma correcta.

Estadística: Maximum

Umbral recomendado: depende de la situación

Justificación del umbral: establezca el umbral de acuerdo con la cuota (límite de gasto de la cuenta) para la cuenta. Elija un umbral que informe con suficiente antelación de que está alcanzando el límite de cuota para que tenga tiempo de solicitar un aumento.

Período: 60

Puntos de datos para la alarma: 5

Períodos de evaluación: 5

Operador de comparación: GREATER_THAN_THRESHOLD

SMSSuccessRate

Dimensiones: TopicName

Descripción de la alarma: esta alarma ayuda a supervisar la tasa de entregas fallidas de mensajes SMS. Puede configurar los Registros de CloudWatch para entender la naturaleza del error y tomar medidas en función de ello.

Finalidad: esta alarma se utiliza para detectar errores en la entrega de mensajes SMS.

Estadística: Average

Umbral recomendado: depende de la situación

Justificación del umbral: establezca el umbral de la alarma de acuerdo con su tolerancia ante la entrega fallida de mensajes SMS.

Período: 60

Puntos de datos para la alarma: 5

Períodos de evaluación: 5

Operador de comparación: GREATER_THAN_THRESHOLD

Amazon SQS

ApproximateAgeOfOldestMessage

Dimensiones: QueueName

Descripción de la alarma: esta alarma comprueba la antigüedad del mensaje más antiguo en la cola. Puede utilizar esta alarma para supervisar si sus consumidores procesan los mensajes SQS a la velocidad deseada. Considere aumentar el número o el rendimiento de los consumidores

para reducir la antigüedad de los mensajes. Esta métrica se puede utilizar en combinación con `ApproximateNumberOfMessagesVisible` para determinar el tamaño de la cola de espera y la rapidez con la que se procesan los mensajes. Para evitar que los mensajes se eliminen antes de procesarlos, considere la posibilidad de configurar la cola de mensajes fallidos para dejar de lado los posibles mensajes de tipo píldora venenosa (mensajes que se reciben, pero no se pueden procesar).

Finalidad: esta alarma se utiliza para detectar si el mensaje más viejo en la cola de QueueName es demasiado antiguo. La antigüedad elevada puede indicar que los mensajes no se procesan con la suficiente rapidez o que hay algunos mensajes de tipo píldora venenosa que se quedan atascados en la cola y no se pueden procesar.

Estadística: Maximum

Umbral recomendado: depende de la situación

Justificación del umbral: el valor del umbral recomendado para esta alarma depende en gran medida del tiempo esperado de procesamiento de los mensajes. Puede utilizar los datos históricos para calcular el tiempo promedio de procesamiento de los mensajes y, a continuación, establecer el umbral en un 50 % más que el tiempo máximo de procesamiento de mensajes de SQS esperado por los usuarios de la cola.

Período: 60

Puntos de datos para la alarma: 15

Períodos de evaluación: 15

Operador de comparación: GREATER_THAN_OR_EQUAL_TO_THRESHOLD

ApproximateNumberOfMessagesNotVisible

Dimensiones: QueueName

Descripción de la alarma: esta alarma ayuda a detectar un gran número de mensajes en tránsito con respecto a `QueueName`. Para solucionar problemas, consulte ¿Cómo puedo evitar que se acumulen mensajes en mi cola de Amazon SQS?.

Finalidad: esta alarma se utiliza para detectar un número elevado de mensajes en tránsito en la cola. Si los consumidores no eliminan los mensajes dentro del período de tiempo de espera de visibilidad, cuando se sondee la cola, los mensajes volverán a aparecer en esta. En el caso de las

colas FIFO, puede haber un máximo de 20 000 mensajes en tránsito. Si alcanza esta cuota, SQS no devolverá ningún mensaje de error. Una cola FIFO examina los primeros 20 000 mensajes para determinar los grupos de mensajes disponibles. Esto significa que, si tiene una acumulación de mensajes en un solo grupo de mensajes, no podrá consumir los mensajes de otros grupos de mensajes que se hayan enviado más tarde a la cola hasta que no consuman de forma correcta los mensajes acumulados.

Estadística: Average

Umbral recomendado: depende de la situación

Justificación del umbral: el valor del umbral recomendado para esta alarma depende en gran medida del número esperado de mensajes en tránsito. Puede utilizar los datos históricos para calcular el número máximo esperado de mensajes en tránsito y establecer el umbral en un 50 % por encima de este valor. Si los usuarios de la cola procesan, pero no eliminan los mensajes de la cola, este número aumentará de forma repentina.

Período: 60

Puntos de datos para la alarma: 15

Períodos de evaluación: 15

Operador de comparación: GREATER_THAN_OR_EQUAL_TO_THRESHOLD

ApproximateNumberOfMessagesVisible

Dimensiones: QueueName

Descripción de la alarma: esta alarma detecta si la cola de mensajes pendientes es mayor de lo esperada, lo que indica que los consumidores son demasiado lentos o que no hay suficientes consumidores. Considere la posibilidad de aumentar el número de consumidores o acelerar el número de consumidores si esta alarma pasa a estar en estado de ALARMA.

Finalidad: esta alarma se utiliza para detectar si el número de mensajes de la cola activa es demasiado alto y si los consumidores tardan en procesar los mensajes o si no hay suficientes consumidores para procesarlos.

Estadística: Average

Umbral recomendado: depende de la situación

Justificación del umbral: un número alto e imprevisto de mensajes visibles indica que un consumidor no procesa los mensajes al ritmo esperado. Debe tener en cuenta los datos históricos al establecer este umbral.

Período: 60

Puntos de datos para la alarma: 15

Períodos de evaluación: 15

Operador de comparación: GREATER_THAN_OR_EQUAL_TO_THRESHOLD

NumberOfMessagesSent

Dimensiones: QueueName

Descripción de la alarma: esta alarma ayuda a detectar si un productor no envía ningún mensaje con respecto a `QueueName`. Para solucionar problemas, compruebe el motivo por el que el productor no envía los mensajes.

Finalidad: esta alarma se utiliza para detectar cuándo un productor deja de enviar mensajes.

Estadística: Sum

Umbral recomendado: 0,0

Justificación del umbral: si el número de mensajes enviados es 0, significa el productor no envía mensajes. Si esta cola tiene un TPS bajo, aumente el número de EvaluationPeriods en consecuencia.

Período: 60

Puntos de datos para la alarma: 15

Períodos de evaluación: 15

Operador de comparación: LESS_THAN_OR_EQUAL_TO_THRESHOLD

AWS VPN

TunnelState

Dimensiones: VpnId

Descripción de la alarma: esta alarma ayuda a comprender si el estado de uno o más túneles está INACTIVO. Para solucionar problemas, consulte How do I troubleshoot VPN tunnel connectivity to an Amazon VPC?.

Finalidad: esta alarma se utiliza para detectar si al menos un túnel está como INACTIVO para esta VPN, de modo que pueda solucionar los problemas de la VPN afectada. Esta alarma siempre estará en estado de ALARMA en las redes que solo tengan un túnel configurado.

Estadística: Minimum

Umbral recomendado: 1,0

Justificación del umbral: un valor inferior a 1 indica que al menos un túnel está en estado INACTIVO.

Período: 300

Puntos de datos para la alarma: 3

Períodos de evaluación: 3

Operador de comparación: LESS_THAN_THRESHOLD

TunnelState

Dimensiones: TunnelIpAddress

Descripción de la alarma: esta alarma ayuda a entender si el estado de este túnel está INACTIVO. Para solucionar problemas, consulte How do I troubleshoot VPN tunnel connectivity to an Amazon VPC?.

Finalidad: esta alarma se utiliza para detectar si el túnel está en estado INACTIVO, de modo que pueda solucionar los problemas de la VPN afectada. Esta alarma siempre estará en estado de ALARMA en las redes que solo tengan un túnel configurado.

Estadística: Minimum

Umbral recomendado: 1,0

Justificación del umbral: un valor inferior a 1 indica que el túnel está en estado INACTIVO.

Período: 300

Puntos de datos para la alarma: 3

Períodos de evaluación: 3

Operador de comparación: LESS_THAN_THRESHOLD

Alarmas y métricas

Los pasos de las siguientes secciones explican cómo crear alarmas de CloudWatch en las métricas.

Cree una alarma de CloudWatch basada en un umbral estático

Elija una métrica de CloudWatch para la alarma que se va a supervisar, y el umbral para la métrica. La alarma pasa al estado `ALARM` cuando la métrica supera el umbral durante un número especificado de periodos de evaluación.

Si está creando una alarma en una cuenta configurada como cuenta de supervisión en la observabilidad entre cuentas de CloudWatch, puede configurar la alarma para ver una métrica en una cuenta de origen vinculada a esta cuenta de supervisión. Para obtener más información, consulte Observabilidad entre cuentas de CloudWatch.

Para crear una alarma basándose en una sola métrica

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Alarms (Alarmas) y, luego, Create Alarm (Crear alarma).
3. Elija Create alarm (Crear alarma).
4. Elija Select Metric (Seleccionar métrica).
5. Realice una de las siguientes acciones siguientes:
   - Elija el espacio de nombres del servicio que contiene la métrica que desea. Continúe eligiendo opciones conforme aparezcan para delimitar las opciones. Cuando aparezca una lista de métricas, active la casilla de verificación situada junto a la que desee utilizar.
   - En el campo de búsqueda, escriba el nombre de una métrica, ID de cuenta, etiqueta de cuenta, dimensión o ID de recurso. A continuación, elija uno de los resultados y continúe hasta que se muestre una lista de métricas. Seleccione la casilla de verificación situada junto a la métrica que desee.
6. Elija la pestaña Métricas diagramadas.

   a. En Statistic (Estadística), elija una de las estadísticas o percentiles predefinidos o especifique un percentil personalizado (por ejemplo, **`p95.45`**).

b. En Period (Periodo), elija el periodo de evaluación de la alarma. Al evaluar la alarma, cada periodo se agrega a un punto de datos.

También puede elegir si aparece la leyenda del eje Y a la izquierda o la derecha, mientras crea la alarma. Esta preferencia se utiliza únicamente mientras se crea la alarma.

c. Elija Select Metric (Seleccionar métrica).

Aparece la página Specify metric and conditions (Especificar métrica y condiciones), que muestra un gráfico y otra información sobre la métrica y la estadística que ha seleccionado.

7. En Conditions (Condiciones), especifique lo siguiente:

a. En Whenever ***`metric`*** is (Siempre que la métrica sea), especifique si la métrica debe ser mayor que, menor que o igual al umbral. En than... (que...), especifique el valor de umbral.

b. Elija Configuración adicional. Para Puntos de datos para alarma, especifique el número de periodos de evaluación (puntos de datos) que deben tener el estado `ALARM` para que se active la alarma. Si estos dos valores coinciden, creará una alarma que pasará al estado `ALARM` si se infringen muchos periodos consecutivos.

Para crear una alarma M de N, especifique un número menor para el primer valor que el especificado para el segundo valor. Para obtener más información, consulte Evaluación de una alarma.

c. En Missing data treatment (Tratamiento de datos que faltan), elija cómo debe comportarse la alarma cuando falten algunos puntos de datos. Para obtener más información, consulte Configuración de la forma en la que las alarmas de CloudWatch tratan los datos que faltan.

d. Si la alarma utiliza un percentil como estadística supervisada, aparece un cuadro Percentiles with low samples (Percentiles con pocas muestras). Utilícelo para seleccionar si desea evaluar o no tener en cuenta los casos con frecuencias de muestreo bajas. Si elige ignore (maintain alarm state) (ignorar (mantener el estado de alarma)), el estado de alarma actual se mantiene siempre cuando el tamaño de la muestra es demasiado bajo. Para obtener más información, consulte Muestras de datos reducidos y alarmas de CloudWatch basadas en percentiles.

8. Elija Siguiente.

9. En Notification (Notificación), seleccione el tema de SNS al que desee enviar la notificación cuando la alarma tenga el estado `ALARM`, `OK` o `INSUFFICIENT_DATA`.

Para que la alarma envíe varias notificaciones para el mismo estado de alarma o para estados de alarma diferentes, seleccione Add notificación (Añadir notificación).

En la observabilidad entre cuentas de CloudWatch, puede elegir que las notificaciones se envíen a varias cuentas AWS. Por ejemplo, tanto a la cuenta de supervisión como a la cuenta de origen.

Para que la alarma no envíe notificaciones, elija Remove (Eliminar).

10. Para que la alarma realice acciones de escalado automático, EC2, Lambda o de Systems Manager, elija el botón correspondiente y seleccione el estado de la alarma y la acción que se debe realizar. Las alarmas solo pueden realizar acciones de Systems Manager cuando entran en el estado ALARMA. Para obtener más información sobre las acciones de Systems Manager, consulte Configuración de CloudWatch para crear OpsItems desde alarmas y Creación de incidentes.

    > **Note**
    >
    > Para crear una alarma que realice una acción de SSM Incident Manager, debe contar con determinados permisos. Para obtener más información, consulte Ejemplos de políticas basadas en identidades del Administrador de incidentes de AWS Systems Manager.

11. Cuando haya terminado, elija Next (Siguiente).
12. Escriba un nombre y la descripción de la alarma. El nombre debe contener solo caracteres UTF-8 y no puede contener caracteres de control ASCII. La descripción puede incluir el formato Markdown, que solo se muestra en la pestaña Detalles de la alarma de la consola de CloudWatch. Markdown puede resultar útil para añadir enlaces a manuales u otros recursos internos. A continuación, elija Next.
13. En Obtener vista previa y crear, confirme que la información y las condiciones son las que desea y, a continuación, elija Crear alarma.

También puede agregar alarmas a un panel. Para obtener más información, consulte Agregue o elimine un widget de alarma desde un panel de CloudWatch .

Crear una alarma de CloudWatch basándose en una expresión matemática métrica

Para crear una alarma basándose en una expresión matemática métrica, elija una o varias métricas de CloudWatch para utilizarlas en la expresión. A continuación, especifique la expresión, el umbral y los periodos de evaluación.

No puede crear una alarma basada en la expresión SEARCH (BUSCAR). Esto se debe a que las expresiones de búsqueda devuelven varias series temporales, y una alarma basada en una expresión matemática sólo puede ver una serie temporal.

Para crear una alarma que se base en una expresión matemática de métrica

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Alarms (Alarmas) y, a continuación, elija All Alarms (Todas las alarmas).
3. Elija Crear alarma.
4. Elija Select Metric (Seleccionar métrica) y, a continuación, realice alguna de las siguientes acciones:
   - Seleccione un espacio de nombres del menú desplegable (Espacios de nombres de AWS o del menú desplegable Espacios de nombres personalizados. Después de seleccionar un espacio de nombres, siga eligiendo opciones hasta que aparezca una lista de métricas, en la que deberá seleccionar la casilla de verificación ubicada junto a la métrica correcta.
   - Utilice el cuadro de búsqueda para buscar una métrica, un ID de cuenta, una dimensión o un ID de recurso. Después de ingresar la métrica, la dimensión o el ID de recurso, sigue eligiendo opciones hasta que aparezca una lista de métricas, en la que deberá seleccionar la casilla de verificación ubicada junto a la métrica correcta.
5. (Opcional) Si desea agregar otra métrica a una expresión matemática de métrica, puede utilizar el cuadro de búsqueda para encontrar una métrica específica. Puede agregar hasta 10 métricas a una expresión matemática de métrica.
6. Seleccione la pestaña Métricas diagramadas. Para cada una de las métricas que agregó anteriormente, realice las siguientes acciones:

   a. En la columna Statistic (Estadística), seleccione el menú desplegable. En el menú desplegable, elija una de las estadísticas o uno de los percentiles predefinidos. Utilice el cuadro de búsqueda del menú desplegable para especificar un percentil personalizado.

   b. En la columna Period (Periodo), seleccione el menú desplegable. En el menú desplegable, elija alguno de los periodos de evaluación predefinidos.

      Mientras cree su alarma, podrá especificar si la leyenda del eje Y debe aparecer a en el lado izquierdo o derecho del gráfico.

> **Note**
>
> Cuando CloudWatch evalúa las alarmas, los periodos se agrupan como puntos de datos únicos.

7. Elija el menú desplegable Add math (Agregar expresión matemática) y, a continuación, seleccione Start with an empty expression (Comenzar con una expresión vacía) de la lista de expresiones matemáticas de métricas predefinidas.

   Después de elegir Start with an empty expression (Comenzar con una expresión vacía), aparecerá un cuadro de expresión matemática en el que podrá aplicar o editar expresiones matemáticas.

8. En el cuadro de expresión matemática, ingrese su expresión matemática y, a continuación, elija Apply (Aplicar).

   Después de elegir Apply (Aplicar), aparecerá una columna ID junto a la columna Label (Etiqueta).

   Para usar una métrica o el resultado de otra expresión matemática de métrica como parte de la fórmula de la expresión matemática actual, debe utilizar el valor que se muestra en la columna ID. Para cambiar el valor de ID, seleccione el icono de bolígrafo y papel que se encuentra junto al valor actual. El nuevo valor debe comenzar por una letra minúscula y puede incluir números, letras y el símbolo de guion bajo. Cambiar el valor de ID por un nombre más significativo puede facilitar la comprensión del gráfico de la alarma.

   Para obtener información acerca de las funciones que se encuentran disponibles para las expresiones matemáticas de métricas, consulte Sintaxis de matemáticas en las métricas y funciones.

9. (Opcional) Añada más expresiones matemáticas, con métricas y resultados de otras expresiones matemáticas en las fórmulas de las nuevas expresiones matemáticas.

10. Cuando tenga la expresión que se va a utilizar para la alarma, quite la marca de las casillas de verificación situadas a la izquierda de las otras expresiones y métricas de la página. Solo debe seleccionarse la casilla de verificación situada junto a la expresión que se va a utilizar para la alarma. La expresión que elija para la alarma debe producir una serie temporal única y mostrar solo una línea en el gráfico. A continuación, elija Select metric (Seleccionar métrica).

Aparece la página Specify metric and conditions (Especificar métrica y condiciones), en la que se muestra un gráfico y otra información acerca de la expresión matemática que ha seleccionado.

11. En Whenever ***`expresión`*** is (Siempre que la expresión sea), especifique si la expresión debe ser mayor, menor o igual que el umbral. En than... (que...), especifique el valor de umbral.
12. Elija Configuración adicional. Para Puntos de datos para alarma, especifique el número de periodos de evaluación (puntos de datos) que deben tener el estado `ALARM` para que se active la alarma. Si estos dos valores coinciden, creará una alarma que pasará al estado `ALARM` si se infringen muchos periodos consecutivos.

    Para crear una alarma M de N, especifique un número menor para el primer valor que el especificado para el segundo valor. Para obtener más información, consulte Evaluación de una alarma.
13. En Missing data treatment (Tratamiento de datos que faltan), elija cómo debe comportarse la alarma cuando falten algunos puntos de datos. Para obtener más información, consulte Configuración de la forma en la que las alarmas de CloudWatch tratan los datos que faltan.
14. Elija Siguiente.
15. En Notification (Notificación), seleccione el tema de SNS al que desee enviar la notificación cuando la alarma tenga el estado `ALARM`, `OK` o `INSUFFICIENT_DATA`.

    Para que la alarma envíe varias notificaciones para el mismo estado de alarma o para estados de alarma diferentes, seleccione Add notificación (Añadir notificación).

    Para que la alarma no envíe notificaciones, elija Remove (Eliminar).
16. Para que la alarma realice acciones de escalado automático, EC2, Lambda o de Systems Manager, elija el botón correspondiente y seleccione el estado de la alarma y la acción que se debe realizar. Si elige una función de Lambda como acción de la alarma, debe especificar el nombre de la función o el ARN y, si lo desea, puede elegir una versión específica de la función.

    Las alarmas solo pueden realizar acciones de Systems Manager cuando entran en el estado ALARMA. Para obtener más información sobre las acciones de Systems Manager, consulte Configuración de CloudWatch para crear OpsItems a partir de alarmas y Creación de incidentes.

> **Note**
>
> Para crear una alarma que realice una acción de SSM Incident Manager, debe contar con determinados permisos. Para obtener más información, consulte Ejemplos de políticas basadas en identidades del Administrador de incidentes de AWS Systems Manager.

17. Cuando haya terminado, elija Next (Siguiente).
18. Escriba un nombre y la descripción de la alarma. A continuación, elija Next.

    El nombre debe contener solo caracteres UTF-8 y no puede contener caracteres de control ASCII. La descripción puede incluir el formato Markdown, que solo se muestra en la pestaña Detalles de la alarma de la consola de CloudWatch. Markdown puede resultar útil para añadir enlaces a manuales u otros recursos internos.
19. En Obtener vista previa y crear, confirme que la información y las condiciones son las que desea y, a continuación, elija Crear alarma.

También puede agregar alarmas a un panel. Para obtener más información, consulte Agregue o elimine un widget de alarma desde un panel de CloudWatch .

Crear una alarma de CloudWatch basada en una consulta de Información de métricas

Puede crear una alarma en cualquier consulta de información de métricas que devuelva una sola serie temporal. Esto puede resultar especialmente útil para crear alarmas dinámicas que controlen las métricas agregadas de una flota de infraestructuras o aplicaciones. Cree la alarma una vez y verá que se ajusta a medida que los recursos de la flota se añaden o eliminan. Por ejemplo, puede crear una alarma que controle la utilización de la CPU de todas las instancias y que esta alarma se ajuste dinámicamente a medida que agregue o elimine instancias.

Para obtener instrucciones completas, consulte Creación de alarmas en las consultas de Información de métricas.

Creación de una alarma basada en un origen de datos conectado

Puede crear alarmas que observen las métricas de orígenes de datos que no estén en CloudWatch. Para obtener más información acerca de la creación de conexiones con los otros orígenes de datos, consulte Consulta de métricas de otros orígenes de datos.

Cómo crear una alarma en las métricas de un origen de datos al que se haya conectado

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, seleccione Métricas y, a continuación, Todas las métricas.
3. Seleccione la pestaña Consulta de múltiples orígenes.
4. En Origen de datos, seleccione el origen de datos que desee utilizar.
5. El generador de consultas le solicita la información necesaria para que la consulta recupere las métricas que se usarán en la alarma. El flujo de trabajo es diferente para cada origen de datos y se adapta a cada origen de datos. Por ejemplo, para Amazon Managed Service para Prometheus y los orígenes de datos de Prometheus, aparece un cuadro del editor de consultas de PromQL con un asistente de consultas.
6. Cuando haya terminado de crear la consulta, elija Consulta de gráficos.
7. Si el gráfico de muestra tiene el aspecto esperado, elija Crear alarma.
8. Aparecerá la página Especificar métrica y condiciones. Si la consulta que está usando produce más de una serie temporal, verá un mensaje de advertencia en la parte superior de la página. Si es así, seleccione una función para añadir las series temporales en la función de agregación.
9. (Opcional) Añada una Etiqueta para la alarma.
10. En Cuando ***`nombre-de-la-métrica`*** sea . . ., elija Mayor, Mayor/Igual, Menor/Igual o Menor. En que . . ., especifique un número para el valor del umbral.
11. Elija Configuración adicional. Para Puntos de datos para alarma, especifique el número de periodos de evaluación (puntos de datos) que deben tener el estado `ALARM` para que se active la alarma. Si estos dos valores coinciden, creará una alarma que pasará al estado `ALARM` si se infringen muchos periodos consecutivos.

    Para crear una alarma M de N, especifique un número menor para el primer valor que el especificado para el segundo valor. Para obtener más información, consulte Evaluación de una alarma.

12. En Tratamiento de datos que faltan, elija cómo debe comportarse la alarma cuando falten algunos puntos de datos. Para obtener más información, consulte Configuración de la forma en la que las alarmas de CloudWatch tratan los datos que faltan.
13. Elija Siguiente.
14. En Notificación, especifique el tema de Amazon SNS al que desee enviar la notificación cuando la alarma tenga una transición al estado `ALARM`, `OK` o `INSUFFICIENT_DATA`.

    a. (Opcional) Para que la alarma envíe varias notificaciones para el mismo estado de alarma o para estados de alarma diferentes, elija Añadir notificación.

    > **Note**
    >
    > Le recomendamos que configure la alarma para que tome medidas cuando pase al estado de datos insuficientes, además de cuando pase al estado de alarma. Esto se debe a que muchos problemas con la función de Lambda que se conecta al origen de datos pueden provocar que la alarma pase a datos insuficientes.

    b. (Opcional) Para que no envíe notificaciones de Amazon SNS, elija Eliminar.
15. Para que la alarma realice acciones de escalado automático, EC2, Lambda o de Systems Manager, elija el botón correspondiente y seleccione el estado de la alarma y la acción que se debe realizar. Si elige una función de Lambda como acción de la alarma, debe especificar el nombre de la función o el ARN y, si lo desea, puede elegir una versión específica de la función.

    Las alarmas solo pueden realizar acciones de Systems Manager cuando entran en el estado ALARMA. Para obtener más información sobre las acciones de Systems Manager, consulte Configuración de CloudWatch para crear OpsItems a partir de alarmas y Creación de incidentes.

    > **Note**
    >
    > Para crear una alarma que realice una acción de SSM Incident Manager, debe contar con determinados permisos. Para obtener más información, consulte Ejemplos de políticas basadas en identidades del Administrador de incidentes de AWS Systems Manager.
16. Elija Siguiente.
17. En Nombre y descripción, escriba el nombre y la descripción de la alarma y elija Siguiente. El nombre debe contener solo caracteres UTF-8 y no puede contener caracteres de control ASCII.

La descripción puede incluir el formato Markdown, que solo se muestra en la pestaña Detalles de la alarma de la consola de CloudWatch. Markdown puede resultar útil para añadir enlaces a manuales u otros recursos internos.

> **Tip**
>
> El nombre de alarma solo debe contener caracteres UTF-8. No puede contener caracteres de control ASCII.

18. En Obtener vista previa y crear, confirme que la información y las condiciones son las correctas y luego, elija Crear alarma.

Detalles sobre las alarmas de los orígenes de datos conectados

- Cuando CloudWatch evalúa una alarma, lo hace cada minuto, incluso si el período de la alarma es superior a un minuto. Para que la alarma funcione, la función de Lambda debe poder devolver una lista de marcas temporales que comiencen en cualquier minuto, no solo en múltiplos de la duración del período. Estas marcas temporales deben estar espaciadas a una distancia de un período.

  Por lo tanto, si el origen de datos consultado por Lambda solo puede devolver marcas temporales que sean múltiplos de la longitud del período, la función debería "volver a muestrear" los datos obtenidos para que coincidan con las marcas temporales esperadas por la solicitud `GetMetricData`.

  Por ejemplo, una alarma con un período de cinco minutos se evalúa cada minuto mediante ventanas de cinco minutos que cambian un minuto cada vez. En este caso:

  - Para la evaluación de la alarma a las 12:15:00, CloudWatch espera puntos de datos con marcas de tiempo de `12:00:00`, `12:05:00` y `12:10:00`.
  - Luego, para la evaluación de la alarma a las 12:16:00, CloudWatch espera puntos de datos con marcas de tiempo de `12:01:00`, `12:06:00` y `12:11:00`.
- Cuando CloudWatch evalúa una alarma, todos los puntos de datos devueltos por la función de Lambda que no se alineen con las marcas temporales esperadas se descartan y la alarma se evalúa utilizando los puntos de datos esperados restantes. Por ejemplo, cuando la alarma se evalúa a las `12:15:00`, se esperan datos con marcas de tiempo de `12:00:00`, `12:05:00` y `12:10:00`. Si recibe datos con marcas de tiempo de `12:00:00`, `12:05:00`, `12:06:00` y `12:10:00`, los datos `12:06:00` se descartan y CloudWatch evalúa la alarma utilizando las demás marcas de tiempo.

Luego, para la siguiente evaluación a las `12:16:00`, se esperan datos con marcas de tiempo de `12:01:00`, `12:06:00` y `12:11:00`. Si solo tiene los datos con marcas de tiempo iguales a `12:00:00`, `12:05:00` y `12:10:00`, todos estos puntos de datos se ignoran a las 12:16:00 y la alarma pasa al estado según el modo en que especificó la alarma para tratar los datos faltantes. Para obtener más información, consulte Evaluación de una alarma.

- Le recomendamos que cree estas alarmas para tomar medidas cuando pasen al estado `INSUFFICIENT_DATA`, ya que varios casos de uso de fallas de la función de Lambda harán la transición de la alarma a `INSUFFICIENT_DATA`, independientemente de la forma en que la configure para tratar los datos faltantes.
- Si la función de Lambda devuelve un error o devuelve datos parciales:
  - Si hay un problema de permisos al llamar a la función de Lambda, la alarma comienza a tener transiciones de datos faltantes según la forma en que especificó la alarma para tratar los datos faltantes cuando la creó.
  - Si la función de Lambda devuelve `'StatusCode' = 'PartialData'`, la evaluación de la alarma falla y la alarma pasa a `INSUFFICIENT_DATA` después de tres intentos. Esto tarda unos tres minutos.
  - Cualquier otro error que provenga de la función de Lambda hace que la alarma pase a `INSUFFICIENT_DATA`.
- Si la métrica solicitada por la función de Lambda presenta algún retraso, por lo que siempre falta el último punto de datos, debe utilizar una solución alternativa. Puede crear una alarma M a partir de N o aumentar el período de evaluación de la alarma. Para obtener más información sobre alarmas M a partir de N, consulte Evaluación de una alarma.

Crear una alarma de CloudWatch en función de la detección de anomalías

Puede crear una alarma basada en la detección de anomalías de CloudWatch, que extrae datos métricos antiguos y crea un modelo de valores esperados. Los valores esperados tienen en cuenta en la métrica los patrones horario, diario o semanal típicos.

Se establece un valor para el umbral de detección de anomalías y CloudWatch utiliza este umbral con el modelo para determinar el intervalo ‘normal’ de valores de la métrica. Un valor mayor del umbral produce un intervalo mayor de valores “normales”.

Puede elegir si la alarma se activa cuando el valor de la métrica está por encima de la banda de valores previstos, por debajo de la banda, o bien por encima o por debajo de la banda.

También puede crear alarmas de detección de anomalías en métricas individuales y en los resultados de expresiones matemáticas métricas. Puede utilizar estas expresiones para crear gráficos en los que se visualicen bandas de detección de anomalías.

En una cuenta configurada como cuenta de supervisión para la observabilidad entre cuentas de CloudWatch, puede crear detectores de anomalías en las métricas de las cuentas de origen, además de las métricas de la cuenta de supervisión.

Para obtener más información, consulte Uso de la detección de anomalías de CloudWatch.

> **Note**
>
> Si crea una alarma de detección de anomalías en una métrica que ya utiliza para la detección de anomalías en la consola de métricas para fines de visualización, el umbral que haya establecido para la alarma no cambia el umbral que ya utiliza para la visualización. Para obtener más información, consulte Creación de un gráfico.

Para crear una alarma basándose en la detección de anomalías

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Alarms (Alarmas) y, luego, Create Alarm (Crear alarma).
3. Elija Create alarm (Crear alarma).
4. Elija Select Metric (Seleccionar métrica).
5. Realice una de las siguientes acciones siguientes:
   - Elija el espacio de nombres de servicio que contiene la métrica y, a continuación, siga eligiendo las opciones a medida que parecen restringir las opciones. Cuando aparezca una lista de métricas, seleccione la casilla de verificación situada junto a su métrica.
   - En el campo de búsqueda, ingrese el nombre de una métrica, dimensión o ID de recurso. A continuación, seleccione uno de los resultados y continúe hasta que se muestren las opciones mientras van apareciendo hasta ver una lista de métricas. Seleccione la casilla de verificación situada junto a su métrica.
6. Elija Métricas diagramadas.

a. (Opcional) Bajo la columna Statistic (Estadística), seleccione el menú desplegable y luego seleccione uno de los percentiles o estadísticas predefinidos. Utilice el cuadro de búsqueda del menú desplegable para especificar un percentil personalizado, como **`p95.45`**.

b. (Opcional) En la columna Period (Período), seleccione el menú desplegable y luego seleccione uno de los periodos de evaluación predefinidos.

> **Note**
>
> Cuando CloudWatch evalúa la alarma, agrupa el período en un único punto de datos. Para la alarma de detección de anomalías, el período de evaluación debe ser un minuto o más.

7. Elija Siguiente.
8. En Conditions (Condiciones), especifique lo siguiente:

a. Elija Anomaly detection (Detección de anomalías).

Si el modelo para esta métrica y estadística ya existe, CloudWatch muestra una vista previa de la banda de detección de anomalías en el gráfico de la parte superior de la pantalla. Después de crear la alarma, la banda de detección de anomalías real puede tardar hasta 15 minutos en aparecer en el gráfico. Antes de eso, la banda que verá será una aproximación de la banda de detección de anomalías.

> **Tip**
>
> Para ver el gráfico en la parte superior de la pantalla en un periodo de tiempo más largo, elija Edit (Editar) en la parte superior derecha de la pantalla.

Si el modelo para esta métrica y estadística no existe, CloudWatch genera la banda de detección de anomalías cuando termina de crear la alarma. Para los modelos nuevos, la banda de detección de anomalías real puede tardar hasta 3 horas en aparecer en el gráfico. El nuevo modelo puede tardar hasta dos semanas en entrenarse, por lo que la banda de detección de anomalías muestra valores esperados más precisos.

b. Para cuando sea que se configure la ***`métrica`***, especifique cuándo se debe desencadenar la alarma. Por ejemplo, cuando la métrica es mayor, inferior o está fuera de la banda (en cualquier dirección).

c. En Anomaly detection threshold (Umbral de detección de anomalías), elija el número que desea utilizar para el umbral de detección de anomalías. Un número mayor crea una banda más gruesa de valores "normales" que es más tolerante a los cambios de métrica. Un número menor crea una banda más delgada que pasará al estado `ALARM` con desviaciones métricas más pequeñas. El número no tiene que ser un número entero.

d. Elija Configuración adicional. Para Puntos de datos para alarma, especifique el número de periodos de evaluación (puntos de datos) que deben tener el estado `ALARM` para que se active la alarma. Si estos dos valores coinciden, creerá una alarma que pasará al estado `ALARM` si se infringen muchos periodos consecutivos.

   Para crear una alarma M de N, especifique un número menor para el primer valor que el especificado para el segundo valor. Para obtener más información, consulte Evaluación de una alarma.

e. En Tratamiento de datos que faltan, elija cómo debe comportarse la alarma cuando falten algunos puntos de datos. Para obtener más información, consulte Configuración de la forma en la que las alarmas de CloudWatch tratan los datos que faltan.

f. Si la alarma utiliza un percentil como estadística supervisada, aparece un cuadro Percentiles with low samples (Percentiles con pocas muestras). Utilícelo para seleccionar si desea evaluar o no tener en cuenta los casos con frecuencias de muestreo bajas. Si elige Ignore (maintain alarm state) (Ignorar (mantener el estado de alarma)), el estado de alarma actual se mantiene siempre cuando el tamaño de la muestra es demasiado bajo. Para obtener más información, consulte Muestras de datos reducidos y alarmas de CloudWatch basadas en percentiles.

9. Elija Siguiente.

10. En Notification (Notificación), seleccione el tema de SNS al que desee enviar la notificación cuando la alarma tenga el estado `ALARM`, `OK` o `INSUFFICIENT_DATA`.

    Para que la alarma envíe varias notificaciones para el mismo estado de alarma o para estados de alarma diferentes, elija Add notificación (Agregar notificación).

    Elija Remove (Eliminar) si no desea que la alarma envíe notificaciones.

11. Puede configurar la alarma para que realice acciones de EC2 o invocar una función de Lambda cuando cambia de estado o para crear un OpsItem o incidente de Systems Manager cuando entra en estado ALARMA. Para que la alarma haga esto, elija el botón correspondiente y luego el estado de alarma y la acción que se debe realizar.

Si elige una función de Lambda como acción de la alarma, debe especificar el nombre de la función o el ARN y, si lo desea, puede elegir una versión específica de la función.

Para obtener más información sobre las acciones de Systems Manager, consulte Configuración de CloudWatch para crear OpsItems desde alarmas y Creación de incidentes.

> **Note**
>
> Para crear una alarma que realice una acción de AWS Systems Manager Incident Manager, debe contar con determinados permisos. Para obtener más información, consulte Ejemplos de políticas basadas en identidades del Administrador de incidentes de AWS Systems Manager.

12. Elija Siguiente.
13. En Nombre y descripción, escriba el nombre y la descripción de la alarma y elija Siguiente. El nombre debe contener solo caracteres UTF-8 y no puede contener caracteres de control ASCII. La descripción puede incluir el formato Markdown, que solo se muestra en la pestaña Detalles de la alarma de la consola de CloudWatch. Markdown puede resultar útil para añadir enlaces a manuales u otros recursos internos.

> **Tip**
>
> El nombre de la alarma debe contener únicamente caracteres UTF-8 y no puede contener caracteres de control ASCII

14. En Obtener vista previa y crear, confirme que la información y las condiciones son las correctas y luego, elija Crear alarma.

Modificación de un modelo de detección de anomalías

Una vez que haya creado una alarma, puede ajustar el modelo de detección de anomalías. Puede excluir determinados períodos de tiempo para que no se utilicen en la creación del modelo. Es fundamental que excluya eventos inusuales, como interrupciones del sistema, implementaciones y días festivos, de los datos de entrenamiento. También puede especificar si desea ajustar el modelo para los cambios de horario de verano.

Para ajustar el modelo de detección de anomalías para una alarma

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Alarms (Alarmas) y, luego, Create Alarm (Crear alarma).
3. Elija el nombre de la alarma. Si es necesario, utilice el cuadro de búsqueda para encontrar la alarma.
4. Elija Analizar, En métricas.
5. En la columna Detalles, elija ANOMALY_DETECTION_BAND y Editar modelo de detección de anomalías.
6. Para excluir un periodo de tiempo de la elaboración del modelo, elija el icono de calendario por Fecha final. Luego, seleccione o ingrese los días y las horas que desea excluir de la formación y elija Apply (Aplicar).
7. Si la métrica es sensible a los cambios de horario de verano, seleccione la zona horaria adecuada en el cuadro Metric timezone (Zona horaria de métricas).
8. Elija Update (Actualizar).

Eliminación de un modelo de detección de anomalías

Usar detección de anomalías para una alarma acumula cargos de . Como práctica recomendada, si la alarma ya no necesita un modelo de detección de anomalías, elimine la alarma primero y luego el modelo. Cuando se evalúan las alarmas de detección de anomalías, se crean en su nombre los detectores de anomalías que faltan. Si elimina el modelo sin eliminar la alarma, la alarma vuelve a crear automáticamente el modelo.

Eliminación de una alarma

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Alarms (Alarmas) y, a continuación, All Alarms (Todas las alarmas).
3. Elija el nombre de la alarma.
4. Elija Acciones, Eliminar.
5. En el cuadro de confirmación, elija Delete (Eliminar).

Para eliminar un modelo de detección de anomalías que se había utilizado para una alarma

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Metrics (Métricas) y, a continuación, All metrics (Todas las métricas).
3. Elija Browse (Navegar) y, a continuación, seleccione la métrica que incluye el modelo de detección de anomalías. Puede buscar la métrica en el cuadro de búsqueda o elegir entre las opciones para seleccionarla.
   - (Opcional) Si utiliza la interfaz original, seleccione All metrics (Todas las métricas) y, a continuación, elija la métrica que incluye el modelo de detección de anomalías. Puede buscar la métrica en el cuadro de búsqueda o elegir entre las opciones para seleccionarla.
4. Elija la pestaña Métricas diagramadas.
5. En la pestaña Métricas diagramadas, elija el nombre del modelo de detección de anomalías que desea eliminar y, a continuación, seleccione Delete anomaly detection model (Eliminar el modelo de detección de anomalías).
   - (Opcional) Si utiliza la interfaz original, elija Edit model (Editar el modelo). Esto lo llevará a una pantalla nueva. En la nueva pantalla, elija Delete model (Eliminar modelo) y, a continuación, seleccione Delete (Eliminar).

Alarma en los registros

Los pasos de las siguientes secciones explican cómo crear alarmas de CloudWatch en los registros.

Crear una alarma de CloudWatch basada en un filtro por métricas del grupo de registro

El procedimiento de esta sección describe cómo crear una alarma basada en un filtro por métricas del grupo de registro. Con los filtros por métricas, puede buscar los términos y patrones de los datos del registro, ya que los datos se envían a CloudWatch. Para obtener más información, consulte Crear métricas a partir de eventos de registro mediante filtros en la Guía del usuario de Registros de Amazon CloudWatch. Antes de crear una alarma basada en un filtro por métricas del grupo de registro, debe llevar a cabo las siguientes acciones:

- Crear un grupo de registros. Para obtener más información, consulte Trabajar con grupos de registros y flujos de registros en la Guía del usuario de Registros de Amazon CloudWatch.

- Crear un filtro de métricas. Para obtener más información, consulte Crear un filtro de métricas para un grupo de registros en la Guía del usuario de Registros de Amazon CloudWatch.

Para crear una alarma basada en un filtro por métricas del grupo de registro

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Logs (Registros) y, luego, Log groups (Grupos de registro).
3. Elija el grupo de registro que incluye el filtro de métricas.
4. Elija Metric filters (Filtros de métricas).
5. En la pestaña de filtros de métricas, seleccione la casilla del filtro de métricas en el que quiera basar la alarma.
6. Elija Crear alarma.
7. (Opcional) En Metric (Métrica), modifique Metric name (Nombre de métrica), Statistic (Estadística) y Period (Periodo).
8. En Conditions (Condiciones), especifique lo siguiente:
   a. En Threshold type (Tipo de límite), elija Static (Estático) o Anomaly detection (Detección de anomalías).
   b. En Whenever ***your-metric-name*** is . . . (Cuando nombre-de-la-métrica sea...), elija Greater (Mayor), Greater/Equal (Mayor/igual), Lower/Equal (Menor/igual) o Lower (Menor).
   c. En Than . . . (Que...), especifique un número para el valor del umbral.
9. Elija Configuración adicional.
   a. En Data points to alarm (Puntos de datos para avisar), especifique cuántos puntos de datos activan la alarma para que pase al estado `ALARM`. Si especifica valores coincidentes, la alarma pasa al estado `ALARM` si se infringen muchos periodos consecutivos. Para crear una alarma M de N, especifique un número menor para el primer valor que el especificado para el segundo valor. Para obtener más información, consulte Uso de las alarmas de Amazon CloudWatch.
   b. En Missing data treatment (Tratamiento de datos faltantes), especifique una opción para especificar cómo tratar los datos que faltan cuando se evalúe la alarma.
10. Elija Siguiente.
11. En Notification (Notificación), especifique el tema de Amazon SNS que se va a notificar cuando el estado de la alarma sea `ALARM`, `OK` o `INSUFFICIENT_DATA`.

a. (Opcional) Para que la alarma envíe varias notificaciones para el mismo estado de alarma o para estados de alarma diferentes, elija Add notification (Agregar notificación).

b. (Opcional) Para que no envíe notificaciones, elija Remove (Eliminar).

12. Para que la alarma realice acciones de escalado automático, EC2, Lambda o de Systems Manager, elija el botón correspondiente y seleccione el estado de la alarma y la acción que se debe realizar. Si elige una función de Lambda como acción de la alarma, debe especificar el nombre de la función o el ARN y, si lo desea, puede elegir una versión específica de la función.

    Las alarmas solo pueden realizar acciones de Systems Manager cuando entran en el estado ALARMA. Para obtener más información sobre las acciones de Systems Manager, consulte Configuración de CloudWatch para crear OpsItems a partir de alarmas y Creación de incidentes.

    > **Note**
    >
    > Para crear una alarma que realice una acción de SSM Incident Manager, debe contar con determinados permisos. Para obtener más información, consulte Ejemplos de políticas basadas en identidades del Administrador de incidentes de AWS Systems Manager.

13. Elija Siguiente.

14. En Name and description (Nombre y descripción), ingrese un nombre y una descripción para la alarma. El nombre debe contener solo caracteres UTF-8 y no puede contener caracteres de control ASCII. La descripción puede incluir el formato Markdown, que solo se muestra en la pestaña Detalles de la alarma de la consola de CloudWatch. Markdown puede resultar útil para añadir enlaces a manuales u otros recursos internos.

15. En Ver la vista previa y crear, compruebe que la configuración sea correcta y elija Crear alarma.

Combinación de alarmas

Con CloudWatch, puede combinar varias alarmas en una alarma compuesta para crear un indicador de estado agregado y resumido para toda una aplicación o un grupo de recursos. Las alarmas compuestas son alarmas que determinan su estado mediante la supervisión de los estados de otras alarmas. Puede definir las reglas para combinar el estado de las alarmas supervisadas mediante la lógica booleana.

Puede utilizar alarmas compuestas para reducir el ruido de las alarmas mediante la toma de acciones a nivel agregado. Por ejemplo, puede crear una alarma compuesta para enviar una notificación al equipo de su servidor web si se activa alguna alarma relacionada a este. Cuando una de esas alarmas pasa al estado de ALARMA, la alarma compuesta pasa por sí misma al estado de ALARMA y envía una notificación a su equipo. Si otras alarmas relacionadas con el servidor web también pasan al estado de ALARMA, su equipo no se verá sobrecargado con notificaciones nuevas, ya que la alarma compuesta ya lo notificó sobre la situación existente.

También puede utilizar alarmas compuestas para crear condiciones de alarma complejas y tomar medidas solo cuando se cumplan numerosas condiciones diferentes. Por ejemplo, puede crear una alarma compuesta que combine una alarma de CPU y una alarma de memoria y que solo notifique al equipo si se activan tanto la alarma de la CPU como la de la memoria.

Uso de alarmas compuestas

Cuando utiliza alarmas compuestas, tiene dos opciones:

- Configure las acciones que desee realizar solo en el nivel de alarma compuesta y cree las alarmas supervisadas subyacentes sin necesidad de realizar ninguna acción
- Configure un conjunto diferente de acciones a nivel de alarma compuesta. Por ejemplo, las acciones de la alarma compuesta podrían involucrar a un equipo diferente en caso de que el problema se generalice.

Las alarmas compuestas solo pueden realizar las siguientes acciones:

- Notificar a los temas de Amazon SNS
- Invocación de funciones de Lambda
- Crear OpsItems en el Centro de operaciones Systems Manager
- Crear incidentes en el Administrador de incidentes de Systems Manager

> **Note**
>
> Todas las alarmas subyacentes de la alarma compuesta deben estar en la misma cuenta y región que su alarma compuesta. Sin embargo, si configura una alarma compuesta en una cuenta de supervisión para la observabilidad entre cuentas de CloudWatch, las alarmas subyacentes pueden observar las métricas de diferentes cuentas de origen y en la propia

cuenta de supervisión. Para obtener más información, consulte Observabilidad entre cuentas de CloudWatch.
Una sola alarma compuesta puede supervisar 100 alarmas subyacentes, y 150 alarmas compuestas pueden supervisar una sola alarma subyacente.

Expresiones de regla

Todas las alarmas compuestas contienen expresiones de regla. Las expresiones de regla indican a las alarmas compuestas qué otras alarmas deben supervisar y determinar sus estados. Las expresiones de regla puede hacer referencia a alarmas de métricas y a alarmas compuestas. Cuando hace referencia a una alarma en una expresión de regla, designa una función para la alarma que determina en cuál de los tres estados siguientes estará la alarma:

- ALARM

  ALARM (alarm-name o alarm-ARN) se establece en TRUE si la alarma tiene el estado ALARM (ALARMA).
- OK (Correcto)

  OK (alarm-name o alarm-ARN) se establece en TRUE si la alarma tiene el en estado OK (CORRECTO).
- INSUFICIENT_DATA

  INSUFFICIENT_DATA (alarm-name o alarm-ARN) se establece en TRUE si la alarma con nombre tiene el estado INSUFFICIENT_DATA (DATOS INSUFICIENTES).

> **Note**
>
> TRUE siempre se evalúa como TRUE (VERDADERO), y FALSE siempre se evalúa como FALSE (FALSO).

Expresiones de ejemplo

El parámetro de solicitud `AlarmRule` admite el uso de los operadores lógicos `AND`, `OR` y `NOT` para que pueda combinar varias funciones en una sola expresión. Las siguientes expresiones de ejemplo muestran cómo puede configurar las alarmas subyacentes en la alarma compuesta:

- `ALARM(CPUUtilizationTooHigh) AND ALARM(DiskReadOpsTooHigh)`

  La expresión especifica que la alarma compuesta entra en `ALARM` solo si `CPUUtilizationTooHigh` y `DiskReadOpsTooHigh` están en `ALARM`.

- `ALARM(CPUUtilizationTooHigh) AND NOT ALARM(DeploymentInProgress)`

  La expresión especifica que la alarma compuesta entra en `ALARM` si `CPUUtilizationTooHigh` está en `ALARM` y `DeploymentInProgress` no está en `ALARM`. He aquí un ejemplo de alarma compuesta que reduce el ruido de la alarma durante una ventana de implementación.

- `(ALARM(CPUUtilizationTooHigh) OR ALARM(DiskReadOpsTooHigh)) AND OK(NetworkOutTooHigh)`

  La expresión especifica que la alarma compuesta entra en `ALARM` si `(ALARM(CPUUtilizationTooHigh)` o `(DiskReadOpsTooHigh)` están en `ALARM` y `(NetworkOutTooHigh)` está en `OK`. He ahora un ejemplo de alarma compuesta que reduce el ruido de la alarma al no enviarle notificaciones cuando alguna de las alarmas subyacentes no está en `ALARM` mientras se produce un problema de red.

Temas

Crear una alarma compuesta

En los pasos de esta sección, se explica cómo usar la consola de CloudWatch para crear una alarma compuesta. También puede utilizar la API o la AWS CLI para crear una alarma compuesta. Para obtener más información, consulte PutCompositeAlarm o put-composite-alarm.

Para crear una alarma compuesta

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Alarms (Alarmas) y, a continuación, elija All Alarms (Todas las alarmas).
3. En la lista de alarmas, seleccione la casilla de verificación situada junto a cada una de las alarmas existentes a las que desee hacer referencia en su expresión de regla, y luego seleccione Create composite alarm (Crear alarma compuesta).

4. En Specify composite alarm conditions (Especificación de condiciones de alarma compuesta), especifique la expresión de regla de la nueva alarma compuesta.

> **Note**
>
> Las alarmas que haya seleccionado de la lista de alarmas se mostrarán automáticamente en el cuadro de Conditions (Condiciones). De forma predeterminada, se ha asignado la función `ALARM` a cada una de sus alarmas, y el operador lógico `OR` se une a cada una de sus alarmas.

Puede realizar los siguientes pasos secundarios para modificar la expresión de regla:

a. Puede cambiar el estado requerido para cada una de las alarmas de `ALARM` a `OK` o `INSUFFICENT_DATA`.

b. Puede cambiar el operador lógico de la expresión de regla de `OR` a `AND` o `NOT` y agregar paréntesis para agrupar las funciones.

c. Puede incluir otras alarmas en la expresión de regla o eliminar alarmas de la misma.

Ejemplo: expresión de regla con condiciones

```
(ALARM("CPUUtilizationTooHigh") OR
ALARM("DiskReadOpsTooHigh")) AND
OK("NetworkOutTooHigh")
```

En la expresión de regla de ejemplo en la que la alarma compuesta entra en `ALARM` cuando ALARM (“CPUUtilizationTooHigh” o ALARM(“DiskReadOpsTooHigh”) está en `ALARM` al mismo tiempo que OK (“NetworkOutTooHigh”) está en `OK`.

5. Cuando haya terminado, elija Next (Siguiente).

6. En Configure actions (Configuración de acciones), puede elegir entre las siguientes opciones:

Para Notification (Notificaciones)

- Select an exisiting SNS topic (Seleccionar un tema de SNS existente), Create a new SNS topic (Crear un nuevo tema de SNS), o bien Use a topic ARN (Usar un ARN de tema) para definir el tema de SNS que recibirá la notificación.

- Add notification (Agregar notificación), para que la alarma pueda enviar varias notificaciones para el mismo estado de alarma o para estados de alarma diferentes.
- Remove (Eliminar), para evitar que la alarma envíe notificaciones o realice acciones.

(Opcional) Para que la alarma invoque una función de Lambda cuando cambie de estado, elija Añadir acción de Lambda. A continuación, especifique el nombre de la función o el ARN y, si lo desea, elija una versión específica de la función.

Para Systems Manager action (Acción del Systems Manager)

- Add Systems Manager action (Agregar acción de Systems Manager), para que la alarma pueda realizar una acción SSM cuando entre en ALARM.

Para obtener más información sobre las acciones de Systems Manager, consulte Configuración de CloudWatch para crear OpsItems desde alarmas en la Guía del usuario de AWS Systems Manager e Incident creation (Creación de incidencias) en la Guía del usuario del Administrador de incidentes. Para crear una alarma que realice una acción de SSM Incident Manager, debe contar con los permisos correctos. Para obtener más información, consulte Ejemplos de políticas basadas en identidades de AWS Systems Manager Inciden Manager en la Guía del usuario de Incident Manager.

7. Cuando haya terminado, elija Next (Siguiente).
8. En Add name and description (Agregar nombre y descripción), introduzca un nombre de alarma y una descripción opcional para su nueva alarma compuesta. El nombre debe contener solo caracteres UTF-8 y no puede contener caracteres de control ASCII. La descripción puede incluir el formato Markdown, que solo se muestra en la pestaña Detalles de la alarma de la consola de CloudWatch. Markdown puede resultar útil para añadir enlaces a manuales u otros recursos internos.
9. Cuando haya terminado, elija Next (Siguiente).
10. En Ver vista previa y crear, confirme la información y, a continuación, seleccione Crear alarma compuesta.

> **Note**
>
> Puede crear un ciclo de alarmas compuestas en el que una alarma compuesta y otra alarma compuesta dependan la una de la otra. Si se da esta situación, sus alarmas compuestas dejarán de evaluarse y no podrá eliminar las alarmas compuestas al

tener dependencia mutua. La forma más fácil de romper el ciclo de dependencia entre alarmas compuestas consiste en cambiar la función `AlarmRule` en una de sus alarmas compuestas a `False`.

Supresión de acciones de las alarmas compuestas

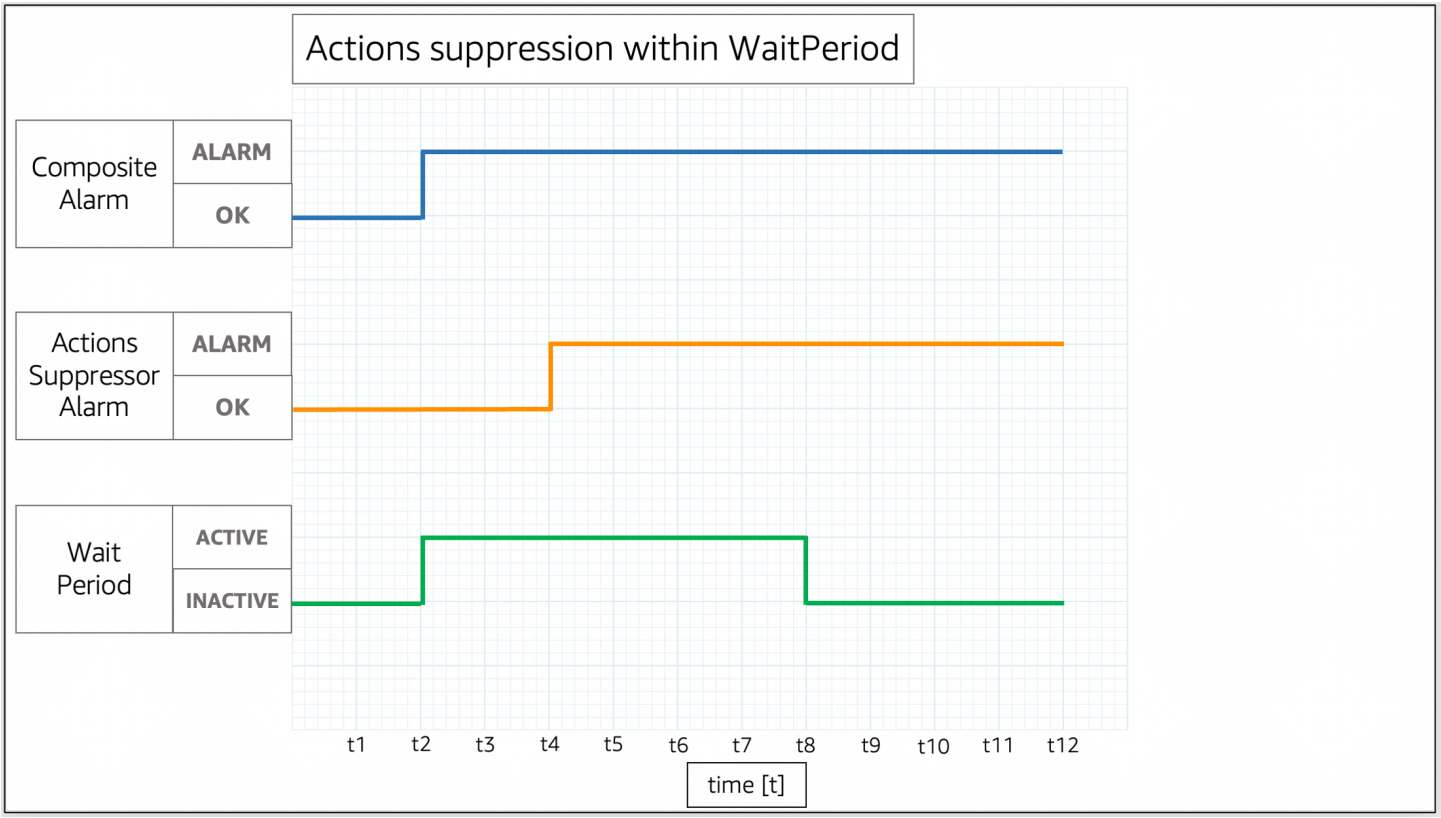
Como las alarmas compuestas permiten obtener una visión global de su estado de salud en varias alarmas, hay situaciones comunes en las que se espera que se activen esas alarmas. Por ejemplo, durante un período de mantenimiento de la aplicación o cuando investiga un incidente en curso. En estas situaciones, es posible que desee suprimir las acciones de sus alarmas compuestas para evitar notificaciones no deseadas o la creación de nuevos tickets de incidentes

Con la supresión de acciones de alarma compuesta, las alarmas se definen como alarmas supresoras. Las alarmas supresoras evitan que las alarmas compuestas realicen acciones. Por ejemplo, puede especificar una alarma supresora que represente el estado de un recurso de apoyo. Si el recurso de apoyo está inactivo, la alarma supresora impide que la alarma compuesta envíe notificaciones. La supresión de acciones de alarma compuesta le ayuda a reducir el ruido de las alarmas, para que pueda dedicar menos tiempo a administrar las alarmas y más a centrarse en sus operaciones.

Las alarmas supresoras quedan especificadas al configurar las alarmas compuestas. Cualquier alarma puede funcionar a modo de alarma supresora. Cuando una alarma supresora cambia de estado `OK` a `ALARM`, su alarma compuesta deja de realizar acciones. Cuando una alarma supresora cambia de estado `ALARM` a `OK`, su alarma compuesta reanuda las acciones.

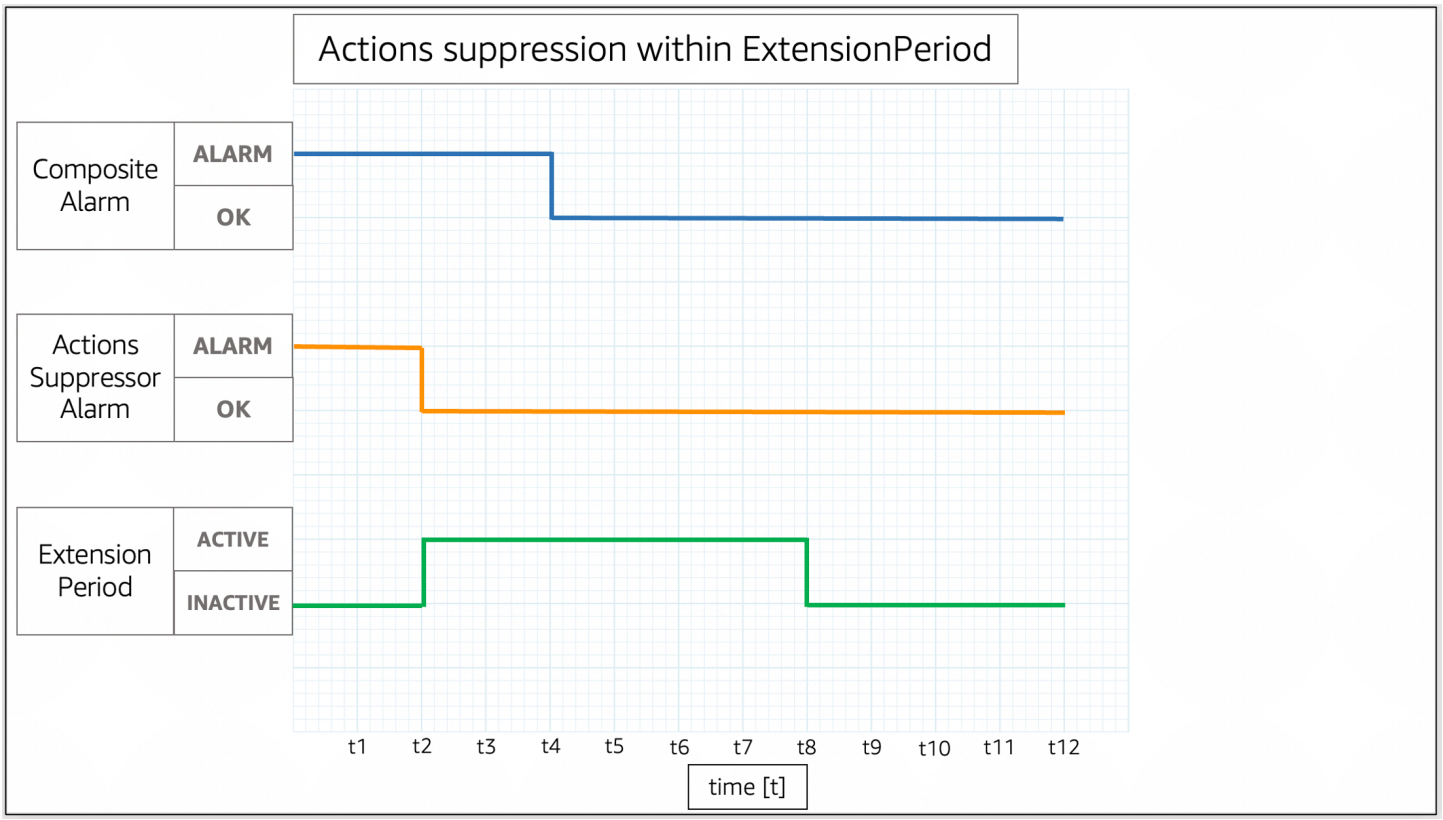
`WaitPeriod` y `ExtensionPeriod`

Cuando especifica una alarma supresora, se establecen los parámetros `WaitPeriod` y `ExtensionPeriod`. Estos parámetros evitan que las alarmas compuestas realicen acciones inesperadas mientras las alarmas supresoras cambian de estado. Use `WaitPeriod` para compensar cualquier retraso que pueda ocurrir cuando una alarma supresora cambie de `OK` a `ALARM`. Por ejemplo, si una alarma supresora cambia de `OK` a `ALARM` en un intervalo de 60 segundos, configure `WaitPeriod` en 60 segundos.



En la imagen, la alarma compuesta cambia de OK a ALARM en t2. Un `WaitPeriod` comienza en t2 y termina en t8. Esto da tiempo a la alarma supresora para cambiar los estados de OK a ALARM en t4 antes de suprimir las acciones de la alarma compuesta cuando el `WaitPeriod` expira en t8.

Use `ExtensionPeriod` para compensar cualquier retraso que pueda ocurrir cuando una alarma compuesta cambie a OK después de que una alarma supresora haga lo propio a OK. Por ejemplo, si una alarma compuesta cambia a OK en un intervalo de 60 segundos desde que una alarma supresora cambia a OK, establezca `ExtensionPeriod` en 60 segundos.



En la imagen, la alarma supresora cambia de `ALARM` a `OK` en t2. Un `ExtensionPeriod` comienza en t2 y termina en t8. Esto da a la alarma compuesta el tiempo necesario para cambiar de `ALARM` a `OK` antes de que el `ExtensionPeriod` expire en t8.

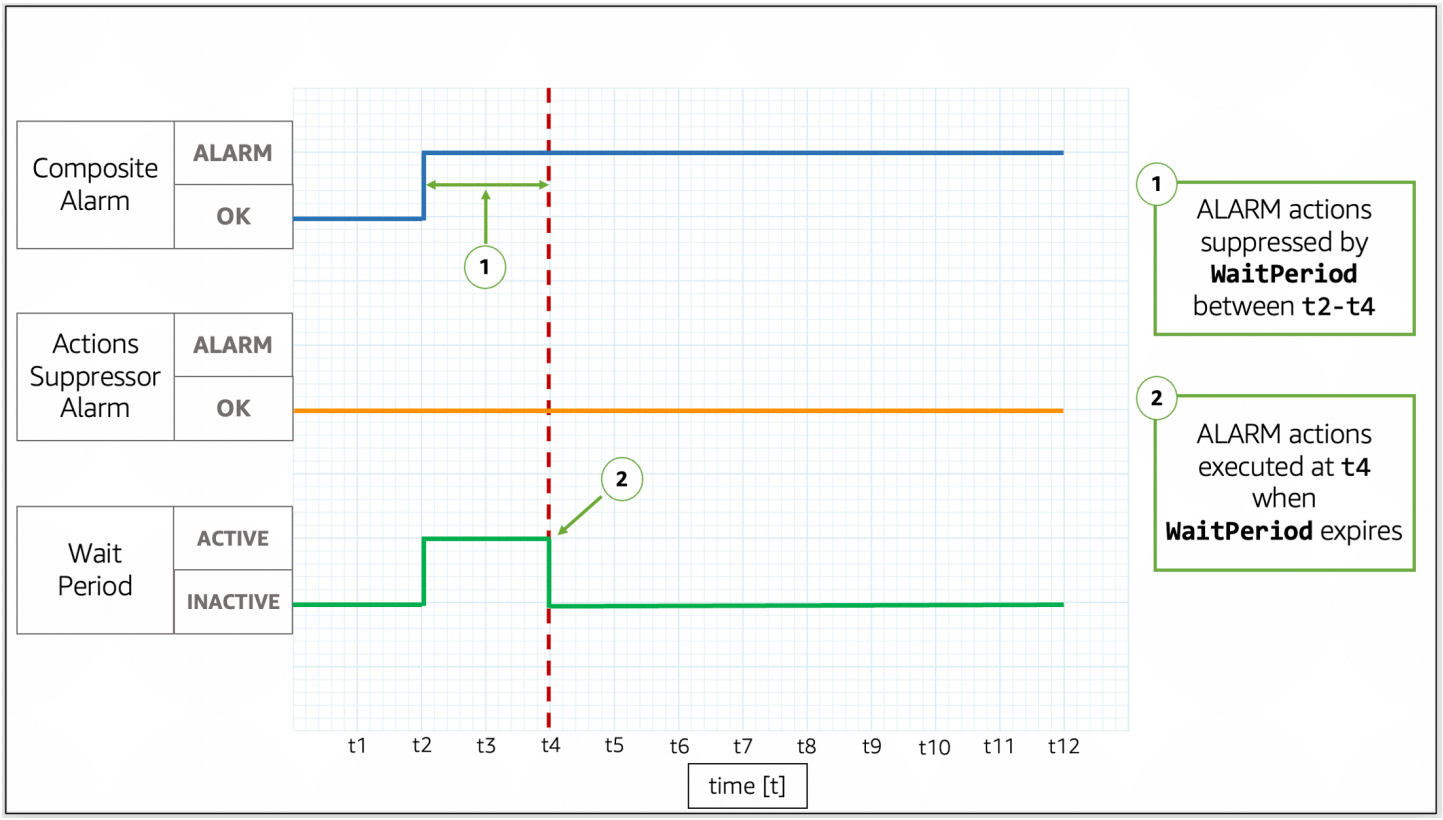
Las alarmas compuestas no realizan acciones cuando `WaitPeriod` y `ExtensionPeriod` se vuelven activos. Las alarmas compuestas realizan acciones que se basan en sus estados de corriente cuando `ExtensionPeriod` y `WaitPeriod` se vuelven inactivos. Le recomendamos que establezca el valor de cada parámetro en 60 segundos, ya que CloudWatch evalúa las alarmas métricas cada minuto. Puede establecer los parámetros en cualquier número entero en segundos.

Los siguientes ejemplos describen con más detalle cómo `WaitPeriod` y `ExtensionPeriod` evitan que las alarmas compuestas realicen acciones inesperadas.

> **Note**
>
> En los siguientes ejemplos, `WaitPeriod` está configurado como 2 unidades de tiempo, y `ExtensionPeriod` está configurado como 3 unidades de tiempo.

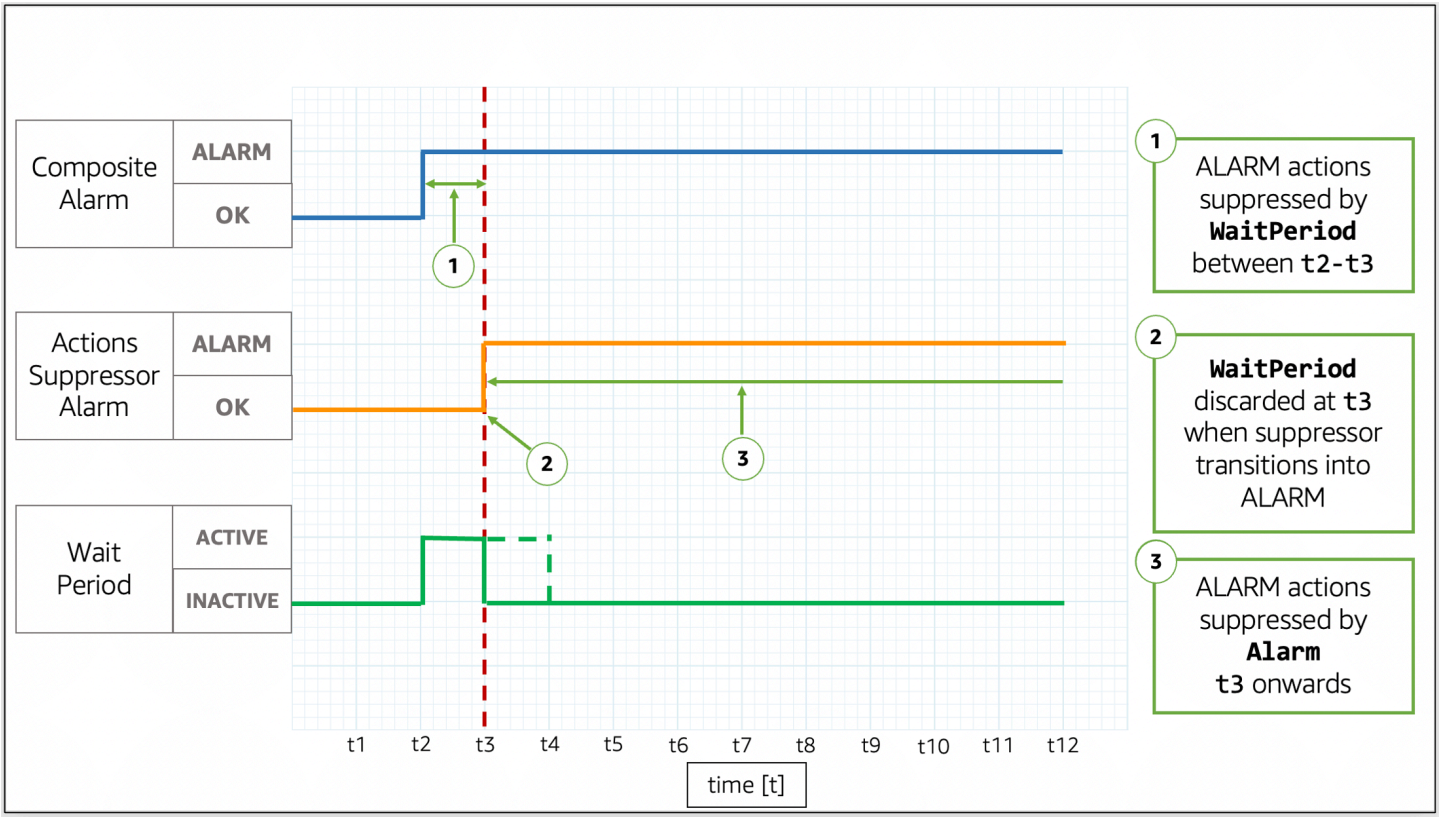
Ejemplos

Ejemplo 1: las acciones no se suprimen después del **`WaitPeriod`**



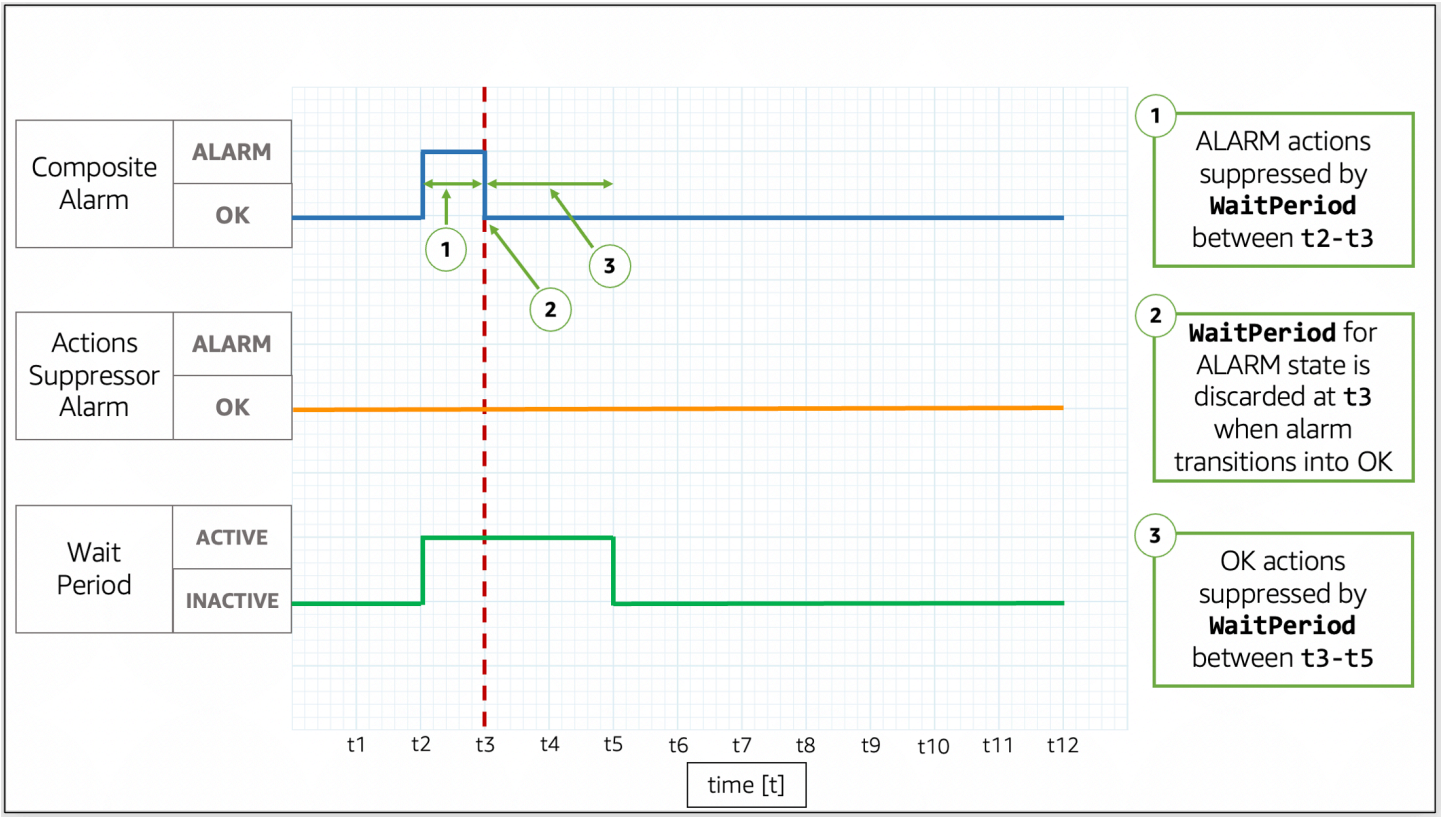
En la imagen, la alarma compuesta cambia los estados de `OK` a `ALARM` en t2. Un `WaitPeriod` comienza en t2 y termina en t4, por lo que puede evitar que la alarma compuesta realice acciones. Una vez que el `WaitPeriod` expire en t4, la alarma compuesta realizará sus acciones, ya que la alarma supresora todavía está en `OK`.

Ejemplo 2: las acciones se suprimen mediante alarma antes de que el **`WaitPeriod`** expire



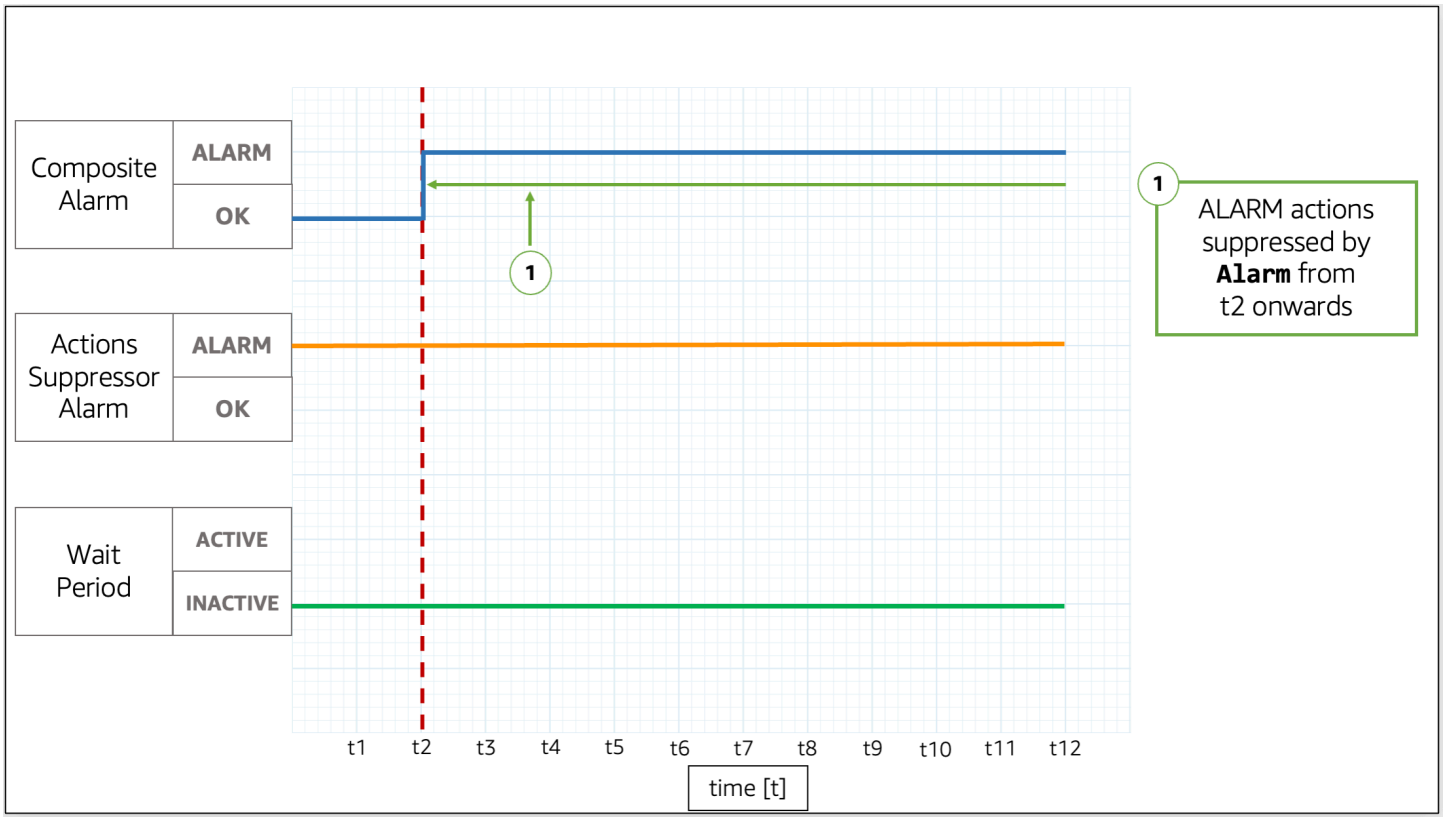
En la imagen, la alarma compuesta cambia los estados de `OK` a `ALARM` en t2. Un `WaitPeriod` comienza en t2 y termina en t4. Esto da tiempo a la alarma supresora para cambiar los estados de `OK` a `ALARM` en t3. Debido a que la alarma supresora cambia los estados de `OK` a `ALARM` en t3, el `WaitPeriod` que comenzó en t2 se descarta, y la alarma supresora impide entonces que la alarma compuesta realice acciones.

Ejemplo 3: transición de estado cuando el **`WaitPeriod`** suprime las acciones



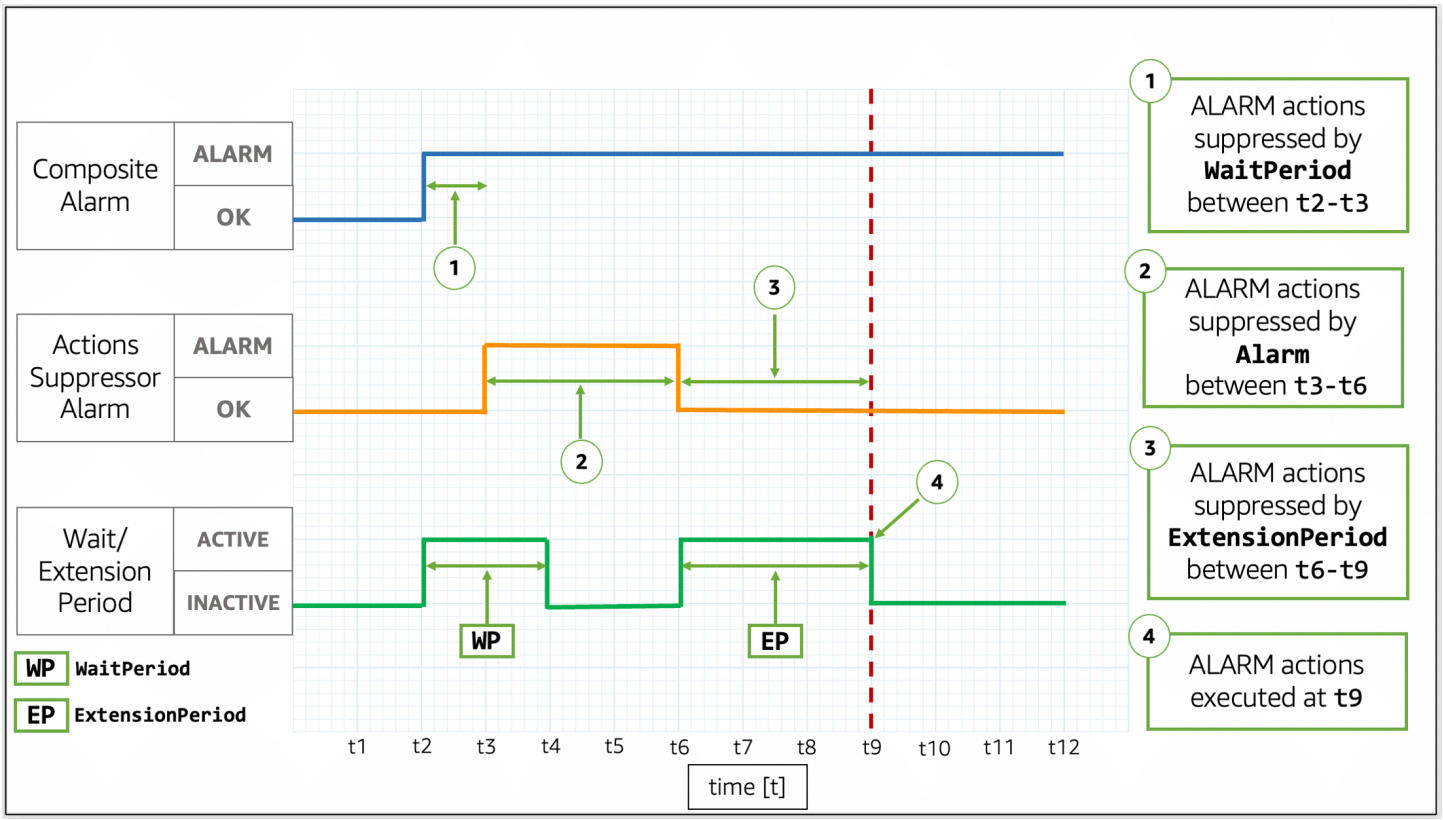
En la imagen, la alarma compuesta cambia los estados de `OK` a `ALARM` en t2. Un `WaitPeriod` comienza en t2 y termina en t4. Esto da tiempo a la alarma supresora para cambiar de estado. La alarma compuesta vuelve a cambiar a `OK` en t3, por lo que el `WaitPeriod` que comenzó en t2 se descarta. Un nuevo `WaitPeriod` comienza en t3 y termina en t5. Una vez que el nuevo `WaitPeriod` expire en t5, la alarma compuesta realizará sus acciones.

Ejemplo 4: transición de estado cuando las acciones se suprimen por alarma



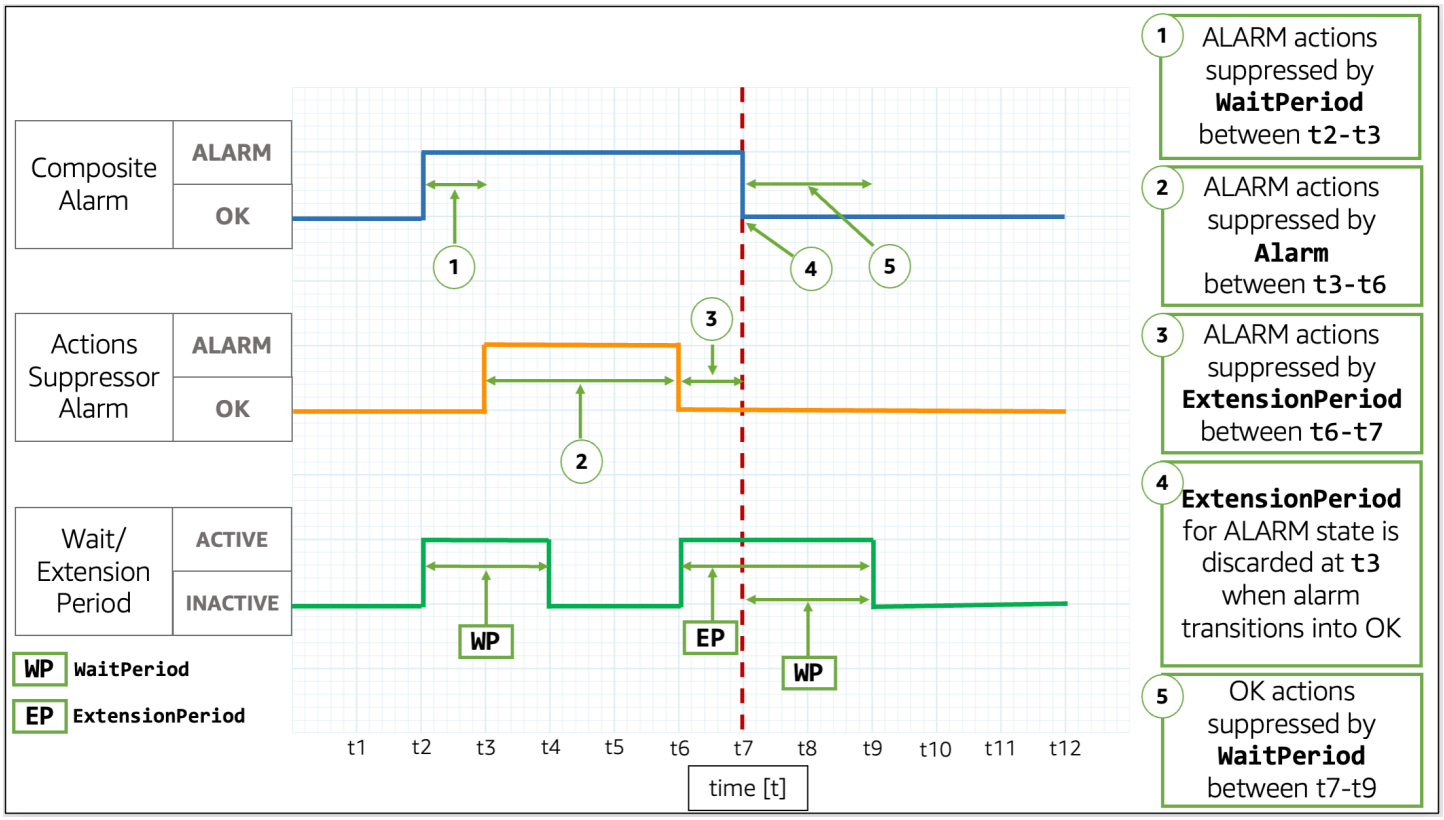
En la imagen, la alarma compuesta cambia los estados de `OK` a `ALARM` en t2. La alarma supresora ya está activada en `ALARM`. La alarma supresora detiene las acciones de la alarma compuesta.

Ejemplo 5: las acciones no se suprimen después del **`ExtensionPeriod`**



En la imagen, la alarma compuesta cambia los estados de OK a ALARM en t2. Un `WaitPeriod` comienza en t2 y termina en t4. Esto da tiempo a la alarma supresora para cambiar los estados de OK a ALARM en t3 antes de que suprima las acciones de la alarma compuesta hasta t6. Debido a que la alarma supresora cambia los estados de OK a ALARM en t3, el `WaitPeriod` que comenzó en t2 se descarta. En t6, la alarma supresora cambia a OK. Un `ExtensionPeriod` comienza en t6 y termina en t9. Después de que el `ExtensionPeriod` caduca, la alarma compuesta realizará acciones.

Ejemplo 6: transición de estado cuando las acciones son suprimidas por **`ExtensionPeriod`**



En la imagen, la alarma compuesta cambia los estados de OK a ALARM en t2. Un `WaitPeriod` comienza en t2 y termina en t4. Esto da tiempo a la alarma supresora para cambiar los estados de OK a ALARM en t3 antes de que suprima las acciones de la alarma compuesta hasta t6. Debido a que la alarma supresora cambia los estados de OK a ALARM en t3, el `WaitPeriod` que comenzó en t2 se descarta. En t6, la alarma supresora vuelve a cambiar a OK. Un `ExtensionPeriod` comienza en t6 y termina en t9. Cuando la alarma compuesta vuelva a cambiar a OK en t7, el `ExtensionPeriod` quedará descartado y un nuevo `WaitPeriod` comenzará en t7 y terminará en t9.

> **Tip**
>
> Si reemplaza la alarma supresora de acción, se descartará cualquier `WaitPeriod` o `ExtensionPeriod` activo.

Actuar ante los cambios de alarma

CloudWatch puede notificar a los usuarios sobre dos tipos de cambios de alarma: cuando una alarma cambia de estado y cuando se actualiza la configuración de una alarma.

Cuando se evalúa una alarma, puede cambiar de un estado a otro, como ALARM, OK o INSUFFICIENT_DATA. Estos cambios en el estado de la alarma pueden indicar un posible incidente, un retorno a la normalidad o la falta de disponibilidad de una métrica. En esos casos, es posible que desee involucrar o notificar a los usuarios utilizando cualquiera de las siguientes opciones:

- Puede configurar la alarma para que envíe una notificación a un tema de SNS como parte de las acciones de la alarma. Un tema SNS puede configurarse para mensajes de aplicación a aplicación (A2A), así como también para notificaciones de aplicación a persona (A2P), incluidos canales como notificaciones por correo electrónico y SMS. Todos los destinos que defina para el tema de SNS reciben la notificación de la alarma. Para obtener más información, consulte destinos de eventos de Amazon SNS.
- Puede configurar las notificaciones para los eventos de cambio de estado de alarma. AWS Las notificaciones de usuario ofrecen una forma nativa de configurar dichas notificaciones y es el método recomendado.

CloudWatch envía eventos a Amazon EventBridge cada vez que se crea, actualiza o elimina una alarma de CloudWatch o cambia su estado. Puede escribir reglas de EventBridge para tomar medidas o recibir notificaciones cuando EventBridge reciba estos eventos.

Temas

Notificar a los usuarios los cambios de alarma

En esta sección se explica cómo puede utilizar las notificaciones de AWS usuario o Amazon Simple Notification Service para que se notifique a los usuarios de los cambios de alarma.

Configuración de las notificaciones de usuario de AWS

Puede usar las notificaciones de usuario de AWS para configurar canales de entrega de y recibir notificaciones sobre cambios de estado de alarma de CloudWatch y eventos de cambio de configuración. Recibirá una notificación cuando un evento coincida con una regla que especifique. Puede recibir notificaciones de eventos a través de varios canales, como correo electrónico, notificaciones por AWS Chatbot o notificaciones push en la aplicación móvil de la consola de AWS. También puede ver las notificaciones en el Centro de notificaciones de la consola. Las notificaciones

de usuario admiten la agregación, lo que puede reducir el número de notificaciones que recibe durante eventos específicos.

Las configuraciones de notificaciones que cree con las notificaciones de usuario de AWS no se tienen en cuenta para el límite del número de acciones que puede configurar por estado de alarma objetivo. A medida que las notificaciones de usuario de AWS coinciden con los eventos emitidos a Amazon EventBridge, envía notificaciones para todas las alarmas de su cuenta y las regiones seleccionadas, a menos que especifique un filtro avanzado para permitir o denegar alarmas o patrones específicos.

El siguiente ejemplo de filtro avanzado coincide con un cambio de estado de alarma de OK a ALARM en la alarma llamada `ServerCpuTooHigh`.

```
{
"detail": {
    "alarmName": ["ServerCpuTooHigh"],
    "previousState": { "value": ["OK"] },
    "state": { "value": ["ALARM"] }
  }
}
```

Puede usar cualquiera de las propiedades publicadas por una alarma en los eventos de EventBridge para crear un filtro. Para obtener más información, consulte Eventos de alarma y EventBridge.

Configuración de notificaciones de Amazon SNS

Puede usar Amazon Simple Notification Service para enviar mensajería de aplicación a aplicación (A2A) y mensajería de aplicación a persona (A2P), incluidos mensajes de texto (SMS) y mensajes de correo electrónico. Para obtener más información, consulte destinos de eventos de Amazon SNS.

Para cada estado que pueda adoptar una alarma, puede configurarla para que envíe un mensaje a un tema de SNS. Cada tema de Amazon SNS que configure para un estado de una alarma determinada se tendrá en cuenta para el límite del número de acciones que puede configurar para esa alarma y estado. Puede enviar mensajes al mismo tema de Amazon SNS desde cualquier alarma de su cuenta y usar el mismo tema de Amazon SNS tanto para los consumidores de aplicaciones (A2A) como para los consumidores personales (A2P). Como esta configuración se realiza a nivel de alarma, solo las alarmas que haya configurado envían mensajes al tema de Amazon SNS seleccionado.

En primer lugar, cree un tema y, a continuación, suscríbase al mismo. Como opción, puede publicar un mensaje de prueba en el tema. Para ver un ejemplo, consulte Configuración de un tema de Amazon SNS mediante la AWS Management Console. Para obtener más información, consulte Introducción a Amazon SNS.

Como alternativa, si tiene previsto crear la alarma de CloudWatch con la AWS Management Console, puede omitir este procedimiento, ya que puede crear el tema al crear la alarma.

Al crear una alarma de CloudWatch, puede añadir acciones para cualquier estado de destino en el que se encuentre la alarma. Añada una notificación de Amazon SNS para el estado sobre el que desee recibir notificaciones y seleccione el tema de Amazon SNS que creó en el paso anterior para enviar una notificación por correo electrónico cuando la alarma entre en el estado seleccionado.

> **Note**
>
> Cuando cree un tema de Amazon SNS, puede elegir convertirlo en un tema estándar o un tema FIFO. CloudWatch garantiza la publicación de todas las notificaciones de alarma en ambos tipos de temas. Sin embargo, incluso si utiliza un tema FIFO, en pocos casos, CloudWatch envía las notificaciones al tema que no se está utilizando. Si utiliza un tema FIFO, la alarma establece que el ID del grupo de mensajes de las notificaciones de alarma sea un hash de los ARN de la alarma.

Prevención de errores del suplente confuso

Para evitar problemas de seguridad con suplentes confusos entre servicios, le recomendamos que utilice las claves de condición global `aws:SourceArn` y `aws:SourceAccount` de la política de recursos de Amazon SNS que conceden permiso a CloudWatch para acceder a los recursos de Amazon SNS.

La política de recursos del ejemplo siguiente ilustra la clave de condición de `aws:SourceArn` para reducir el permiso `SNS:Publish` para ser utilizado únicamente por las alarmas de CloudWatch de la cuenta especificada.

```
{
    "Statement": [{
        "Effect": "Allow",
        "Principal": {
            "Service": "cloudwatch.amazonaws.com"
```

```
        },
        "Action": "SNS:Publish",
        "Resource": "arn:aws:sns:us-east-2:444455556666:MyTopic",
        "Condition": {
            "ArnLike": {
                "aws:SourceArn": "arn:aws:cloudwatch:us-east-2:111122223333:alarm:*"
            },
            "StringEquals": {
                "aws:SourceAccount": "111122223333"
            }
        }
    }]
}
```

Si un ARN de alarma incluye caracteres que no sean ASCII, utilice únicamente la clave de condición global `aws:SourceAccount` para limitar los permisos.

Configuración de un tema de Amazon SNS mediante la AWS Management Console

En primer lugar, cree un tema y, a continuación, suscríbase al mismo. Como opción, puede publicar un mensaje de prueba en el tema.

Para crear un tema de SNS

1. Abra la consola de Amazon SNS en https://console.aws.amazon.com/sns/v3/home.
2. En el panel de Amazon SNS, en Common actions (Acciones comunes), elija Create Topic (Crear tema).
3. En el cuadro de diálogo Create new topic (Crear un nuevo tema), en Topic name (Nombre del tema), escriba un nombre para el tema (por ejemplo, **`my-topic`**).
4. Elija Create new topic (Crear nuevo tema).
5. Copie el Topic ARN (ARN de tema) en la siguiente tarea (por ejemplo, arn:aws:sns:us-east-1:111122223333:my-topic).

Para suscribirse a un tema de SNS

1. Abra la consola de Amazon SNS en https://console.aws.amazon.com/sns/v3/home.
2. En el panel de navegación, elija Subscriptions, Create subscription.
3. En el cuadro de diálogo Create subscription, en Topic ARN, pegue el ARN del tema que creó en la tarea anterior.

4. En Protocolo, elija Correo electrónico.
5. En Endpoint (Punto de enlace), escriba una dirección de correo electrónico que puede utilizar para recibir la notificación y, a continuación, elija Create subscription (Crear suscripción).
6. Desde su aplicación de correo electrónico, abra el mensaje de Notificaciones de AWS y confirme la suscripción.

   El navegador web muestra una respuesta de confirmación de Amazon SNS.

Para publicar un mensaje de prueba en un tema de SNS

1. Abra la consola de Amazon SNS en https://console.aws.amazon.com/sns/v3/home.
2. En el panel de navegación, elija Temas.
3. En la página Topics (Temas), seleccione un tema y elija Publish to topic (Publicar en tema).
4. En la página Publish a message (Publicar un mensaje), en Subject (Asunto), escriba una línea de asunto para el mensaje y en Message (Mensaje), escriba un mensaje breve.
5. Elija Publish Message (Publicar mensaje).
6. Compruebe el correo electrónico para confirmar que ha recibido el mensaje.

Configuración de un tema de SNS mediante la AWS CLI

Primero cree un tema de SNS y, a continuación, publique un mensaje directamente en el tema para comprobar que lo ha configurado correctamente.

Para configurar un tema de SNS

1. Cree el tema utilizando el comando create-topic como se indica a continuación.

   ```
   aws sns create-topic --name my-topic
   ```

   Amazon SNS muestra un ARN del tema con el siguiente formato:

   ```
   {
       "TopicArn": "arn:aws:sns:us-east-1:111122223333:my-topic"
   }
   ```

2. Suscríbase a su dirección de correo electrónico para utilizar el comando subscribe. Si la solicitud de suscripción tiene éxito, recibe un mensaje de correo electrónico de confirmación.

```
aws sns subscribe --topic-arn arn:aws:sns:us-east-1:111122223333:my-topic --
protocol email --notification-endpoint my-email-address
```

Amazon SNS devuelve lo siguiente:

```
{
    "SubscriptionArn": "pending confirmation"
}
```

3. Desde su aplicación de correo electrónico, abra el mensaje de Notificaciones de AWS y confirme la suscripción.

   En el navegador web se muestra una respuesta de confirmación de Amazon Simple Notification Service.

4. Compruebe la suscripción mediante el comando list-subscriptions-by-topic.

```
aws sns list-subscriptions-by-topic --topic-arn arn:aws:sns:us-
east-1:111122223333:my-topic
```

Amazon SNS devuelve lo siguiente:

```
{
  "Subscriptions": [
    {
        "Owner": "111122223333",
        "Endpoint": "me@mycompany.com",
        "Protocol": "email",
        "TopicArn": "arn:aws:sns:us-east-1:111122223333:my-topic",
        "SubscriptionArn": "arn:aws:sns:us-east-1:111122223333:my-topic:64886986-
bf10-48fb-a2f1-dab033aa67a3"
    }
  ]
}
```

5. (Opcional) Publique un mensaje de prueba en el tema mediante el comando publish.

```
aws sns publish --message "Verification" --topic arn:aws:sns:us-
east-1:111122223333:my-topic
```

Amazon SNS devuelve lo siguiente.

```
{
    "MessageId": "42f189a0-3094-5cf6-8fd7-c2dde61a4d7d"
}
```

6. Compruebe el correo electrónico para confirmar que ha recibido el mensaje.

Eventos de alarma y EventBridge

CloudWatch envía eventos a Amazon EventBridge cada vez que se crea, actualiza o elimina una alarma de CloudWatch o cambia su estado. Puede utilizar EventBridge y estos eventos para registrar reglas que realicen acciones, como enviar una notificación cuando una alarma cambie de estado. Para obtener más información, consulte What is Amazon EventBridge? (¿Qué es Amazon EventBridge?).

CloudWatch garantiza la entrega de eventos de cambio de estado de la alarma a EventBridge.

Eventos de ejemplo de CloudWatch

En esta sección se incluyen eventos de ejemplo de CloudWatch.

Cambio de estado de una alarma con una sola métrica

```
{
    "version": "0",
    "id": "c4c1c1c9-6542-e61b-6ef0-8c4d36933a92",
    "detail-type": "CloudWatch Alarm State Change",
    "source": "aws.cloudwatch",
    "account": "123456789012",
    "time": "2019-10-02T17:04:40Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:cloudwatch:us-east-1:123456789012:alarm:ServerCpuTooHigh"
    ],
    "detail": {
        "alarmName": "ServerCpuTooHigh",
        "configuration": {
            "description": "Goes into alarm when server CPU utilization is too high!",
            "metrics": [
```

```
                {
                    "id": "30b6c6b2-a864-43a2-4877-c09a1afc3b87",
                    "metricStat": {
                        "metric": {
                            "dimensions": {
                                "InstanceId": "i-12345678901234567"
                            },
                            "name": "CPUUtilization",
                            "namespace": "AWS/EC2"
                        },
                        "period": 300,
                        "stat": "Average"
                    },
                    "returnData": true
                }
            ]
        },
        "previousState": {
            "reason": "Threshold Crossed: 1 out of the last 1 datapoints
 [0.0666851903306472 (01/10/19 13:46:00)] was not greater than the threshold (50.0)
 (minimum 1 datapoint for ALARM -> OK transition).",
            "reasonData": "{\"version\":\"1.0\",\"queryDate\":
\"2019-10-01T13:56:40.985+0000\",\"startDate\":\"2019-10-01T13:46:00.000+0000\",
\"statistic\":\"Average\",\"period\":300,\"recentDatapoints\":[0.0666851903306472],
\"threshold\":50.0}",
            "timestamp": "2019-10-01T13:56:40.987+0000",
            "value": "OK"
        },
        "state": {
            "reason": "Threshold Crossed: 1 out of the last 1 datapoints
 [99.50160229693434 (02/10/19 16:59:00)] was greater than the threshold (50.0) (minimum
 1 datapoint for OK -> ALARM transition).",
            "reasonData": "{\"version\":\"1.0\",\"queryDate\":
\"2019-10-02T17:04:40.985+0000\",\"startDate\":\"2019-10-02T16:59:00.000+0000\",
\"statistic\":\"Average\",\"period\":300,\"recentDatapoints\":[99.50160229693434],
\"threshold\":50.0}",
            "timestamp": "2019-10-02T17:04:40.989+0000",
            "value": "ALARM"
        }
    }
}
```


Cambio de estado de una alarma con una métrica matemática

```
{
    "version": "0",
    "id": "2dde0eb1-528b-d2d5-9ca6-6d590caf2329",
    "detail-type": "CloudWatch Alarm State Change",
    "source": "aws.cloudwatch",
    "account": "123456789012",
    "time": "2019-10-02T17:20:48Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:cloudwatch:us-east-1:123456789012:alarm:TotalNetworkTrafficTooHigh"
    ],
    "detail": {
        "alarmName": "TotalNetworkTrafficTooHigh",
        "configuration": {
            "description": "Goes into alarm if total network traffic exceeds 10Kb",
            "metrics": [
                {
                    "expression": "SUM(METRICS())",
                    "id": "e1",
                    "label": "Total Network Traffic",
                    "returnData": true
                },
                {
                    "id": "m1",
                    "metricStat": {
                        "metric": {
                            "dimensions": {
                                "InstanceId": "i-12345678901234567"
                            },
                            "name": "NetworkIn",
                            "namespace": "AWS/EC2"
                        },
                        "period": 300,
                        "stat": "Maximum"
                    },
                    "returnData": false
                },
                {
                    "id": "m2",
                    "metricStat": {
                        "metric": {
                            "dimensions": {
                                "InstanceId": "i-12345678901234567"
```

```
                            },
                            "name": "NetworkOut",
                            "namespace": "AWS/EC2"
                        },
                        "period": 300,
                        "stat": "Maximum"
                    },
                    "returnData": false
                }
            ]
        },
        "previousState": {
            "reason": "Unchecked: Initial alarm creation",
            "timestamp": "2019-10-02T17:20:03.642+0000",
            "value": "INSUFFICIENT_DATA"
        },
        "state": {
            "reason": "Threshold Crossed: 1 out of the last 1 datapoints [45628.0
 (02/10/19 17:10:00)] was greater than the threshold (10000.0) (minimum 1 datapoint for
 OK -> ALARM transition).",
            "reasonData": "{\"version\":\"1.0\",\"queryDate\":
\"2019-10-02T17:20:48.551+0000\",\"startDate\":\"2019-10-02T17:10:00.000+0000\",
\"period\":300,\"recentDatapoints\":[45628.0],\"threshold\":10000.0}",
            "timestamp": "2019-10-02T17:20:48.554+0000",
            "value": "ALARM"
        }
    }
}
```

Cambio de estado de una alarma de detección de anomalías

```
{
    "version": "0",
    "id": "daafc9f1-bddd-c6c9-83af-74971fcfc4ef",
    "detail-type": "CloudWatch Alarm State Change",
    "source": "aws.cloudwatch",
    "account": "123456789012",
    "time": "2019-10-03T16:00:04Z",
    "region": "us-east-1",
    "resources": ["arn:aws:cloudwatch:us-east-1:123456789012:alarm:EC2 CPU Utilization
 Anomaly"],
    "detail": {
        "alarmName": "EC2 CPU Utilization Anomaly",
```

```
        "state": {
            "value": "ALARM",
            "reason": "Thresholds Crossed: 1 out of the last 1 datapoints [0.0
 (03/10/19 15:58:00)] was less than the lower thresholds [0.020599444741798756] or
 greater than the upper thresholds [0.3006915352732461] (minimum 1 datapoint for OK ->
 ALARM transition).",
            "reasonData": "{\"version\":\"1.0\",\"queryDate\":
\"2019-10-03T16:00:04.650+0000\",\"startDate\":\"2019-10-03T15:58:00.000+0000\",
\"period\":60,\"recentDatapoints\":[0.0],\"recentLowerThresholds\":
[0.020599444741798756],\"recentUpperThresholds\":[0.3006915352732461]}",
            "timestamp": "2019-10-03T16:00:04.653+0000"
        },
        "previousState": {
            "value": "OK",
            "reason": "Thresholds Crossed: 1 out of the last 1 datapoints
 [0.166666666664241 (03/10/19 15:57:00)] was not less than the lower thresholds
 [0.0206719426210418] or not greater than the upper thresholds [0.30076870222143803]
 (minimum 1 datapoint for ALARM -> OK transition).",
            "reasonData": "{\"version\":\"1.0\",\"queryDate\":
\"2019-10-03T15:59:04.670+0000\",\"startDate\":\"2019-10-03T15:57:00.000+0000\",
\"period\":60,\"recentDatapoints\":[0.166666666664241],\"recentLowerThresholds\":
[0.0206719426210418],\"recentUpperThresholds\":[0.30076870222143803]}",
            "timestamp": "2019-10-03T15:59:04.672+0000"
        },
        "configuration": {
            "description": "Goes into alarm if CPU Utilization is out of band",
            "metrics": [{
                "id": "m1",
                "metricStat": {
                    "metric": {
                        "namespace": "AWS/EC2",
                        "name": "CPUUtilization",
                        "dimensions": {
                            "InstanceId": "i-12345678901234567"
                        }
                    },
                    "period": 60,
                    "stat": "Average"
                },
                "returnData": true
            }, {
                "id": "ad1",
                "expression": "ANOMALY_DETECTION_BAND(m1, 0.8)",
                "label": "CPUUtilization (expected)",
```

```
                "returnData": true
            }]
        }
    }
}
```

Cambio de estado para una alarma compuesta con una alarma supresora

```
{
    "version": "0",
    "id": "d3dfc86d-384d-24c8-0345-9f7986db0b80",
    "detail-type": "CloudWatch Alarm State Change",
    "source": "aws.cloudwatch",
    "account": "123456789012",
    "time": "2022-07-22T15:57:45Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:cloudwatch:us-east-1:123456789012:alarm:ServiceAggregatedAlarm"
    ],
    "detail": {
        "alarmName": "ServiceAggregatedAlarm",
        "state": {
            "actionsSuppressedBy": "WaitPeriod",
            "actionsSuppressedReason": "Actions suppressed by WaitPeriod",
            "value": "ALARM",
            "reason": "arn:aws:cloudwatch:us-
east-1:123456789012:alarm:SuppressionDemo.EventBridge.FirstChild transitioned to ALARM
 at Friday 22 July, 2022 15:57:45 UTC",
            "reasonData": "{\"triggeringAlarms\":[{\"arn\":\"arn:aws:cloudwatch:us-
east-1:123456789012:alarm:ServerCpuTooHigh\",\"state\":{\"value\":\"ALARM\",\"timestamp
\":\"2022-07-22T15:57:45.394+0000\"}}]}",
            "timestamp": "2022-07-22T15:57:45.394+0000"
        },
        "previousState": {
            "value": "OK",
            "reason": "arn:aws:cloudwatch:us-
east-1:123456789012:alarm:SuppressionDemo.EventBridge.Main was created and its alarm
 rule evaluates to OK",
            "reasonData": "{\"triggeringAlarms\":[{\"arn\":\"arn:aws:cloudwatch:us-
east-1:123456789012:alarm:TotalNetworkTrafficTooHigh\",\"state\":{\"value\":\"OK\",
\"timestamp\":\"2022-07-14T16:28:57.770+0000\"}},{\"arn\":\"arn:aws:cloudwatch:us-
east-1:123456789012:alarm:ServerCpuTooHigh\",\"state\":{\"value\":\"OK\",\"timestamp\":
\"2022-07-14T16:28:54.191+0000\"}}]}",
```

```
            "timestamp": "2022-07-22T15:56:14.552+0000"
        },
        "configuration": {
            "alarmRule": "ALARM(ServerCpuTooHigh) OR
 ALARM(TotalNetworkTrafficTooHigh)",
            "actionsSuppressor": "ServiceMaintenanceAlarm",
            "actionsSuppressorWaitPeriod": 120,
            "actionsSuppressorExtensionPeriod": 180
        }
    }
}
```

Creación de una alarma compuesta

```
{

    "version": "0",
    "id": "91535fdd-1e9c-849d-624b-9a9f2b1d09d0",
    "detail-type": "CloudWatch Alarm Configuration Change",
    "source": "aws.cloudwatch",
    "account": "123456789012",
    "time": "2022-03-03T17:06:22Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:cloudwatch:us-east-1:123456789012:alarm:ServiceAggregatedAlarm"
    ],
    "detail": {
        "alarmName": "ServiceAggregatedAlarm",
        "operation": "create",
        "state": {
            "value": "INSUFFICIENT_DATA",
            "timestamp": "2022-03-03T17:06:22.289+0000"
        },
        "configuration": {
            "alarmRule": "ALARM(ServerCpuTooHigh) OR
 ALARM(TotalNetworkTrafficTooHigh)",
            "alarmName": "ServiceAggregatedAlarm",
            "description": "Aggregated monitor for instance",
            "actionsEnabled": true,
            "timestamp": "2022-03-03T17:06:22.289+0000",
            "okActions": [],
            "alarmActions": [],
            "insufficientDataActions": []
```

```
        }
    }
}
```

Creación de una alarma compuesta con una alarma supresora

```
{
    "version": "0",
    "id": "454773e1-09f7-945b-aa2c-590af1c3f8e0",
    "detail-type": "CloudWatch Alarm Configuration Change",
    "source": "aws.cloudwatch",
    "account": "123456789012",
    "time": "2022-07-14T13:59:46Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:cloudwatch:us-east-1:123456789012:alarm:ServiceAggregatedAlarm"
    ],
    "detail": {
        "alarmName": "ServiceAggregatedAlarm",
        "operation": "create",
        "state": {
            "value": "INSUFFICIENT_DATA",
            "timestamp": "2022-07-14T13:59:46.425+0000"
        },
        "configuration": {
            "alarmRule": "ALARM(ServerCpuTooHigh) OR
 ALARM(TotalNetworkTrafficTooHigh)",
            "actionsSuppressor": "ServiceMaintenanceAlarm",
            "actionsSuppressorWaitPeriod": 120,
            "actionsSuppressorExtensionPeriod": 180,
            "alarmName": "ServiceAggregatedAlarm",
            "actionsEnabled": true,
            "timestamp": "2022-07-14T13:59:46.425+0000",
            "okActions": [],
            "alarmActions": [],
            "insufficientDataActions": []
        }
    }
}
```

Actualización de una alarma de métrica

```
{
```

```
"version": "0",
"id": "bc7d3391-47f8-ae47-f457-1b4d06118d50",
"detail-type": "CloudWatch Alarm Configuration Change",
"source": "aws.cloudwatch",
"account": "123456789012",
"time": "2022-03-03T17:06:34Z",
"region": "us-east-1",
"resources": [
    "arn:aws:cloudwatch:us-east-1:123456789012:alarm:ServerCpuTooHigh"
],
"detail": {
    "alarmName": "ServerCpuTooHigh",
    "operation": "update",
    "state": {
        "value": "INSUFFICIENT_DATA",
        "timestamp": "2022-03-03T17:06:13.757+0000"
    },
    "configuration": {
        "evaluationPeriods": 1,
        "threshold": 80,
        "comparisonOperator": "GreaterThanThreshold",
        "treatMissingData": "ignore",
        "metrics": [
            {
                "id": "86bfa85f-b14c-ebf7-8916-7da014ce23c0",
                "metricStat": {
                    "metric": {
                        "namespace": "AWS/EC2",
                        "name": "CPUUtilization",
                        "dimensions": {
                            "InstanceId": "i-12345678901234567"
                        }
                    },
                    "period": 300,
                    "stat": "Average"
                },
                "returnData": true
            }
        ],
        "alarmName": "ServerCpuTooHigh",
        "description": "Goes into alarm when server CPU utilization is too high!",
        "actionsEnabled": true,
        "timestamp": "2022-03-03T17:06:34.267+0000",
```

```
            "okActions": [],
            "alarmActions": [],
            "insufficientDataActions": []
        },
        "previousConfiguration": {
            "evaluationPeriods": 1,
            "threshold": 70,
            "comparisonOperator": "GreaterThanThreshold",
            "treatMissingData": "ignore",
            "metrics": [
                {
                    "id": "d6bfa85f-893e-b052-a58b-4f9295c9111a",
                    "metricStat": {
                        "metric": {
                            "namespace": "AWS/EC2",
                            "name": "CPUUtilization",
                            "dimensions": {
                                "InstanceId": "i-12345678901234567"
                            }
                        },
                        "period": 300,
                        "stat": "Average"
                    },
                    "returnData": true
                }
            ],
            "alarmName": "ServerCpuTooHigh",
            "description": "Goes into alarm when server CPU utilization is too high!",
            "actionsEnabled": true,
            "timestamp": "2022-03-03T17:06:13.757+0000",
            "okActions": [],
            "alarmActions": [],
            "insufficientDataActions": []
        }
    }
}
```

Actualización de una alarma compuesta con una alarma supresora

```
{
    "version": "0",
    "id": "4c6f4177-6bd5-c0ca-9f05-b4151c54568b",
    "detail-type": "CloudWatch Alarm Configuration Change",
```

```
    "source": "aws.cloudwatch",
    "account": "123456789012",
    "time": "2022-07-14T13:59:56Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:cloudwatch:us-east-1:123456789012:alarm:ServiceAggregatedAlarm"
    ],
    "detail": {
        "alarmName": "ServiceAggregatedAlarm",
        "operation": "update",
        "state": {
            "actionsSuppressedBy": "WaitPeriod",
            "value": "ALARM",
            "timestamp": "2022-07-14T13:59:46.425+0000"
        },
        "configuration": {
            "alarmRule": "ALARM(ServerCpuTooHigh) OR
 ALARM(TotalNetworkTrafficTooHigh)",
            "actionsSuppressor": "ServiceMaintenanceAlarm",
            "actionsSuppressorWaitPeriod": 120,
            "actionsSuppressorExtensionPeriod": 360,
            "alarmName": "ServiceAggregatedAlarm",
            "actionsEnabled": true,
            "timestamp": "2022-07-14T13:59:56.290+0000",
            "okActions": [],
            "alarmActions": [],
            "insufficientDataActions": []
        },
        "previousConfiguration": {
            "alarmRule": "ALARM(ServerCpuTooHigh) OR
 ALARM(TotalNetworkTrafficTooHigh)",
            "actionsSuppressor": "ServiceMaintenanceAlarm",
            "actionsSuppressorWaitPeriod": 120,
            "actionsSuppressorExtensionPeriod": 180,
            "alarmName": "ServiceAggregatedAlarm",
            "actionsEnabled": true,
            "timestamp": "2022-07-14T13:59:46.425+0000",
            "okActions": [],
            "alarmActions": [],
            "insufficientDataActions": []
        }
    }
}
```

Eliminación de una alarma con métrica matemática


```
{

    "version": "0",
    "id": "f171d220-9e1c-c252-5042-2677347a83ed",
    "detail-type": "CloudWatch Alarm Configuration Change",
    "source": "aws.cloudwatch",
    "account": "123456789012",
    "time": "2022-03-03T17:07:13Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:cloudwatch:us-east-1:123456789012:alarm:TotalNetworkTrafficTooHigh"
    ],
    "detail": {
        "alarmName": "TotalNetworkTrafficTooHigh",
        "operation": "delete",
        "state": {
            "value": "INSUFFICIENT_DATA",
            "timestamp": "2022-03-03T17:06:17.672+0000"
        },
        "configuration": {
            "evaluationPeriods": 1,
            "threshold": 10000,
            "comparisonOperator": "GreaterThanThreshold",
            "treatMissingData": "ignore",
            "metrics": [{
                    "id": "m1",
                    "metricStat": {
                        "metric": {
                            "namespace": "AWS/EC2",
                            "name": "NetworkIn",
                            "dimensions": {
                                "InstanceId": "i-12345678901234567"
                            }
                        },
                        "period": 300,
                        "stat": "Maximum"
                    },
                    "returnData": false
                },
                {
                    "id": "m2",
                    "metricStat": {
```

```
                        "metric": {
                            "namespace": "AWS/EC2",
                            "name": "NetworkOut",
                            "dimensions": {
                                "InstanceId": "i-12345678901234567"
                            }
                        },
                        "period": 300,
                        "stat": "Maximum"
                    },
                    "returnData": false
                },
                {
                    "id": "e1",
                    "expression": "SUM(METRICS())",
                    "label": "Total Network Traffic",
                    "returnData": true
                }
            ],
            "alarmName": "TotalNetworkTrafficTooHigh",
            "description": "Goes into alarm if total network traffic exceeds 10Kb",
            "actionsEnabled": true,
            "timestamp": "2022-03-03T17:06:17.672+0000",
            "okActions": [],
            "alarmActions": [],
            "insufficientDataActions": []

        }
    }
}
```

Eliminación de una alarma compuesta con una alarma supresora

```
{
    "version": "0",
    "id": "e34592a1-46c0-b316-f614-1b17a87be9dc",
    "detail-type": "CloudWatch Alarm Configuration Change",
    "source": "aws.cloudwatch",
    "account": "123456789012",
    "time": "2022-07-14T14:00:01Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:cloudwatch:us-east-1:123456789012:alarm:ServiceAggregatedAlarm"
```

```
    ],
    "detail": {
        "alarmName": "ServiceAggregatedAlarm",
        "operation": "delete",
        "state": {
            "actionsSuppressedBy": "WaitPeriod",
            "value": "ALARM",
            "timestamp": "2022-07-14T13:59:46.425+0000"
        },
        "configuration": {
            "alarmRule": "ALARM(ServerCpuTooHigh) OR
 ALARM(TotalNetworkTrafficTooHigh)",
            "actionsSuppressor": "ServiceMaintenanceAlarm",
            "actionsSuppressorWaitPeriod": 120,
            "actionsSuppressorExtensionPeriod": 360,
            "alarmName": "ServiceAggregatedAlarm",
            "actionsEnabled": true,
            "timestamp": "2022-07-14T13:59:56.290+0000",
            "okActions": [],
            "alarmActions": [],
            "insufficientDataActions": []
        }
    }
}
```

Administración de alarmas

Edición o eliminación de una alarma de CloudWatch

Puede editar o eliminar una alarma existente.

No puede cambiar el nombre de una alarma existente. Puede copiar la alarma y asignar un nombre diferente a la nueva alarma. Para copiar una alarma, seleccione la casilla de verificación situada junto al nombre de la alarma en la lista de alarmas y elija Action (Acción), Copy (Copiar).

Para editar una alarma

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Alarms (Alarmas) y, a continuación, All Alarms (Todas las alarmas).
3. Elija el nombre de la alarma.

4. Para buscar etiquetas, elija la pestaña Etiquetas y, a continuación, elija Administrar etiquetas.
5. Para editar otras partes de la alarma, elija Acciones, Editar.

   Aparece la página Specify metric and conditions (Especificar métrica y condiciones), que muestra un gráfico y otra información sobre la métrica y la estadística que ha seleccionado.
6. Para cambiar la métrica, elija Edit (Editar), elija la pestaña All metrics (Todas las métricas) y realice una de las siguientes operaciones:
   - Elija el espacio de nombres del servicio que contiene la métrica que desea. Continúe eligiendo opciones conforme aparezcan para delimitar las opciones. Cuando aparezca una lista de métricas, active la casilla de verificación situada junto a la que desee utilizar.
   - En el campo de búsqueda, escriba el nombre de una métrica, dimensión o ID de recurso y pulse Intro. A continuación, elija uno de los resultados y continúe hasta que se muestre una lista de métricas. Seleccione la casilla de verificación situada junto a la métrica que desee.

   Elija Seleccionar métrica.
7. Para cambiar otros aspectos de la alarma, elija las opciones apropiadas. Para cambiar la cantidad de puntos de datos que deben estar fuera del umbral de la alarma para pasar al estado `ALARM` o para modificar la forma en que se tratan los datos que faltan, elija Additional configuration (Configuración adicional).
8. Elija Siguiente.
9. En Notification (Notificación), Auto Scaling action (Acción de Auto Scaling) y EC2 action (Acción de EC2), edite, si lo desea, las acciones que se realizarán cuando se active la alarma A continuación, elija Next.
10. También puede cambiar la descripción de la alarma.

    No puede cambiar el nombre de una alarma existente. Puede copiar la alarma y asignar un nombre diferente a la nueva alarma. Para copiar una alarma, seleccione la casilla de verificación situada junto al nombre de la alarma en la lista de alarmas y elija Action (Acción), Copy (Copiar).
11. Elija Siguiente.
12. En Obtener vista previa y crear, confirme que la información y las condiciones son las que desea y, a continuación, elija Actualizar alarma.

Para actualizar una lista de notificación por email que se creó con la consola de Amazon SNS

1. Abra la consola de Amazon SNS en https://console.aws.amazon.com/sns/v3/home.

2. En el panel de navegación, elija Topics (Temas) y, a continuación, seleccione el ARN de su lista de notificación (tema).
3. Realice una de las siguientes acciones siguientes:
   - Para añadir una dirección de correo electrónico, elija Create subscription. En Protocolo, elija Correo electrónico. En Endpoint (Punto de enlace), escriba la dirección de correo electrónico del nuevo destinatario. Seleccione Crear una suscripción.
   - Para eliminar una dirección de correo electrónico, elija el Subscription ID. Elija Other subscription actions (Otras acciones de la suscripción), Delete subscriptions (Eliminar suscripciones).
4. Elija Publish to topic (Publicar en el tema).

Eliminación de una alarma

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Alarmas.
3. Active la casilla de verificación situada a la izquierda del nombre de la alarma y elija Acciones, Eliminar.
4. Elija Eliminar.

Ocultación de alarmas de Auto Scaling

Cuando visualice las alarmas en la AWS Management Console, puede ocultar las alarmas relacionadas con Amazon EC2 Auto Scaling y Application Auto Scaling. Esta característica solo está disponible en la AWS Management Console.

Para ocultar temporalmente las alarmas de Auto Scaling

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Alarms (Alarmas), All alarms (Todas las alarmas) y Hide Auto Scaling alarms (Ocultar alarmas de Auto Scaling).

Casos de uso y ejemplos de alarmas

En las siguientes secciones, se proporcionan ejemplos y tutoriales de alarmas para casos de uso comunes.

Crear una alarma de facturación para supervisar los cargos estimados de AWS

Puede supervisar los cargos estimados de AWS mediante Amazon CloudWatch. Al habilitar la supervisión de los cargos estimados para la cuenta de AWS, los cargos estimados se calculan y se envían varias veces al día a CloudWatch como datos de métricas.

Los datos de métricas de facturación se almacenan en la Región EE. UU. Este (Norte de Virginia) y representan cargos en todo el mundo. Estos datos incluyen los cargos estimados para todos los servicios de AWS que utilice, además del conjunto total estimado de los cargos de AWS.

La alarma se activa cuando la facturación de su cuenta supera el umbral que ha especificado. Se activa solo cuando la facturación actual supera el umbral. No utiliza previsiones mensuales en función del uso hasta el momento.

Si crea una alarma de facturación en un momento en que los cargos ya han superado el umbral, la alarma pasa al estado `ALARM` inmediatamente.

> **Note**
>
> Para obtener información sobre cómo analizar los cargos de CloudWatch que ya se le han facturado, consulte Facturación y costo de CloudWatch.

Tareas

- Habilitación de alertas de facturación
- Crear una alarma de facturación
- Eliminación de una alarma de facturación

Habilitación de alertas de facturación

Antes de crear una alarma para los cargos estimados, debe habilitar las alertas de facturación, a fin de que pueda supervisar los cargos estimados de AWS y crear una alarma a través de los datos de las métricas de facturación. Después de habilitar las alertas de facturación, no puede deshabilitar la recopilación de datos, pero puede eliminar las alarmas de facturación que ha creado.

Después de habilitar las alertas de facturación por primera vez, se tardan unos 15 minutos antes de poder ver los datos de facturación y definir alertas de facturación.

Requisitos

- Debe haber iniciado sesión con las credenciales de superusuario de la cuenta o como usuario de IAM al que se le han concedido el permiso para ver la información de facturación.
- En el caso de las cuentas de facturación consolidada, los datos de facturación de cada cuenta vinculada pueden encontrarse iniciando sesión en la cuenta de pago. Puede consultar los datos de facturación de los cargos totales estimados y de los cargos estimados por servicio de cada cuenta vinculada, además de la cuenta consolidada.
- En una cuenta de facturación unificada, las métricas de la cuenta vinculada al miembro solo se capturan si la cuenta del pagador habilita la preferencia Recibir alertas de facturación. Si cambia qué cuenta es su cuenta de administración/pagador, debe activar las alertas de facturación en la nueva cuenta de administración/pagador.
- La cuenta no debe formar parte de la Red de socios de Amazon (APN) debido a que las métricas de facturación no se publican en CloudWatch para cuentas APN. Para obtener más información, consulte Red de socios de AWS.

Para habilitar la supervisión de los cargos estimados

1. Abra la Consola de AWS Billing en https://console.aws.amazon.com/billing/.
2. En el panel de navegación, elija Billing preferences (Preferencias de facturación).
3. En Preferencias de alertas, seleccione Editar.
4. Elija Recibir alertas de facturación de CloudWatch.
5. Elija Guardar preferencias.

Crear una alarma de facturación

> **Important**
>
> Antes de crear una alarma de facturación, debe configurar su región en Este de EE. UU. (Norte de Virginia). Los datos de métricas de facturación se almacenan en esta región y representan cargos en todo el mundo. También debe habilitar las alertas de facturación en su cuenta o en la cuenta de administración o pagador (si utiliza la facturación unificada). Para obtener más información, consulte Habilitación de alertas de facturación.

En este procedimiento, se crea una alarma que envía una notificación cuando los cargos estimados de AWS superan un umbral definido.

Para crear una alarma mediante la consola de CloudWatch

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Alarms (Alarmas) y, a continuación, elija All Alarms (Todas las alarmas).
3. Elija Create alarm (Crear alarma).
4. Elija Select Metric (Seleccionar métrica). En Browse (Navegar), elija Billing (Facturación) y, a continuación, elija Total Estimated Charge (Cargo total estimado).

   > **Note**
   >
   > Si no ve la métrica Facturación o Cargo total estimado, habilite las alertas de facturación y cambie su región a Este de EE. UU. (Norte de Virginia). Para obtener más información, consulte Habilitación de alertas de facturación.

5. Seleccione la casilla de la métrica EstimatedCharges y, a continuación, elija Select metric (Seleccionar métrica).
6. En Statistic (Estadística), elija Maximum (Máximo).
7. En Period (Período), seleccione 6 hours (6 horas).
8. En Threshold type (Tipo de umbral), elija Static (Estático).
9. En Whenever EstimatedCharges is . . . (Siempre que EstimatedCharges sea...), elija Greater (Mayor).
10. Para entonces..., defina el valor que desea que haga activar la alarma. Por ejemplo, **200** USD.

    Los valores métricos de EstimatedCharges están expresados únicamente en dólares estadounidenses (USD) y Amazon Services LLC se encarga de la conversión de divisas. Para obtener más información, consulte ¿Qué es AWS Billing?.

    > **Note**
    >
    > Después de definir un valor de umbral, el gráfico de vista previa muestra los cargos estimados para el mes actual.

11. Seleccione Configuración adicional y haga lo siguiente:

- En Datapoints to alarm (Puntos de datos para alarma), especifique 1 out of 1 (1 de 1).
- En Missing data treatment (Tratamiento de datos faltantes), elija Treat missing data as missing (Tratar los datos que faltan como faltantes).

12. Elija Siguiente.
13. En Notificación, asegúrese de seleccionar En alarma. A continuación, especifique el tema de Amazon SNS con el que se le notificará cuando su alarma se encuentre en el estado `ALARM`. El tema de Amazon SNS puede incluir su dirección de correo electrónico para que reciba un email cuando el importe de facturación supere el umbral que haya especificado.

    Puede seleccionar un tema de Amazon SNS existente, crear un tema de Amazon SNS nuevo o elegir un ARN de tema para notificar a otra cuenta. Si quiere que la alarma envíe varias notificaciones para el mismo estado de alarma o para estados de alarma diferentes, elija Add notification (Agregar notificación).
14. Elija Siguiente.
15. En Name and description (Nombre y descripción), ingrese un nombre para su alarma. El nombre debe contener solo caracteres UTF-8 y no puede contener caracteres de control ASCII.

    - (Opcional) Ingrese una descripción de la alarma. La descripción puede incluir el formato Markdown, que solo se muestra en la pestaña Detalles de la alarma de la consola de CloudWatch. Markdown puede resultar útil para añadir enlaces a manuales u otros recursos internos.
16. Elija Siguiente.
17. En Ver la vista previa y crear, asegúrese de que la configuración sea correcta y, a continuación, elija Crear alarma.

Eliminación de una alarma de facturación

Puede eliminar la alarma de facturación cuando deje de necesitarla.

Para eliminar una alarma de facturación

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. Si es necesario, cambie la región a Este de EE. UU. (Norte de Virginia). Los datos de métricas de facturación se almacenan en esta región y reflejan cargos en todo el mundo.
3. En el panel de navegación, elija Alarms (Alarmas) y, luego, Create Alarm (Crear alarma).

4. Seleccione la casilla de verificación junto a la alarma y elija Actions (Acciones), Delete (Eliminar).
5. Cuando se le indique que confirme, seleccione Yes, Delete (Sí, borrar).

Crear una alarma de uso de CPU

Puede crear una alarma de CloudWatch que envíe una notificación con Amazon SNS cuando la alarma cambie el estado de `OK` a `ALARM`.

La alarma cambia al estado `ALARM` cuando el uso promedio de la CPU de una instancia EC2 supera un umbral especificado para los períodos consecutivos especificados.

Configuración de una alarma de uso de CPU con la AWS Management Console

Utilice estos pasos para utilizar la AWS Management Console para crear una alarma de uso de CPU.

Para crear una alarma basada en el uso de la CPU

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Alarms (Alarmas) y, a continuación, All Alarms (Todas las alarmas).
3. Elija Create alarm (Crear alarma).
4. Elija Select Metric (Seleccionar métrica).
5. En la pestaña Todas las métricas, elija la opción de métricas de EC2.
6. Elija una categoría de métricas (por ejemplo, Métricas por instancia).
7. Busque la fila con la instancia que desea que aparezca en la columna InstanceId y CPUUtilization en la columna Metric Name (Nombre de métrica). Seleccione la casilla de verificación situada junto a esta fila y elija Seleccionar una métrica.
8. En Especifique la métrica y las condiciones, en Estadística elija Media y elija uno de los percentiles predefinidos o especifique un percentil personalizado (por ejemplo, **`p95.45`**).
9. Seleccione un periodo (por ejemplo, **`5 minutes`**).
10. En Conditions (Condiciones), especifique lo siguiente:

    a. En Threshold type (Tipo de umbral), elija Static (Estático).

    b. En la opción de cuando CPUUtilization es, especifique mayor que. En que..., especifique el umbral que provocará que la alarma vaya al estado ALARM si la utilización de la CPU supera este porcentaje. Por ejemplo: 70.

c. Elija Configuración adicional. Para Puntos de datos para alarma, especifique el número de periodos de evaluación (puntos de datos) que deben tener el estado `ALARM` para que se active la alarma. Si estos dos valores coinciden, creará una alarma que pasará al estado `ALARM` si se infringen muchos periodos consecutivos.

   Para crear una alarma M de N, especifique un número menor para el primer valor que el especificado para el segundo valor. Para obtener más información, consulte Evaluación de una alarma.

d. En Missing data treatment (Tratamiento de datos que faltan), elija cómo debe comportarse la alarma cuando falten algunos puntos de datos. Para obtener más información, consulte Configuración de la forma en la que las alarmas de CloudWatch tratan los datos que faltan.

e. Si la alarma utiliza un percentil como estadística supervisada, aparece un cuadro Percentiles with low samples (Percentiles con pocas muestras). Utilícelo para seleccionar si desea evaluar o no tener en cuenta los casos con frecuencias de muestreo bajas. Si elige ignore (maintain alarm state) (ignorar (mantener el estado de alarma)), el estado de alarma actual se mantiene siempre cuando el tamaño de la muestra es demasiado bajo. Para obtener más información, consulte Muestras de datos reducidos y alarmas de CloudWatch basadas en percentiles.

11. Elija Siguiente.
12. En Notification (Notificación), elija In alarm (Con alarma) y seleccione el tema de SNS que enviará las notificaciones cuando la alarma se encuentre en el estado `ALARM`.

    Para que la alarma envíe varias notificaciones para el mismo estado de alarma o para estados de alarma diferentes, seleccione Add notificación (Añadir notificación).

    Para que la alarma no envíe notificaciones, elija Remove (Eliminar).
13. Cuando haya terminado, elija Next (Siguiente).
14. Escriba un nombre y la descripción de la alarma. A continuación, elija Next.

    El nombre debe contener solo caracteres UTF-8 y no puede contener caracteres de control ASCII. La descripción puede incluir el formato Markdown, que solo se muestra en la pestaña Detalles de la alarma de la consola de CloudWatch. Markdown puede resultar útil para añadir enlaces a manuales u otros recursos internos.
15. En Obtener vista previa y crear, confirme que la información y las condiciones son las que desea y, a continuación, elija Crear alarma.

Configuración de una alarma de uso de CPU con la AWS CLI

Utilice estos pasos para utilizar la AWS CLI para crear una alarma de uso de CPU.

Para crear una alarma basada en el uso de la CPU

1. Configurar un tema de SNS. Para obtener más información, consulte Configuración de notificaciones de Amazon SNS.
2. Crear una alarma utilizando el comando put-metric-alarm como se indica a continuación.

```
aws cloudwatch put-metric-alarm --alarm-name cpu-mon --alarm-description "Alarm
 when CPU exceeds 70%" --metric-name CPUUtilization --namespace AWS/EC2 --statistic
 Average --period 300 --threshold 70 --comparison-operator GreaterThanThreshold --
dimensions  Name=InstanceId,Value=i-12345678 --evaluation-periods 2 --alarm-actions
 arn:aws:sns:us-east-1:111122223333:my-topic --unit Percent
```

3. Pruebe la alarma forzando un cambio de estado de alarma mediante el comando set-alarm-state.

   a. Cambie el estado de alarma de `INSUFFICIENT_DATA` a `OK`.

   ```
   aws cloudwatch set-alarm-state --alarm-name cpu-mon --state-reason
    "initializing" --state-value OK
   ```

   b. Cambie el estado de alarma de `OK` a `ALARM`.

   ```
   aws cloudwatch set-alarm-state --alarm-name cpu-mon --state-reason
    "initializing" --state-value ALARM
   ```

   c. Verifique que haya recibido una notificación por email acerca de la alarma.

Crear una alarma de latencia del equilibrador de carga que envíe un correo electrónico

Puede configurar una notificación de Amazon SNS y configurar una alarma que supervise la latencia que supere los 100 ms para el Classic Load Balancer.

Configuración de una alarma de latencia con la AWS Management Console

Utilice estos pasos para utilizar la AWS Management Console para crear una alarma de latencia del balanceador de carga.

Para crear una alarma de latencia del balanceador de carga

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Alarms (Alarmas) y, a continuación, All Alarms (Todas las alarmas).
3. Elija Create alarm (Crear alarma).
4. En CloudWatch Metrics by Category, elija la categoría ELB Metrics.
5. Seleccione la fila con el Classic Load Balancer y la métrica Latency (Latencia).
6. Para la estadística, elija Average (Promedio), elija uno de los percentiles predefinidos o especifique un percentil personalizado (por ejemplo, **`p95.45`**).
7. Para el periodo, elija 1 Minute (1 minuto).
8. Elija Siguiente.
9. En Alarm Threshold (Umbral de alarma), escriba un nombre único para la alarma (por ejemplo, **`myHighCpuAlarm`**) y una descripción de la alarma (por ejemplo, **`Alarm when Latency exceeds 100s`**). Los nombres de alarma deben contener solo caracteres UTF-8 y no pueden contener caracteres de control ASCII

   El nombre debe contener solo caracteres UTF-8 y no puede contener caracteres de control ASCII. La descripción puede incluir el formato Markdown, que solo se muestra en la pestaña Detalles de la alarma de la consola de CloudWatch. Markdown puede resultar útil para añadir enlaces a manuales u otros recursos internos.
10. En Whenever (Siempre que), en is (sea), elija **>** y escriba **`0.1`**. En for (para), escriba **`3`**.
11. En Additional settings (Configuración adicional), en Treat missing data as (Tratar datos que faltan como), elija ignore (maintain alarm state) [omitir (mantener estado de alarma)] para que los puntos de datos que faltan no activen cambios del estado de alarma.

    En Percentiles with low samples (Percentiles con pocas muestras), elija ignore (maintain the alarm state) [omitir (mantener el estado de alarma)] de modo que la alarma evalúe únicamente situaciones con un número suficiente de muestras de datos.
12. En Acciones, en Siempre que esta alarma, seleccione El estado es ALARMA. En Send notification to, elija un tema de SNS existente o cree uno nuevo.

Para crear un tema de SNS, elija New list (Nueva lista). En Send notification to (Enviar notificación a), escriba un nombre para el tema de SNS (por ejemplo, **myHighCpuAlarm**) y en Email list (Lista de correo electrónico), escriba una lista de las direcciones de correo electrónico separadas por comas que recibirán una notificación cuando la alarma cambie al estado ALARM. A cada dirección de correo electrónico se envía un correo electrónico de confirmación de suscripción del tema. Debe confirmar la suscripción antes de que se puedan enviar las notificaciones.

13. Elija Create Alarm (Crear alarma).

Configuración de una alarma de latencia con la AWS CLI

Utilice estos pasos para utilizar la AWS CLI para crear una alarma de latencia del balanceador de carga.

Para crear una alarma de latencia del balanceador de carga

1. Configurar un tema de SNS. Para obtener más información, consulte Configuración de notificaciones de Amazon SNS.
2. Crear la alarma utilizando el comando put-metric-alarm como se indica a continuación:

```
aws cloudwatch put-metric-alarm --alarm-name lb-mon --alarm-description "Alarm
 when Latency exceeds 100s" --metric-name Latency --namespace AWS/ELB --statistic
 Average --period 60 --threshold 100 --comparison-operator GreaterThanThreshold --
dimensions Name=LoadBalancerName,Value=my-server --evaluation-periods 3 --alarm-
actions arn:aws:sns:us-east-1:111122223333:my-topic --unit Seconds
```

3. Pruebe la alarma forzando un cambio de estado de alarma mediante el comando set-alarm-state.

   a. Cambie el estado de alarma de INSUFFICIENT_DATA a OK.

   ```
   aws cloudwatch set-alarm-state --alarm-name lb-mon --state-reason
    "initializing" --state-value OK
   ```

   b. Cambie el estado de alarma de OK a ALARM.

   ```
   aws cloudwatch set-alarm-state --alarm-name lb-mon --state-reason
    "initializing" --state-value ALARM
   ```

c. Compruebe que ha recibido una notificación por correo electrónico acerca de la alarma.

Crear una alarma de rendimiento de almacenamiento que envíe un correo electrónico

Puede configurar una notificación de SNS y una alarma que se active cuando Amazon EBS supere los 100 MB de rendimiento.

Configuración de una alarma de rendimiento de almacenamiento con la AWS Management Console

Siga estos pasos para utilizar la AWS Management Console a fin de crear una alarma basada en el rendimiento de Amazon EBS.

Para crear una alarma de rendimiento de almacenamiento

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Alarms (Alarmas) y, a continuación, All Alarms (Todas las alarmas).
3. Elija Create alarm (Crear alarma).
4. En EBS Metrics, elija una categoría de métricas.
5. Seleccione la fila con el volumen y la métrica VolumeWriteBytes.
6. Para la estadística, elija Average. Para el periodo, elija 5 Minutes. Elija Siguiente.
7. En Alarm Threshold (Umbral de alarma), escriba un nombre único para la alarma (por ejemplo, **`myHighWriteAlarm`**) y una descripción de la alarma (por ejemplo, **`VolumeWriteBytes exceeds 100,000 KiB/s`**). El nombre debe contener solo caracteres UTF-8 y no puede contener caracteres de control ASCII. La descripción puede incluir el formato Markdown, que solo se muestra en la pestaña Detalles de la alarma de la consola de CloudWatch. Markdown puede resultar útil para añadir enlaces a manuales u otros recursos internos.
8. En Whenever (Siempre que), en is (sea), elija **>** y escriba **`100000`**. En for (para), escriba **`15`** periodos consecutivos.

   En Vista previa de alarma se muestra una representación gráfica del umbral.
9. En Additional settings (Configuración adicional), en Treat missing data as (Tratar datos que faltan como), elija ignore (maintain alarm state) [omitir (mantener estado de alarma)] para que los puntos de datos que faltan no activen cambios del estado de alarma.

10. En Acciones, en Siempre que esta alarma, seleccione El estado es ALARMA. En Send notification to, elija un tema de SNS existente o cree uno.

    Para crear un tema de SNS, elija New list (Nueva lista). En Send notification to (Enviar notificación a), escriba un nombre para el tema de SNS (por ejemplo, **myHighCpuAlarm**) y en Email list (Lista de correo electrónico), escriba una lista de las direcciones de correo electrónico separadas por comas que recibirán una notificación cuando la alarma cambie al estado `ALARM`. A cada dirección de correo electrónico se envía un correo electrónico de confirmación de suscripción del tema. Debe confirmar la suscripción antes de que las notificaciones se puedan enviar a una dirección de correo electrónico.

11. Elija Create Alarm (Crear alarma).

Configuración de una alarma de rendimiento de almacenamiento con la AWS CLI

Siga estos pasos para utilizar la AWS CLI a fin de crear una alarma basada en el rendimiento de Amazon EBS.

Para crear una alarma de rendimiento de almacenamiento

1. Cree un tema de SNS. Para obtener más información, consulte Configuración de notificaciones de Amazon SNS.
2. Cree la alarma.

```
aws cloudwatch put-metric-alarm --alarm-name ebs-mon --alarm-description "Alarm
 when EBS volume exceeds 100MB throughput" --metric-name VolumeReadBytes --
namespace AWS/EBS --statistic Average --period 300 --threshold 100000000 --
comparison-operator GreaterThanThreshold --dimensions Name=VolumeId,Value=my-
volume-id --evaluation-periods 3 --alarm-actions arn:aws:sns:us-
east-1:111122223333:my-alarm-topic --insufficient-data-actions arn:aws:sns:us-
east-1:111122223333:my-insufficient-data-topic
```

3. Pruebe la alarma forzando un cambio de estado de alarma mediante el comando set-alarm-state.

    a. Cambie el estado de alarma de `INSUFFICIENT_DATA` a `OK`.

```
aws cloudwatch set-alarm-state --alarm-name ebs-mon --state-reason
 "initializing" --state-value OK
```

b. Cambie el estado de alarma de OK a ALARM.

```
aws cloudwatch set-alarm-state --alarm-name ebs-mon --state-reason
 "initializing" --state-value ALARM
```

c. Cambie el estado de alarma de ALARM a INSUFFICIENT_DATA.

```
aws cloudwatch set-alarm-state --alarm-name ebs-mon --state-reason
 "initializing" --state-value INSUFFICIENT_DATA
```

d. Compruebe que ha recibido una notificación por correo electrónico acerca de la alarma.

Crear una alarma en las métricas del contador de Performance Insights desde una base de datos AWS

CloudWatch incluye una función matemática métrica DB_PERF_INSIGHTS que puede utilizar para incorporar las contramétricas de Performance Insights a CloudWatch desde Amazon Relational Database Service y Amazon DocumentDB (con compatibilidad con MongoDB). DB_PERF_INSIGHTS también incluye la métrica DBLoad en intervalos de menos de un minuto. Puede establecer alarmas de CloudWatch sobre estas métricas.

Para obtener más información sobre Información de rendimiento de Amazon RDS, consulte Supervisión de la carga de la base de datos con Información de rendimiento de Amazon RDS.

Para obtener más información sobre Información de rendimiento de Amazon DocumentDB, consulte Supervisión con Performance Insights.

La detección de anomalías no es compatible con las alarmas basadas en la función DB_PERF_INSIGHTS.

> **Note**
>
> Las métricas de alta resolución con una granularidad inferior a un minuto recuperadas por DB_PERF_INSIGHTS solo se aplican a la métrica DBLoad, o a las métricas del sistema operativo si ha activado la supervisión mejorada con una resolución más alta. Para obtener más información sobre la supervisión mejorada de Amazon RDS, consulte Supervisión de las métricas del SO con Supervisión mejorada.
> Puede crear una alarma de alta resolución mediante la función DB_PERF_INSIGHTS. El rango máximo de evaluación de una alarma de alta resolución es de tres horas. Puede

utilizar la consola de CloudWatch para representar gráficamente las métricas recuperadas con la función DB_PERF_INSIGHTS para cualquier intervalo de tiempo.

Crear una alarma basada en métricas de Performance Insights

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Alarms (Alarmas) y, a continuación, elija All Alarms (Todas las alarmas).
3. Elija Create alarm (Crear alarma).
4. Elija Select Metric (Seleccionar métrica).
5. Seleccione el menú desplegable Añadir matemáticas y, a continuación, seleccione Database Performance Metrics (DB_PERF_INSIGHTS) en la lista.

   Tras seleccionar DB_PERF_INSIGHTS, aparecerá un cuadro de expresiones matemáticas en el que podrá aplicar o editar expresiones matemáticas.
6. En el cuadro de expresión matemática, introduzca la expresión matemática DB_PERF_INSIGHTS y, a continuación, seleccione Aplicar.

   Por ejemplo, **`DB_PERF_INSIGHTS('RDS', 'db-ABCDEFGHIJKLMNOPQRSTUVWXY1', 'os.cpuUtilization.user.avg')`**

   > **Important**
   >
   > Al utilizar la expresión matemática DB_PERF_INSIGHTS, debe especificar el identificador único de recurso de base de datos de la base de datos. Es diferente del identificador de la base de datos. Para encontrar el ID de recurso de base de datos en la consola de Amazon RDS, elija la instancia de base de datos para ver los detalles. A continuación, elija la pestaña Configuration (Configuración). El ID de recurso se muestra en la sección Configuración.

   Para obtener información sobre la función DB_PERF_INSIGHTS y otras funciones disponibles para las matemáticas métricas, consulte Sintaxis de matemáticas en las métricas y funciones.
7. Elija Seleccionar métrica.

Aparece la página Specify metric and conditions (Especificar métrica y condiciones), en la que se muestra un gráfico y otra información acerca de la expresión matemática que ha seleccionado.

8. En Whenever ***`expresión`*** is (Siempre que la expresión sea), especifique si la expresión debe ser mayor, menor o igual que el umbral. En than... (que...), especifique el valor de umbral.
9. Elija Configuración adicional. Para Puntos de datos para alarma, especifique el número de periodos de evaluación (puntos de datos) que deben tener el estado `ALARM` para que se active la alarma. Si estos dos valores coinciden, creará una alarma que pasará al estado `ALARM` si se infringen muchos periodos consecutivos.

   Para crear una alarma M de N, especifique un número menor para el primer valor que el especificado para el segundo valor. Para obtener más información, consulte Evaluación de una alarma.
10. En Missing data treatment (Tratamiento de datos que faltan), elija cómo debe comportarse la alarma cuando falten algunos puntos de datos. Para obtener más información, consulte Configuración de la forma en la que las alarmas de CloudWatch tratan los datos que faltan.
11. Elija Siguiente.
12. En Notification (Notificación), seleccione el tema de SNS al que desee enviar la notificación cuando la alarma tenga el estado `ALARM`, `OK` o `INSUFFICIENT_DATA`.

    Para que la alarma envíe varias notificaciones para el mismo estado de alarma o para estados de alarma diferentes, seleccione Add notificación (Añadir notificación).

    Para que la alarma no envíe notificaciones, elija Remove (Eliminar).
13. Para que la alarma realice acciones de escalado automático, EC2, Lambda o de Systems Manager, elija el botón correspondiente y seleccione el estado de la alarma y la acción que se debe realizar. Si elige una función de Lambda como acción de la alarma, debe especificar el nombre de la función o el ARN y, si lo desea, puede elegir una versión específica de la función.

    Las alarmas solo pueden realizar acciones de Systems Manager cuando entran en el estado ALARMA. Para obtener más información sobre las acciones de Systems Manager, consulte Configuración de CloudWatch para crear OpsItems a partir de alarmas y Creación de incidentes.

> **Note**
>
> Para crear una alarma que realice una acción de SSM Incident Manager, debe contar con determinados permisos. Para obtener más información, consulte Ejemplos de políticas basadas en identidades del Administrador de incidentes de AWS Systems Manager.

14. Cuando haya terminado, elija Next (Siguiente).
15. Escriba un nombre y la descripción de la alarma. A continuación, elija Next.

    El nombre debe contener solo caracteres UTF-8 y no puede contener caracteres de control ASCII. La descripción puede incluir el formato Markdown, que solo se muestra en la pestaña Detalles de la alarma de la consola de CloudWatch. Markdown puede resultar útil para añadir enlaces a manuales u otros recursos internos.
16. En Obtener vista previa y crear, confirme que la información y las condiciones son las que desea y, a continuación, elija Crear alarma.

Crear alarmas para detener, terminar, reiniciar o recuperar una instancia EC2

Mediante las acciones de la alarma de Amazon CloudWatch, puede crear alarmas que detienen, terminan, reinician o recuperan automáticamente las instancias EC2. Puede utilizar las acciones detener o terminar para ayudarle a ahorrar dinero cuando ya no necesita que se ejecute una instancia. Puede utilizar las acciones reiniciar y recuperar para reiniciar automáticamente dichas instancias o recuperarlas en nuevo hardware si se produce un deterioro del sistema.

Hay una serie de situaciones en las que es posible que desee detener o terminar la instancia automáticamente. Por ejemplo, es posible que tenga instancias dedicadas a trabajos de procesamiento de nóminas por lotes o tareas de cálculo científico que se ejecutan durante un período de tiempo y después completan su trabajo. En lugar de dejar dichas instancias inactivas (y acumulando cargos), puede pararlas o terminarlas, lo que le ayuda a ahorrar dinero. La principal diferencia entre utilizar las acciones de alarma detener y terminar es que puede reiniciar fácilmente una instancia detenida si necesita ejecutarla de nuevo más tarde. También puede mantener el mismo ID de instancia y volumen raíz. Sin embargo, no se puede reiniciar una instancia terminada. En su lugar, debe lanzar una nueva instancia.

Puede agregar las acciones detener, terminar, reiniciar en cualquier alarma establecida en una métrica por instancia de Amazon EC2, incluidas las métricas de monitoreo detallado y básico que Amazon CloudWatch proporciona (en el espacio de nombres AWS/EC2), además de cualquier métrica personalizada que incluya la dimensión "InstanceId =", siempre que el valor InstanceId se refiera a una instancia válida de Amazon EC2 en ejecución. También puede agregar la acción de recuperación a las alarmas configuradas en cualquier métrica por instancia de Amazon EC2, excepto en `StatusCheckFailed_Instance`.

Para configurar una acción en la alarma de CloudWatch que pueda reiniciar, detener o terminar una instancia, debe utilizar un rol de IAM vinculado al servicio, AWSServiceRoleForCloudWatchEvents. El rol de IAM AWSServiceRoleForCloudWatchEvents le permite a AWS realizar acciones en la alarma en su nombre.

Para crear un rol vinculado al servicio para CloudWatch Events, utilice el siguiente comando:

```
aws iam create-service-linked-role --aws-service-name events.amazonaws.com
```

Soporte de consola

Puede crear alarmas mediante la consola de Amazon EC2 o la consola de CloudWatch. Los procedimientos de esta documentación utilizan la consola de CloudWatch. Para los procedimientos que utilizan la consola de Amazon EC2, consulte Create Alarms That Stop, Terminate, Reboot, or Recover an Instance (Crear alarmas que detienen, terminan, reinician o recuperan una instancia) en la Guía del usuario para instancias de Linux.

Permisos

Si utiliza una cuenta de AWS Identity and Access Management (IAM) para crear o modificar una alarma que realice acciones de EC2 o acciones de OpsItem de Systems Manager, debe tener el permiso `iam:CreateServiceLinkedRole`.

Contenido

Agregar acciones de detención a las alarmas de Amazon CloudWatch

Puede crear una alarma que detenga una instancia Amazon EC2 cuando se alcance un umbral determinado. Por ejemplo, podría ejecutar instancias de desarrollo o de prueba y en ocasiones olvidarse de apagarlas. Puede crear una alarma que se active cuando el porcentaje de uso medio de la CPU haya estado por debajo del 10 por ciento durante 24 horas, hecho indicativo de que ha estado inactiva y que ya no se está usando. Puede ajustar el umbral, la duración y el periodo que se adapten mejor a sus necesidades, además puede añadir una notificación de SNS, de forma que recibirá un correo electrónico cuando se active la alarma.

Las instancias de Amazon EC2 que utilizan un volumen de Amazon Elastic Block Store como dispositivo raíz se pueden detener o terminar, mientras que aquellas que utilizan el almacén de instancias como dispositivo raíz solo se pueden terminar.

Para crear una alarma con el fin de detener una instancia inactiva mediante la consola de Amazon CloudWatch

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Alarms (Alarmas) y, luego, Create Alarm (Crear alarma).
3. Elija Create alarm (Crear alarma).
4. Elija Select Metric (Seleccionar métrica).
5. En los espacios de nombres de AWS, elija EC2.
6. Haga lo siguiente:
    a. Elija Per-Instance Metrics (Métricas por instancia).
    b. Seleccione la casilla de verificación en la fila con la instancia correcta y la métrica CPUUtilization.
    c. Elija la pestaña Métricas diagramadas.
    d. Para la estadística, elija Average.
    e. Seleccione un periodo (por ejemplo, **`1 Hour`**).
    f. Elija Seleccionar métrica.
7. En el paso Define Alarm, haga lo siguiente:
    a. En Conditions (Condiciones), elija Static (Estático).
    b. En Whenever the CPUUtilization is (Siempre que la CPUUtilization sea), elija Lower (Más baja).

c. Para than (que), escriba **`10`**.

d. Elija Siguiente.

e. En Notification, en Send notification to, elija un tema de SNS existente o cree uno nuevo.

   Para crear un tema de SNS, elija New list (Nueva lista). En Send a notification to (Enviar una notificación a), escriba un nombre para el tema de SNS (por ejemplo, Stop_EC2_Instance). En Email list (Lista de correo electrónico), escriba una lista separada por comas con las direcciones de correo electrónico a las que se van a enviar notificaciones cuando la alarma cambie al estado `ALARM`. A cada dirección de correo electrónico se envía un correo electrónico de confirmación de suscripción del tema. Debe confirmar la suscripción antes de que las notificaciones se puedan enviar a una dirección de correo electrónico.

f. Elija Add EC2 Action (Agregar acción de EC2).

g. Para Alarm state trigger (Desencadenador de estado de alarma), elija In Alarm (En alarma). Para Take the following action (Realizar la siguiente acción), elija Stop this instance (Detener esta instancia).

h. Elija Siguiente.

i. Escriba un nombre y la descripción de la alarma. El nombre solo debe contener caracteres ASCII. A continuación, elija Next.

j. En Preview and create (Obtener vista previa y crear), confirme que la información y las condiciones son las que desea y, a continuación, elija Create alarm (Crear alarma).

Agregar acciones de terminación a las alarmas de Amazon CloudWatch

Puede crear una alarma que termina una instancia de EC2 automáticamente cuando se alcanza un umbral determinado (siempre y cuando la protección de terminación no esté habilitada para la instancia). Por ejemplo, es posible que desee terminar una instancia cuando haya completado su trabajo y no necesita la instancia de nuevo. En caso de que desee utilizar la instancia en otro momento, debe detener la instancia en lugar de terminarla. Para obtener más información sobre la activación y desactivación de la protección de terminación de una instancia, consulte Enabling Termination Protection for an Instance (Activación de la protección de terminación para una instancia) en la Guía del usuario de Amazon EC2 para las instancias de Linux.

Para crear una alarma para terminar una instancia inactiva mediante la consola de Amazon CloudWatch

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Alarms, Create Alarm.
3. En el paso Select Metric, haga lo siguiente:
   a. En EC2 Metrics, elija Per-Instance Metrics.
   b. Seleccione la fila con la instancia y la métrica CPUUtilization.
   c. Para la estadística, elija Average.
   d. Seleccione un periodo (por ejemplo, **1 Hour**).
   e. Elija Siguiente.
4. En el paso Define Alarm, haga lo siguiente:
   a. En Alarm Threshold, escriba un nombre único para la alarma (por ejemplo, Terminar instancia EC2) y una descripción de la alarma (por ejemplo, Terminar la instancia EC2 cuando la CPU está inactiva demasiado tiempo). Los nombres de alarma solo pueden contener caracteres ASCII.
   b. En Whenever (Siempre que), en is (es), elija **<** y escriba **10**. En for (para), escriba **24** periodos consecutivos.

      En Vista previa de alarma se muestra una representación gráfica del umbral.
   c. En Notification, en Send notification to, elija un tema de SNS existente o cree uno nuevo.

      Para crear un tema de SNS, elija New list (Nueva lista). En Send a notification to (Enviar una notificación a), escriba un nombre para el tema de SNS (por ejemplo, Terminate_EC2_Instance) En Email list (Lista de correo electrónico), escriba una lista separada por comas con las direcciones de correo electrónico a las que se van a enviar notificaciones cuando la alarma cambie al estado `ALARM`. A cada dirección de correo electrónico se envía un correo electrónico de confirmación de suscripción del tema. Debe confirmar la suscripción antes de que las notificaciones se puedan enviar a una dirección de correo electrónico.
   d. Elija EC2 Action.
   e. En Whenever this alarm, elija State is ALARM. En Take this action, elija Terminate this instance.
   f. Seleccione Crear alarma.

Agregar acciones de reinicio a las alarmas de Amazon CloudWatch

Puede crear una alarma de Amazon CloudWatch que monitorice una instancia Amazon EC2 y reinicie la instancia automáticamente. La acción de alarma de reinicio se recomienda para errores de comprobación de estado de instancia (en contraposición a la acción de alarma recuperar, que es adecuada para los errores de comprobación de estado del sistema). Un reinicio de instancia es equivalente a un reinicio del sistema operativo. En la mayoría de los casos, solo necesita unos minutos para reiniciar su instancia. Cuando se reinicia una instancia, sigue estando en el mismo host físico, por lo que la instancia mantiene su nombre de DNS público, dirección IP privada y todos los datos en sus volúmenes de almacén de instancia.

El reinicio de una instancia no comienza una nueva hora de facturación de instancia, a diferencia de parar y reiniciar la instancia. Para obtener más información acerca de cómo se reinicia una instancia, consulte Reboot Your Instance (Reiniciar la instancia) en la Guía del usuario de Amazon EC2 para instancias de Linux.

> **Important**
>
> Para evitar una condición de carrera entre el reinicio y las acciones de recuperación, evitar configurar el mismo periodo de evaluación para una alarma de reinicio y otra de recuperación. Le recomendamos que configure las alarmas de reinicio en tres periodos de un minuto cada uno.

Para crear una alarma para reiniciar una instancia mediante la consola de Amazon CloudWatch

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Alarms, Create Alarm.
3. En el paso Select Metric, haga lo siguiente:
    a. En EC2 Metrics, elija Per-Instance Metrics.
    b. Seleccione la fila con la instancia y la métrica StatusCheckFailed_Instance.
    c. Para la estadística, elija Minimum.
    d. Seleccione un periodo (por ejemplo, **`1 Minute`**).
    e. Elija Siguiente.
4. En el paso Define Alarm, haga lo siguiente:

a. En Alarm Threshold, escriba un nombre único para la alarma (por ejemplo, Reiniciar instancia EC2) y una descripción de la alarma (por ejemplo, Reiniciar instancia EC2 cuando fallan las comprobaciones de estado). Los nombres de alarma solo pueden contener caracteres ASCII.

b. En Whenever (Siempre que), en is (es), elija **>** y escriba **0**. En for (para), escriba **3** periodos consecutivos.

   En Vista previa de alarma se muestra una representación gráfica del umbral.

c. En Notification, en Send notification to, elija un tema de SNS existente o cree uno nuevo.

   Para crear un tema de SNS, elija New list (Nueva lista). En Send a notification to (Enviar una notificación a), escriba un nombre para el tema de SNS (por ejemplo, Reboot_EC2_Instance). En Email list (Lista de correo electrónico), escriba una lista separada por comas con las direcciones de correo electrónico a las que se van a enviar notificaciones cuando la alarma cambie al estado `ALARM`. A cada dirección de correo electrónico se envía un correo electrónico de confirmación de suscripción del tema. Debe confirmar la suscripción antes de que las notificaciones se puedan enviar a una dirección de correo electrónico.

d. Elija EC2 Action.

e. En Whenever this alarm, elija State is ALARM. En Take this action, elija Reboot this instance.

f. Seleccione Crear alarma.

Agregar acciones de recuperación a las alarmas de Amazon CloudWatch

Puede crear una alarma de Amazon CloudWatch que supervise una instancia de Amazon EC2 y recupere de forma automática la instancia si deja de funcionar debido a un error de hardware subyacente o un problema que requiera la intervención de AWS en la reparación. Las instancias terminadas no se pueden recuperar. Una instancia recuperada es idéntica a la instancia original, incluido el ID de instancia, direcciones IP privadas, direcciones IP elásticas y todos los metadatos de la instancia.

Cuando se activa la alarma `StatusCheckFailed_System` y se inicia la acción de recuperación, se le notificará mediante el tema de Amazon SNS que ha eligido al crear la alarma y la acción de recuperación asociada. Durante la recuperación de la instancia, la instancia se migró durante un reinicio de instancia y los datos que hay en la memoria se pierden. Cuando el proceso se ha

completado, la información se publica en el tema de SNS que haya configurado para la alarma. Cualquier persona que esté suscrita a este tema de SNS recibirá una notificación por correo electrónico que incluye el estado del intento de recuperación e instrucciones adicionales. Observará un reinicio de instancia en la instancia recuperada.

La acción de recuperación solo se puede utilizar con `StatusCheckFailed_System`, no con `StatusCheckFailed_Instance`.

Entre los ejemplos de problemas que provocan errores en las comprobaciones de estado del sistema se incluyen:

- Pérdida de conectividad de red
- Pérdida de potencia del sistema
- Problemas de software en el host físico
- Problemas de hardware en el host físico que afectan a la accesibilidad a la red

La acción de recuperación solo se admite en algunos tipos de instancias. Para obtener más información sobre los tipos de instancias compatibles y otros requisitos, consulte Recover your instance (Recupere su instancia) y Requirements (Requisitos).

> **Important**
>
> Para evitar una condición de carrera entre el reinicio y las acciones de recuperación, evitar configurar el mismo periodo de evaluación para una alarma de reinicio y otra de recuperación. Le recomendamos que configure las alarmas de recuperación en dos periodos de evaluación de un minuto cada uno y las alarmas de reinicio en tres periodos de evaluación de un minuto cada uno.

Para crear una alarma con el fin de recuperar una instancia mediante la consola de Amazon CloudWatch

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Alarms, Create Alarm.
3. En el paso Select Metric, haga lo siguiente:

    a. En EC2 Metrics, elija Per-Instance Metrics.

b. Seleccione la fila con la instancia y la métrica StatusCheckFailed_System.

c. Para la estadística, elija Minimum.

d. Seleccione un periodo (por ejemplo, **1 Minute**).

> **Important**
>
> Para evitar una condición de carrera entre el reinicio y las acciones de recuperación, evitar configurar el mismo periodo de evaluación para una alarma de reinicio y otra de recuperación. Le recomendamos que configure las alarmas de recuperación en dos periodos de un minuto cada uno.

e. Elija Siguiente.

4. En el paso Define Alarm, haga lo siguiente:

a. En Alarm Threshold, escriba un nombre único para la alarma (por ejemplo, Recuperar instancia EC2) y una descripción de la alarma (por ejemplo, Recuperar instancia EC2 cuando fallan las comprobaciones de estado). Los nombres de alarma solo pueden contener caracteres ASCII.

b. En Whenever (Siempre que), en is (es), elija **>** y escriba **0**. En for (para), escriba **2** periodos consecutivos.

c. En Notification, en Send notification to, elija un tema de SNS existente o cree uno nuevo.

Para crear un tema de SNS, elija New list (Nueva lista). En Send a notification to (Enviar una notificación a), escriba un nombre para el tema de SNS (por ejemplo, Recover_EC2_Instance). En Email list (Lista de correo electrónico), escriba una lista separada por comas con las direcciones de correo electrónico a las que se van a enviar notificaciones cuando la alarma cambie al estado `ALARM`. A cada dirección de correo electrónico se envía un correo electrónico de confirmación de suscripción del tema. Debe confirmar la suscripción antes de que las notificaciones se puedan enviar a una dirección de correo electrónico.

d. Elija EC2 Action.

e. En Whenever this alarm, elija State is ALARM. En Take this action, elija Recover this instance.

f. Seleccione Crear alarma.

Ver el historial de alarmas activadas y acciones

Puede ver el historial de alarmas y de acciones en la consola de Amazon CloudWatch El historial de alarmas y acciones en Amazon CloudWatch se guarda por un período de 30 días.

Para ver el historial de las alarmas activadas y acciones

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Alarms (Alarmas) y seleccione una alarma.
3. Para ver la transición de estado más reciente junto con los valores de métricas y tiempo, elija Details.
4. Para ver las entradas de historial más recientes, elija History (Historial).

Alarmas y etiquetado

Las etiquetas son pares clave-valor que pueden ayudarle a organizar y categorizar sus recursos. También puede utilizarlas para lograr permisos de usuario, mediante la concesión de un permiso de usuario para acceder o cambiar únicamente recursos con determinados valores de etiqueta. Para obtener más información general sobre los recursos de etiquetado, consulte Tagging your AWS resources.

La lista siguiente explica algunos detalles sobre cómo funciona el etiquetado con las alarmas de CloudWatch.

- Para poder establecer o actualizar etiquetas para un recurso de CloudWatch, debe haber iniciado sesión en una cuenta que tenga el permiso `cloudwatch:TagResource`. Por ejemplo, para crear una alarma y establecer etiquetas para ella, debe tener el permiso `cloudwatch:TagResource` además del permiso `the cloudwatch:PutMetricAlarm` . Le recomendamos que se asegure de que todas las personas de su organización que vayan a crear o actualizar los recursos de CloudWatch tengan el permiso `cloudwatch:TagResource`.
- Una etiqueta puede usarse para un control de autorización basado en etiquetas. Por ejemplo, los permisos de usuario o rol de IAM pueden incluir condiciones para limitar las llamadas de CloudWatch a recursos específicos en función de sus etiquetas. Sin embargo, tenga en cuenta lo siguiente:
  - Las etiquetas con nombres que comiencen por `aws:` no pueden usarse para el control de autorizaciones basado en etiquetas.
  - Una alarma compuesta no admite control de autorización basado en etiquetas.

Application Signals

⚠ Application Signals está en la versión preliminar. Si tiene algún comentario sobre esta característica, puede ponerse en contacto con nosotros en app-signals-feedback@amazon.com.

Utilice CloudWatch Application Signals para instrumentar sus aplicaciones de manera automática en AWS de forma que pueda monitorear el estado actual de las aplicaciones y realizar un seguimiento del rendimiento de las aplicaciones a largo plazo en comparación con sus objetivos empresariales. Application Signals le proporciona una visión unificada y centrada en las aplicaciones de sus aplicaciones, servicios y dependencias y lo ayuda a monitorear y evaluar el estado de las aplicaciones.

- Active Application Signals para recopilar de manera automática métricas y seguimientos de sus aplicaciones y muestre métricas clave, como el volumen de llamadas, la disponibilidad, la latencia, las fallas y los errores. Vea y clasifique rápidamente el estado operativo actual y si sus aplicaciones cumplen sus objetivos de rendimiento a largo plazo, sin necesidad de escribir códigos personalizados ni crear paneles.
- Cree y monitoree los objetivos de nivel de servicio (SLO) con Application Signals. De manera sencilla, cree y realice un seguimiento del estado de los SLO relacionados con las métricas de CloudWatch, incluidas las nuevas métricas de aplicaciones estándar que recopila Application Signals. Consulte y realice un seguimiento del estado del indicador de nivel de servicio (SLI) de los servicios de las aplicaciones en una lista de servicios y un mapa topológico. Cree alarmas para realizar un seguimiento de sus SLO y de las nuevas métricas de aplicaciones estándar que recopila Application Signals.
- Consulte un mapa de la topología de su aplicación que Application Signals detecta automáticamente y que le ofrece una representación visual de las aplicaciones, las dependencias y la conectividad.
- Application Signals funciona con CloudWatch RUM, canarios de CloudWatch Synthetics, AWS Service Catalog AppRegistry y Amazon EC2 Auto Scaling para mostrar las páginas de sus clientes, los canarios de Synthetics y los nombres de las aplicaciones en paneles y mapas.

Uso de Application Signals para un monitoreo diario de las aplicaciones

Utilice Application Signals en la consola de CloudWatch, como parte de la supervisión diaria de las aplicaciones:

1. Si ha creado objetivos de nivel de servicio (SLO) para sus servicios, comience por la página de Objetivos de nivel de servicio (SLO). Esto le proporciona una visión inmediata del estado de sus servicios y operaciones más importantes. Elija el nombre del servicio o la operación de un SLO para abrir la página Detalles del servicio y ver la información detallada del servicio a medida que soluciona problemas.
2. Abra la página Servicios para obtener un resumen de todos sus servicios y ver rápidamente los servicios con la mayor tasa de errores o latencia. Si ha creado los SLO, consulte la tabla de servicios para ver qué servicios tienen indicadores de nivel de servicio (SLI) que no funcionan de forma correcta. Si un servicio concreto no funciona de forma correcta, selecciónelo para abrir la página de detalles del servicio y ver las operaciones del servicio, las dependencias, los valores controlados de Synthetics y las solicitudes de los clientes. Seleccione un punto de un gráfico para ver los seguimientos correlacionados, de forma que pueda solucionar e identificar la causa raíz de los problemas operativos.
3. Si se han implementado nuevos servicios o se han modificado las dependencias, abra el mapa de servicios para inspeccionar la topología de la aplicación. Vea un mapa de las aplicaciones que muestre la relación entre los clientes, los valores controlados de Synthetics, los servicios y las dependencias. Consulte rápidamente el estado del SLI, vea las métricas clave, como el volumen de llamadas, la tasa de errores y la latencia y profundice para obtener información más detallada en la página Detalles del servicio.

El uso de Application Signals también incurre en cargos. Para obtener más información sobre los precios de CloudWatch, consulte Precios de Amazon CloudWatch.

> **Note**
>
> No es necesario activar Application Signals para utilizar CloudWatch Synthetics, CloudWatch RUM o CloudWatch Evidently. Sin embargo, Synthetics y CloudWatch RUM funcionan con Application Signals para ofrecer beneficios cuando se utilizan estas características juntas.

Idiomas y arquitecturas compatibles

Actualmente, Application Signals admite aplicaciones Java y Pyhton.

Application Signals es compatible y está probado en Amazon EKS, Amazon ECS y Amazon EC2. En los clústeres de Amazon EKS, detecta automáticamente los nombres de los servicios y clústeres. En otras arquitecturas, debe proporcionar los nombres de los servicios y entornos al activar dichos servicios para Application Signals.

Las instrucciones para activar Application Signals en Amazon EC2 deberían funcionar en cualquier arquitectura que admita el agente CloudWatch y AWS Distro para OpenTelemetry. Sin embargo, las instrucciones no se han probado en otras arquitecturas que no sean Amazon ECS y Amazon EC2.

Regiones admitidas

Para esta versión preliminar, Application Signals es compatible con las siguientes regiones.

- Este de EE. UU. (Norte de Virginia)
- US East (Ohio)
- Oeste de EE. UU. (Oregón)
- Asia-Pacífico (Sídney)
- Asia-Pacífico (Tokio)
- Europa (Irlanda)

Vista previa del SDK

Existe una versión preliminar disponible del SDK para descargar.

> **Warning**
>
> Las operaciones y los parámetros de la API están sujetos a cambios antes de que Application Signals esté disponible de forma general. Estos cambios pueden ser cambios importantes. No utilice la versión de vista previa del SDK con fines de producción.

Para instalar la vista previa del SDK, primero instale o actualice la última versión de la versión 2 de la AWS CLI. Para obtener más información, consulte Instalar o actualizar la última versión de la AWS CLI.

A continuación, utilice los siguientes comandos para descargar el archivo zip del SDK del bucket de Amazon S3 y, luego, extraiga su contenido. Cada archivo zip del SDK contiene las instrucciones del SDK y la documentación de la API.

> **Note**
>
> El SDK se proporciona en varios lenguajes de programación para que pueda usar las API de Application Signals con cualquiera de estos lenguajes de programación. Sin embargo, la instrumentación automática de la aplicación para enviar datos a Application Signals solo se admite en aplicaciones Java y Python.

- SDK de Java V2: `aws s3 cp s3://application-signals-preview-sdk/awsJavaSdkV2.zip ./`
- SDK para JavaScript V3: `aws s3 cp s3://application-signals-preview-sdk/jsSdkV3.zip ./`
- SDK para JavaScript V2: `aws s3 cp s3://application-signals-preview-sdk/jsSdkV2.zip ./`
- SDK para Python: `aws s3 cp s3://application-signals-preview-sdk/pythonSdk.zip ./`
- SDK para Kotlin: `aws s3 cp s3://application-signals-preview-sdk/kotlin.zip ./`
- SDK para Android: `aws s3 cp s3://application-signals-preview-sdk/andriod.zip ./`
- SDK para C++: `aws s3 cp s3://application-signals-preview-sdk/awsCppSdk.zip ./`
- SDK para PHP: `aws s3 cp s3://application-signals-preview-sdk/awsSdkPhp.zip ./`
- SDK para Ruby: `aws s3 cp s3://application-signals-preview-sdk/awsSdkRuby.zip ./`
- SDK para Go V2: `aws s3 cp s3://application-signals-preview-sdk/awsSdkGoV2.zip ./`
- SDK para Go V1: `aws s3 cp s3://application-signals-preview-sdk/go.zip ./`
- SDK para iOS: `aws s3 cp s3://application-signals-preview-sdk/iOS.zip ./`

Temas

- Objetivos de nivel de servicio (SLO)
- Monitoreo del estado operativo de sus aplicaciones con Application Signals
- Recopilación de métricas de aplicaciones estándar
- Uso de la supervisión sintética
- Realice lanzamientos y experimentos A/B con CloudWatch Evidently
- Uso de CloudWatch RUM

Permisos necesarios para Application Signals

⚠ Application Signals se encuentra en versión preliminar para Amazon CloudWatch y está sujeto a cambios.

En esta sección se explican los permisos necesarios para habilitar, administrar y operar Application Signals.

Permisos para habilitar y administrar Application Signals

Para administrar Application Signals, debe iniciar sesión con los siguientes permisos:

```
{
 "Version": "2012-10-17",
 "Statement": [
    {
        "Sid": "CloudWatchApplicationSignalsFullAccessPermissions",
        "Effect": "Allow",
        "Action": "application-signals:*",
        "Resource": "*"
    },
    {
        "Sid": "CloudWatchApplicationSignalsAlarmsPermissions",
        "Effect": "Allow",
        "Action": [
            "cloudwatch:DescribeAlarms"
        ],
        "Resource": "*"
    },
    {
```

```
        "Sid": "CloudWatchApplicationSignalsMetricsPermissions",
        "Effect": "Allow",
        "Action": [
            "cloudwatch:GetMetricData",
            "cloudwatch:ListMetrics"
        ],
        "Resource": "*"
    },
    {
        "Sid": "CloudWatchApplicationSignalsLogGroupPermissions",
        "Effect": "Allow",
        "Action": [
            "logs:StartQuery",
            "logs:DescribeMetricFilters"
        ],
        "Resource": "arn:aws:logs:*:*:log-group:/aws/application-signals/data:*"
    },
    {
        "Sid": "CloudWatchApplicationSignalsLogsPermissions",
        "Effect": "Allow",
        "Action": [
            "logs:GetQueryResults",
            "logs:StopQuery"
        ],
        "Resource": "*"
    },
    {
        "Sid": "CloudWatchApplicationSignalsSyntheticsPermissions",
        "Effect": "Allow",
        "Action": [
            "synthetics:DescribeCanaries",
            "synthetics:DescribeCanariesLastRun",
            "synthetics:GetCanaryRuns"
        ],
        "Resource": "*"
    },
    {
        "Sid": "CloudWatchApplicationSignalsRumPermissions",
        "Effect": "Allow",
        "Action": [
            "rum:BatchCreateRumMetricDefinitions",
            "rum:BatchDeleteRumMetricDefinitions",
            "rum:BatchGetRumMetricDefinitions",
            "rum:GetAppMonitor",
```

```
            "rum:GetAppMonitorData",
            "rum:ListAppMonitors",
            "rum:PutRumMetricsDestination",
            "rum:UpdateRumMetricDefinition"
        ],
        "Resource": "*"
    },
    {
        "Sid": "CloudWatchApplicationSignalsXrayPermissions",
        "Effect": "Allow",
        "Action": [
            "xray:GetTraceSummaries"
        ],
        "Resource": "*"
    },
    {
        "Sid": "CloudWatchApplicationSignalsPutMetricAlarmPermissions",
        "Effect": "Allow",
        "Action": "cloudwatch:PutMetricAlarm",
        "Resource": [
            "arn:aws:cloudwatch:*:*:alarm:SLO-AttainmentGoalAlarm-*",
            "arn:aws:cloudwatch:*:*:alarm:SLO-WarningAlarm-*",
            "arn:aws:cloudwatch:*:*:alarm:SLI-HealthAlarm-*"
        ]
    },
    {
        "Sid": "CloudWatchApplicationSignalsCreateServiceLinkedRolePermissions",
        "Effect": "Allow",
        "Action": "iam:CreateServiceLinkedRole",
        "Resource": "arn:aws:iam::*:role/aws-service-role/application-
signals.cloudwatch.amazonaws.com/AWSServiceRoleForCloudWatchApplicationSignals",
        "Condition": {
            "StringLike": {
                "iam:AWSServiceName": "application-signals.cloudwatch.amazonaws.com"
            }
        }
    },
    {
        "Sid": "CloudWatchApplicationSignalsGetRolePermissions",
        "Effect": "Allow",
        "Action": "iam:GetRole",
        "Resource": "arn:aws:iam::*:role/aws-service-role/application-
signals.cloudwatch.amazonaws.com/AWSServiceRoleForCloudWatchApplicationSignals"
    },
```

```
    {
        "Sid": "CloudWatchApplicationSignalsSnsWritePermissions",
        "Effect": "Allow",
        "Action": [
            "sns:CreateTopic",
            "sns:Subscribe"
        ],
        "Resource": "arn:aws:sns:*:*:cloudwatch-application-signals-*"
    },
    {
        "Sid": "CloudWatchApplicationSignalsSnsReadPermissions",
        "Effect": "Allow",
        "Action": "sns:ListTopics",
        "Resource": "*"
    }
 ]
}
```

Para habilitar Application Signals en Amazon EC2, Kubernetes o arquitecturas personalizadas, consulte Habilitar Application Signals en otras plataformas con una configuración personalizada. Para habilitar y administrar Application Signals en Amazon EKS mediante el complemento de observabilidad de EKS de Amazon CloudWatch, necesita los siguientes permisos.

> **Important**
>
> Estos permisos incluyen `iam:PassRole` con `Resource "*"` y `eks:CreateAddon` con `Resource "*"`. Se trata de permisos potentes y debe tener cuidado al concederlos.

```
{
"Version": "2012-10-17",
    "Statement": [
        {
"Sid": "CloudWatchApplicationSignalsEksAddonManagementPermissions",
            "Effect": "Allow",
            "Action": [
                "eks:AccessKubernetesApi",
                "eks:CreateAddon",
                "eks:DescribeAddon",
                "eks:DescribeAddonConfiguration",
                "eks:DescribeAddonVersions",
                "eks:DescribeCluster",
```

```
                "eks:DescribeUpdate",
                "eks:ListAddons",
                "eks:ListClusters",
                "eks:ListUpdates",
                "iam:ListRoles",
                "iam:PassRole"
            ],
            "Resource": "*"
        },
        {
"Sid":
 "CloudWatchApplicationSignalsEksCloudWatchObservabilityAddonManagementPermissions",
            "Effect": "Allow",
            "Action": [
                "eks:DeleteAddon",
                "eks:UpdateAddon"
            ],
            "Resource": "arn:aws:eks:*:*:addon/*/amazon-cloudwatch-observability/*"
        }
    ]
}
```

El panel de Application Signals muestra las aplicaciones de AWS Service Catalog AppRegistry a las que están asociados sus SLO. Para ver estas aplicaciones en las páginas de SLO, debe contar con los siguientes permisos:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "CloudWatchApplicationSignalsTaggingReadPermissions",
            "Effect": "Allow",
            "Action": "tag:GetResources",
            "Resource": "*"
        }
    ]
}
```

Funcionamiento de Application Signals

Los operadores de servicio que utilizan Application Signals para supervisar los servicios y los SLO deben iniciar sesión en una cuenta con los siguientes permisos de solo lectura:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "CloudWatchApplicationSignalsReadOnlyAccessPermissions",
            "Effect": "Allow",
            "Action": [
                "application-signals:BatchGet*",
                "application-signals:Get*",
                "application-signals:List*"
            ],
            "Resource": "*"
        },
        {
            "Sid": "CloudWatchApplicationSignalsGetRolePermissions",
            "Effect": "Allow",
            "Action": "iam:GetRole",
            "Resource": "arn:aws:iam::*:role/aws-service-role/application-
signals.cloudwatch.amazonaws.com/AWSServiceRoleForCloudWatchApplicationSignals"
        },
        {
        "Sid": "CloudWatchApplicationSignalsLogGroupPermissions",
        "Effect": "Allow",
        "Action": [
            "logs:StartQuery",
            "logs:DescribeMetricFilters"
        ],
        "Resource": "arn:aws:logs:*:*:log-group:/aws/application-signals/data:*"
    },
    {
        "Sid": "CloudWatchApplicationSignalsLogsPermissions",
        "Effect": "Allow",
        "Action": [
            "logs:GetQueryResults",
            "logs:StopQuery"
        ],
        "Resource": "*"
    },
        {
            "Sid": "CloudWatchApplicationSignalsAlarmsReadPermissions",
            "Effect": "Allow",
            "Action": [
                "cloudwatch:DescribeAlarms"
```

```
            ],
            "Resource": "*"
        },
        {
            "Sid": "CloudWatchApplicationSignalsMetricsReadPermissions",
            "Effect": "Allow",
            "Action": [
                "cloudwatch:GetMetricData",
                "cloudwatch:ListMetrics"
            ],
            "Resource": "*"
        },
        {
            "Sid": "CloudWatchApplicationSignalsSyntheticsReadPermissions",
            "Effect": "Allow",
            "Action": [
                "synthetics:DescribeCanaries",
                "synthetics:DescribeCanariesLastRun",
                "synthetics:GetCanaryRuns"
            ],
            "Resource": "*"
        },
        {
            "Sid": "CloudWatchApplicationSignalsRumReadPermissions",
            "Effect": "Allow",
            "Action": [
                "rum:BatchGetRumMetricDefinitions",
                "rum:GetAppMonitor",
                "rum:GetAppMonitorData",
                "rum:ListAppMonitors"
            ],
            "Resource": "*"
        },
        {
            "Sid": "CloudWatchApplicationSignalsXrayReadPermissions",
            "Effect": "Allow",
            "Action": [
                "xray:GetTraceSummaries"
            ],
            "Resource": "*"
        }
    ]
}
```

Para ver qué aplicaciones de AWS Service Catalog AppRegistry están asociadas con sus SLO en el panel de Application Signals, necesita los siguientes permisos:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "CloudWatchApplicationSignalsTaggingReadPermissions",
            "Effect": "Allow",
            "Action": "tag:GetResources",
            "Resource": "*"
        }
    ]
}
```

Para comprobar si las Application Signals en Amazon EKS que utilizan el complemento de observabilidad de EKS de Amazon CloudWatch están habilitadas, debe tener los siguientes permisos:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "CloudWatchApplicationSignalsEksReadPermissions",
            "Effect": "Allow",
            "Action": [
                "eks:ListAddons",
                "eks:ListClusters"
            ],
            "Resource": "*"
        },
        {
            "Sid": "CloudWatchApplicationSignalsEksDescribeAddonReadPermissions",
            "Effect": "Allow",
            "Action": [
                "eks:DescribeAddon"
            ],
            "Resource": "arn:aws:eks:*:*:addon/*/amazon-cloudwatch-observability/*"
        }
    ]
}
```

Habilitar señales de aplicaciones

⚠ Application Signals está en versión de prueba. Si tiene algún comentario sobre esta característica, puede ponerse en contacto con nosotros en app-signals-feedback@amazon.com.

En los temas de esta sección se explica cómo habilitar las señales de aplicación de CloudWatch en el entorno. Application Signals es compatible con los clústeres de Amazon EKS con un flujo de trabajo de configuración mediante la consola. También es compatible con otras plataformas, incluida Amazon EC2, con un proceso de configuración personalizado.

Temas

Sistemas compatibles con Application Signals

⚠ Application Signals está en versión de prueba. Si tiene algún comentario sobre esta característica, puede ponerse en contacto con nosotros en app-signals-feedback@amazon.com.

Application Signals es compatible y está probado en Amazon EKS, Amazon ECS y Amazon EC2. Las instrucciones para habilitar Application Signals en Amazon EC2 deberían funcionar en cualquier plataforma que admita el agente CloudWatch y AWS Distro para OpenTelemetry, pero no se han probado en otras plataformas.

Compatibilidad con Java

Application Signals admite aplicaciones de Java, así como las mismas bibliotecas y marcos de Java que AWS Distro para OpenTelemetry. Para obtener más información, consulte Bibliotecas, marcos, servidores de aplicaciones y máquinas virtuales compatibles.

Se admiten las versiones 8, 11 y 17 de JVM.

Compatibilidad con Python

Application Signals admite las mismas bibliotecas y marcos que AWS Distro para OpenTelemetry. Para obtener más información, consulte Supported packages en opentelemetry-python-contrib.

Las versiones 3.8 y posteriores de Python son compatibles.

Antes de activar las señales de aplicación para sus aplicaciones Python, debe tener en cuenta las consideraciones siguientes.

- En algunas aplicaciones en contenedores, la falta de una variable de entorno `PYTHONPATH` a veces puede provocar que la aplicación no se inicie. Para solucionar este problema, asegúrese de configurar la variable de entorno `PYTHONPATH` en la ubicación del directorio de trabajo de la aplicación. Esto se debe a un problema conocido con la instrumentación automática de OpenTelemetry. Para obtener más información sobre este problema, consulte Python autoinstrumentation setting of PYTHONPATH is not compliant.
- Para las aplicaciones de Django, se requieren configuraciones adicionales, que se describen en la documentación de Python de OpenTelemetry.
  - Use el indicador `--noreload` para evitar la recarga automática.
  - Establezca la variable de entorno `DJANGO_SETTINGS_MODULE` en la ubicación del archivo `settings.py` de su aplicación Django. Esto garantiza que OpenTelemetry pueda acceder correctamente a la configuración de Django e integrarse correctamente con ella.

Consideraciones sobre la compatibilidad de OpenTelemetry

⚠ Application Signals está en versión de prueba. Si tiene algún comentario sobre esta característica, puede ponerse en contacto con nosotros en app-signals-feedback@amazon.com.

Para incorporar las aplicaciones a CloudWatch Application Signals, le recomendamos que elimine por completo cualquier solución de supervisión del rendimiento de las aplicaciones existente en la aplicación de antemano. Esto incluye eliminar cualquier configuración y código de instrumentación.

Si bien Application Signals utiliza instrumentación OpenTelemetry, no se garantiza que sea compatible con la instrumentación o configuración de OpenTelemetry existente. En el mejor de los casos, es posible que pueda conservar algunas de las funciones de OpenTelemetry, como las métricas personalizadas. Sin embargo, asegúrese de leer las secciones siguientes para obtener más detalles.

Consideraciones si ya usa OpenTelemetry

Si ya utiliza OpenTelemetry con la aplicación, el resto de esta sección contiene información importante para lograr la compatibilidad con Application Signals.

- Antes de habilitar la aplicación para Application Signals, debe eliminar la inyección de cualquier otro agente de instrumentación automática basado en OpenTelemetry de su aplicación. Esto ayuda a evitar conflictos de configuración. Puede seguir utilizando la instrumentación manual mediante las API de OpenTelemetry compatibles junto con Application Signals.
- Si utiliza instrumentación manual para generar intervalos o métricas personalizados a partir de la aplicación, según la complejidad de la instrumentación, habilitar Application Signals podría provocar que dejaran de generar datos o que se produjeran otros comportamientos no deseados. Es posible que pueda utilizar algunas de las configuraciones disponibles en OpenTelemetry (excepto las que se mencionan en la tabla que aparece más adelante en esta sección) para retener el comportamiento deseado de métricas o intervalos existentes. Para obtener más información sobre estas configuraciones, consulte SDK Configuration en la documentación de OpenTelemetry.

  Por ejemplo, si utiliza la configuración `OTEL_EXPORTER_OTLP_METRICS_ENDPOINT` y una instancia de OpenTelemetry Collector autoadministrada, es posible que pueda seguir enviando métricas personalizadas al destino que desee.
- Algunas variables de entorno o propiedades del sistema no deben usarse con Application Signals, mientras que puede usar otras siempre que siga las instrucciones de la tabla. Para obtener más información, consulte la tabla siguiente.

Variable de entorno	Recomendación con Application Signals
Variables de entorno generales	
`OTEL_SDK_DISABLED`	No debe establecerse en `true`.
`OTEL_TRACES_EXPORTER`	Debe establecerse en `otlp`.
`OTEL_EXPORTER_OTLP_ENDPOINT`	No debe usarse.
`OTEL_EXPORTER_OTLP_TRACES_ENDPOINT`	No debe usarse.
`OTEL_ATTRIBUTE_COUNT_LIMIT`	Si se establece, debe tener un valor lo suficientemente alto como para incluir aproximadamente 10 atributos de tramo más añadidos por CloudWatch Application Signals.
`OTEL_PROPAGATORS`	Si está establecido, debe incluirse `xray` para el rastreo final.
`OTEL_TRACES_SAMPLER`	Si está configurado, debe ser `xray` para utilizar la muestra de trazas centralizada de X-Ray. Para utilizar el muestreo local, configúrelo en `parentbased_traceidratio` y especifique la frecuencia de muestreo en `OTEL_TRACES_SAMPLER_ARG` .
`OTEL_TRACES_SAMPLER_ARG`	Si utiliza la muestra de trazas centralizada de X-Ray por defecto, no debe utilizar esta variable. Si, en su lugar, utiliza el muestreo local, establezca la frecuencia de muestreo en esta variable. Por ejemplo, `0.05` para una frecuencia de muestreo del 5 %.
Variables de entorno específicas de Java	

Variable de entorno	Recomendación con Application Signals
`OTEL_JAVA_ENABLED_RESOURCE_PROVIDERS`	Si está configurado, debe incluir detectores AWS de recursos.
Variables de entorno específicas de Python	
`OTEL_PYTHON_CONFIGURATOR`	Si se usa, debe configurarse en `aws_configurator`
`OTEL_PYTHON_DISTRO`	Si se usa, debe configurarse en `aws_distro`

Habilite Application Signals en los clústeres de Amazon EKS

⚠ Application Signals está en versión de prueba. Si tiene algún comentario sobre esta característica, puede ponerse en contacto con nosotros en app-signals-feedback@amazon.com.

CloudWatch Application Signals es compatible con las aplicaciones Java y Python que se ejecutan en clústeres de Amazon EKS. Para habilitar Application Signals para las aplicaciones de un clúster de Amazon EKS, tiene dos opciones:

- Para habilitar Application Signals para aplicaciones en un clúster de Amazon EKS existente, siga los pasos que se indican en Habilite Application Signals en un clúster de Amazon EKS con servicios.
- Para probar Application Signals en un entorno que no sea de producción con una aplicación de muestra, siga las instrucciones que se indican en Habilitar Application Signals en un nuevo clúster de Amazon EKS con una aplicación de muestra. Este flujo de trabajo utiliza scripts proporcionados por AWS para crear un nuevo clúster de Amazon EKS e instalar una aplicación de muestra habilitada para Application Signals. Esto le permite ver y probar la funcionalidad integral de Application Signals.

Temas

- Habilite Application Signals en un clúster de Amazon EKS con servicios

- Habilitar Application Signals en un nuevo clúster de Amazon EKS con una aplicación de muestra

Habilite Application Signals en un clúster de Amazon EKS con servicios

⚠ Application Signals está en versión de prueba. Si tiene algún comentario sobre esta característica, puede ponerse en contacto con nosotros en app-signals-feedback@amazon.com.

Para habilitar CloudWatch Application Signals en las aplicaciones de un clúster de Amazon EKS existente, siga las instrucciones de esta sección.

⚠ **Important**

Si ya utiliza OpenTelemetry con una aplicación que pretende habilitar para Application Signals, consulte Consideraciones sobre la compatibilidad de OpenTelemetry antes de activar Application Signals.

Para habilitar Application Signals para aplicaciones en un clúster de Amazon EKS existente

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Servicios.
3. Si aún no ha activado Application Signals en esta cuenta, debe conceder a Application Signals los permisos que necesita para descubrir los servicios. Para ello, haga lo siguiente. Solo es necesario hacerlo una vez para la cuenta.
   a. Elija Comenzar a descubrir sus servicios.
   b. Seleccione la casilla de verificación y elija Empezar a descubrir servicios.

      Al completar este paso por primera vez en la cuenta, se crea el rol vinculado al servicio AWSServiceRoleForCloudWatchApplicationSignals. Este rol otorga a Application Signals los siguientes permisos:

      - `xray:GetServiceGraph`
      - `logs:StartQuery`

- `logs:GetQueryResults`
- `cloudwatch:GetMetricData`
- `cloudwatch:ListMetrics`
- `tag:GetResources`

Para obtener más información acerca de este rol, consulte Permisos de roles vinculados a un servicio para CloudWatch Application Signals.

4. Seleccione Habilitar señales de aplicaciones.
5. En Especificar plataforma, elija EKS.
6. En Seleccionar un clúster EKS, seleccione el clúster en el que desee habilitar Application Signals.
7. Si este clúster aún no tiene habilitado el complemento Observabilidad de Amazon CloudWatch en EKS, se le solicitará que lo habilite. En este caso, realice lo siguiente:

   a. Seleccione Añadir el complemento de observabilidad CloudWatch de EKS. Aparece la consola de Amazon EKS.

   b. Seleccione la casilla Observabilidad de Amazon CloudWatch y elija Siguiente.

   El complemento CloudWatch Observability de EKS brinda tanto a Application Signals como a Información de contenedores de CloudWatch capacidad de observabilidad mejorada para Amazon EKS. Para obtener más información sobre Información de contenedores, consulte Información de contenedores.

   c. Seleccione la versión más reciente del complemento para instalar.

   d. Seleccione un rol de IAM para usarlo en el complemento. Si elige Heredar del nodo, asocie los permisos correctos al rol de IAM utilizado por los nodos de trabajo. Sustituya *my-worker-node-role* por el rol de IAM que utilizan sus nodos de trabajo de Kubernetes.

   ```
   aws iam attach-role-policy \
   --role-name my-worker-node-role \
   --policy-arn arn:aws:iam::aws:policy/CloudWatchAgentServerPolicy \
   --policy-arn arn:aws:iam::aws:policy/AWSXRayWriteOnlyAccess
   ```

   e. Si desea crear un rol de servicio para utilizar el complemento, consulteInstalar el agente de CloudWatch mediante el complemento de EKS de observabilidad de Amazon CloudWatch.

   f. Seleccione Siguiente, confirme la información de la pantalla y seleccione Crear.

g. En la siguiente pantalla, seleccione Habilitar CloudWatch Application Signals para volver a la consola de CloudWatch y finalizar el proceso.

8. Hay dos opciones para habilitar sus aplicaciones para Application Signals. Para mantener la coherencia, recomendamos elegir una opción por clúster.

   - La opción de consola es más sencilla. El uso de este método hace que los pods se reinicien inmediatamente.
   - El método Annotate Manifest File le brinda más control sobre cuándo se reinician sus pods y también puede ayudarlo a administrar su monitoreo de una manera más descentralizada si no desea centralizarlo.

   Console

   La opción de consola usa la configuración avanzada del complemento de EKS de observabilidad de Amazon CloudWatch para configurar Application Signals para sus servicios. Para obtener más información acerca del complemento, consulte Configuraciones adicionales (Opcional).

   Si no ve una lista de cargas de trabajo y espacios de nombres, asegúrese de tener los permisos correctos para verlos en este clúster. Para obtener más información, consulte Permisos necesarios.

   Puede supervisar cargas de trabajo individuales o espacios de nombres completos.

   Para supervisar una sola carga de trabajo:

   1. Seleccione la casilla de verificación junto a la carga de trabajo que desea monitorear.
   2. Seleccione el idioma de la carga de trabajo. En el caso de las aplicaciones Python, asegúrese de que la aplicación cumpla los requisitos previos necesarios antes de continuar. Para obtener más información, consulte La aplicación Python no se inicia después de activar Application Signals.
   3. Seleccione Listo. El complemento de EKS de observabilidad de Amazon CloudWatch inyectará inmediatamente los SDK de AWS Distro para OpenTelemetry autoinstrumentation (ADOT) en sus pods y activará reinicios de pods para permitir la recopilación de métricas y seguimientos de aplicaciones.

   Para supervisar un espacio de nombres completo:

1. Seleccione la casilla de verificación junto al espacio de nombres que desea monitorear.
2. Seleccione el idioma de la carga de trabajo. Esto se aplica a todas las cargas de trabajo de este espacio de nombres, tanto si están implementadas actualmente como si se implementarán en el futuro. En el caso de las aplicaciones Python, asegúrese de que la aplicación cumpla los requisitos previos necesarios antes de continuar. Para obtener más información, consulte La aplicación Python no se inicia después de activar Application Signals.
3. Seleccione Listo. El complemento de EKS de observabilidad de Amazon CloudWatch inyectará inmediatamente los SDK de AWS Distro para OpenTelemetry autoinstrumentation (ADOT) en sus pods y activará reinicios de pods para permitir la recopilación de métricas y seguimientos de aplicaciones.

Para habilitar Application Signals en otro clúster de Amazon EKS, seleccione Habilitar Application Signals en la pantalla Servicios.

Annotate manifest file

En la consola de CloudWatch, la sección Servicios del monitor explica que debe agregar una anotación a un manifiesto YAML en el clúster. La adición de esta anotación instrumenta automáticamente la aplicación para enviar métricas, seguimientos y registros a Application Signals.

Dispone de dos opciones para realizar la anotación:

- Anotar carga de trabajo instrumenta automáticamente una sola carga de trabajo en el clúster.
- Anotar el espacio de nombres instrumenta automáticamente todas las cargas de trabajo desplegadas en el espacio de nombres seleccionado.

Elija una de esas opciones y siga los pasos correspondientes:

- Para anotar una sola carga de trabajo:
  1. Elija Anotar carga de trabajo.
  2. Pegue una de las siguientes líneas en la sección `PodTemplate` del archivo de manifiesto de carga de trabajo.

- Para cargas de trabajo Java: `annotations: instrumentation.opentelemetry.io/inject-java: "true"`
- Para cargas de trabajo Python: `annotations: instrumentation.opentelemetry.io/inject-python: "true"`

  Para las aplicaciones Python, se requieren configuraciones adicionales. Para obtener más información, consulte La aplicación Python no se inicia después de activar Application Signals.

3. En la terminal, ingrese `kubectl apply -f` *`your_deployment_yaml`* para aplicar el cambio.

- Para anotar todas las cargas de trabajo en un espacio de nombres:

1. Seleccione Anotar el espacio de nombres.
2. Pegue una de las siguientes líneas en la sección de metadatos del archivo de manifiesto del espacio de nombres. Si el espacio de nombres incluye cargas de trabajo Java y Python, pegue estas dos líneas en el archivo de manifiesto del espacio de nombres.
   - Si hay cargas de trabajo Java en el espacio de nombres: `annotations: instrumentation.opentelemetry.io/inject-java: "true"`
   - Si hay cargas de trabajo Python en el espacio de nombres: `annotations: instrumentation.opentelemetry.io/inject-python: "true"`

     Para las aplicaciones Python, se requieren configuraciones adicionales. Para obtener más información, consulte La aplicación Python no se inicia después de activar Application Signals.
3. En la terminal, ingrese `kubectl apply -f` *`your_namespace_yaml`* para aplicar el cambio.
4. En la terminal, introduzca un comando para reiniciar todos los pods del espacio de nombres. Un ejemplo de comando para reiniciar las cargas de trabajo de implementación es `kubectl rollout restart deployment -n` *`namespace_name`*

9. Elija Ver servicios cuando haya terminado. Esto lo llevará a la vista de los servicios de Application Signals, donde podrá ver los datos que Application Signals recopila. Es posible que se tarde unos minutos en mostrar los datos.

   Para habilitar Application Signals en otro clúster de Amazon EKS, seleccione Habilitar Application Signals en la pantalla Servicios.

Para obtener más información sobre la vista de los Servicios, consulte Monitoreo del estado operativo de sus aplicaciones con Application Signals.

> **Note**
>
> Hemos identificado algunas consideraciones que debe tener en cuenta al habilitar las aplicaciones Python para Application Signals. Para obtener más información, consulte La aplicación Python no se inicia después de activar Application Signals.

Habilitar Application Signals en un nuevo clúster de Amazon EKS con una aplicación de muestra

> **Application Signals está en versión de prueba. Si tiene algún comentario sobre esta característica, puede ponerse en contacto con nosotros en app-signals-feedback@amazon.com.**

Para probar CloudWatch Application Signals en una aplicación de muestra antes de instrumentar sus propias aplicaciones con ella, siga las instrucciones de esta sección. Estas instrucciones utilizan scripts para ayudarle a crear un clúster de Amazon EKS, instalar una aplicación de ejemplo e instrumentar la aplicación de ejemplo para que funcione con Application Signals.

La aplicación de ejemplo es una aplicación de Spring llamada “Pet Clinic” que se compone de cuatro microservicios. Estos servicios se ejecutan en Amazon EKS en Amazon EC2 y utilizan los scripts de activación de Application Signals para habilitar el clúster con el agente de autoinstrumentación de Java o Python.

Requisitos

- Actualmente, Application Signals supervisa solo las aplicaciones Java y Python.
- Debe tener instalado la AWS CLI en la instancia. Recomendamos la AWS CLI versión 2, pero la versión 1 también debería funcionar. Para obtener información sobre cómo instalar AWS CLI, consulte Instalar o actualizar la versión más reciente de la AWS CLI.
- Los scripts de esta sección están diseñados para ejecutarse en entornos Linux y macOS. En el caso de las instancias de Windows, se recomienda utilizar un entorno AWS Cloud9 para ejecutar

estos scripts. Para obtener más información acerca de AWS Cloud9, consulte ¿Qué es AWS Cloud9?

- Instale una versión compatible de `kubectl`. Debe utilizar una versión de `kubectl` con una diferencia de versión menor de un número que el plano de control del clúster de Amazon EKS. Por ejemplo, un cliente de `kubectl` 1.26 debe funcionar con los clústeres 1.25, 1.26 y 1.27 de Kubernetes. Si ya tiene un clúster de Amazon EKS, es posible que tenga que configurar las credenciales de AWS para `kubectl`. Para obtener más información, consulte Creación o actualización de un archivo `kubeconfig` para un clúster de Amazon EKS.
- Instale `eksctl`. `eksctl` usa la AWS CLI para interactuar con AWS, lo que significa que usa las mismas credenciales de AWS que la AWS CLI. Para obtener más información, consulte Instalación o actualización de `eksctl`.
- Instalar `jq`. `jq` es necesario para ejecutar los scripts de activación de Application Signals. Para obtener más información, consulte Descargar jq.

Paso 1: descargar los scripts

Para descargar los scripts para configurar CloudWatch Application Signals con una aplicación de muestra, puede descargar y descomprimir el archivo de proyecto de GitHub comprimido en una unidad local, o puede clonar el proyecto de GitHub.

Para clonar el proyecto, abra una ventana de terminal e ingrese el siguiente comando de Git en un directorio de trabajo determinado:

```
git clone https://github.com/aws-observability/application-signals-demo.git
```

Paso 2: compilar y ejecutar la aplicación de ejemplo

Para crear e insertar las imágenes de la aplicación de muestra, siga estas instrucciones.

Paso 3: implementar y habilitar Application Signals y la aplicación de muestra

Asegúrese de haber completado los requisitos enumerados en Habilitar Application Signals en un nuevo clúster de Amazon EKS con una aplicación de muestra antes de completar los siguientes pasos.

Para implementar y habilitar Application Signals y la aplicación de muestra

1. Ingrese el siguiente comando en la terminal local donde descomprimió el script de incorporación. Sustituya *nuevo-nombre-clúster* por el nombre que desee para el nuevo clúster. Sustituya *nombre-región* por el nombre de la región AWS, como `us-west-1`.

   Este comando configura la aplicación de muestra que se ejecuta en un nuevo clúster de Amazon EKS con Application Signals activado.

   ```
   # assuming the current working directory is 'onboarding'
   # this script sets up a new cluster, enables Application Signals, and deploys the
   # sample application
   cd application-signals-demo/scripts/eks/appsignals/one-step && ./setup.sh new-
   cluster-name region-name
   ```

   El script de configuración tarda unos 30 minutos en ejecutarse y hace lo siguiente:

   - Crea un nuevo clúster de Amazon EKS en la región especificada.
   - Crea los permisos de IAM necesarios para Application Signals (`arn:aws:iam::aws:policy/AWSXrayWriteOnlyAccess` y `arn:aws:iam::aws:policy/CloudWatchAgentServerPolicy`).
   - Habilita Application Signals instalando el agente de CloudWatch e instrumentando automáticamente la aplicación de muestra para las métricas de CloudWatch y los seguimientos de X-Ray.
   - Implementa la aplicación de ejemplo PetClinic Spring en el mismo clúster de Amazon EKS.
   - Crea cinco valores controlados de CloudWatch Synthetics, denominados `pc-add-vist`, `pc-create-owners`, `pc-visit-pet`, `pc-visit-vet`, `pc-clinic-traffic`. Estos valores controlados se ejecutarán a una frecuencia de un minuto para generar tráfico sintético para la aplicación de muestra y demostrar cómo aparecen los valores controlados Synthetics en Application Signals.
   - Crea cuatro objetivos de nivel de servicio (SLO) para la aplicación PetClinic con los siguientes nombres:
     - Disponibilidad para buscar un propietario
     - Latencia para la búsqueda de un propietario
     - Disponibilidad para registrar un propietario
     - Latencia para registrar un propietario

- Crea el rol de IAM necesario con una política de confianza personalizada que otorga a Application Signals los siguientes permisos:
  - `cloudwatch:PutMetricData`
  - `cloudwatch:GetMetricData`
  - `xray:GetServiceGraph`
  - `logs:StartQuery`
  - `logs:GetQueryResults`

2. (Opcional) Si desea revisar el código de origen de la aplicación de muestra de PetClinic, puede encontrarlo en la carpeta raíz.

```
- application-signals-demo
  - spring-petclinic-admin-server
  - spring-petclinic-api-gateway
  - spring-petclinic-config-server
  - spring-petclinic-customers-service
  - spring-petclinic-discovery-server
  - spring-petclinic-vets-service
  - spring-petclinic-visits-service
```

3. Para ver la aplicación de muestra de PetClinic implementada, ejecute el siguiente comando para buscar la URL:

```
kubectl get ingress
```

Paso 4: supervisar la aplicación de muestra

Tras completar los pasos de la sección anterior para crear el clúster de Amazon EKS e implementar la aplicación de ejemplo, puede usar Application Signals para monitorizar la aplicación.

> **Note**
>
> Para que la consola de Application Signals comience a llenarse, parte del tráfico debe llegar a la aplicación de muestra. En parte de los pasos anteriores, se crearon valores controlados de CloudWatch Synthetics que generan tráfico a la aplicación de muestra.

Supervisión del estado de los servicios

Una vez habilitada, CloudWatch Application Signals descubre y completa automáticamente una lista de servicios sin necesidad de ninguna configuración adicional.

Para ver la lista de servicios descubiertos y supervisar su estado

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación de servicio, elija Application Signals, Servicios.
3. Para ver un servicio, sus operaciones y sus dependencias, elija el nombre de uno de los servicios de la lista.

   Esta vista unificada y centrada en las aplicaciones ayuda a proporcionar una perspectiva completa de la forma en que los usuarios interactúan con el servicio. Esto puede ayudarle a clasificar los problemas en caso de que se produzcan anomalías en el rendimiento. Para obtener detalles completos sobre la vista de servicios, consulte Monitoreo del estado operativo de sus aplicaciones con Application Signals.
4. Seleccione la pestaña Operaciones de servicio para ver las métricas de aplicación estándar para las operaciones de ese servicio. Las operaciones son las operaciones de API a las que llama el servicio, por ejemplo.

   A continuación, para ver los gráficos de una sola operación de ese servicio, elija el nombre de esa operación.
5. Seleccione la pestaña Dependencias para ver las dependencias que tiene la aplicación, junto con las métricas de aplicación críticas de cada dependencia. Las dependencias incluyen AWS los servicios y los servicios de terceros a los que recurre la aplicación.
6. Para ver los seguimientos correlacionados desde la página de detalles del servicio, elija un punto de datos en uno de los tres gráficos que se encuentran arriba de la tabla. Esto rellena un nuevo panel con los seguimientos filtrados del período de tiempo. Estos seguimientos se ordenan y filtran en función del gráfico que haya elegido. Por ejemplo, si elige el gráfico de latencia, los seguimientos se ordenan por tiempo de respuesta del servicio.
7. En el panel de navegación de la consola de CloudWatch, elija SLOs. Verá los SLO que el script creó para la aplicación de ejemplo. Para obtener más información acerca de los SLO, consulte Objetivos de nivel de servicio (SLO).

(Opcional) Paso 5: limpieza

Cuando termine de probar las señales de la aplicación, puede usar un script proporcionado por Amazon para limpiar y eliminar los artefactos creados en la cuenta para la aplicación de muestra. Para realizar la limpieza, ingrese el siguiente comando. Sustituya *nuevo-nombre-clúster* por el nombre del clúster que creó para la aplicación de ejemplo y reemplace nombre-*región* por el nombre de la región AWS, por ejemplo `us-west-1`.

```
cd application-signals-demo/scripts/eks/appsignals/one-step && ./cleanup.sh new-
cluster-name region-name
```

Habilite Application Signals en otras plataformas con una configuración personalizada

> ⚠ Application Signals está en versión de prueba. Si tiene algún comentario sobre esta característica, puede ponerse en contacto con nosotros en app-signals-feedback@amazon.com.

Habilite CloudWatch Application Signals en plataformas distintas de Amazon EKS mediante los pasos de configuración personalizados de estas secciones. En estas arquitecturas, usted mismo instala y configura el agente de CloudWatch y la distribución para AWS OpenTelemetry.

En estas arquitecturas, Application Signals no descubre automáticamente los nombres de los servicios ni sus clústeres o hosts. Debe especificar estos nombres durante la configuración personalizada, y los nombres que especifique son los que aparecen en los paneles de Application Signals.

Temas

- Use una configuración personalizada para habilitar Application Signals en Amazon ECS
- Use una configuración personalizada para habilitar Application Signals en Amazon EC2 y otras plataformas

Use una configuración personalizada para habilitar Application Signals en Amazon ECS

⚠ Application Signals está en versión de prueba. Si tiene algún comentario sobre esta característica, puede ponerse en contacto con nosotros en app-signals-feedback@amazon.com.

Utilice estas instrucciones de configuración personalizadas para incorporar aplicaciones de Amazon ECS a CloudWatch Application Signals. Usted mismo instala y configura el agente y la distribución de CloudWatch para AWS OpenTelemetry.

En los clústeres de Amazon ECS, Application Signals no descubre automáticamente los nombres de los servicios ni los clústeres en los que se ejecutan. Debe especificar estos nombres durante la configuración personalizada, y los nombres que especifique son los que aparecen en los paneles de Application Signals.

⚠ Important

Solo se admite el modo de red `awsvpc`.

Paso 1: habilitar Application Signals en la cuenta

Si aún no ha activado Application Signals en esta cuenta, debe conceder a Application Signals los permisos que necesita para descubrir los servicios. Para ello, haga lo siguiente. Solo es necesario hacerlo una vez para la cuenta.

Para habilitar Application Signals para las aplicaciones

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Servicios.
3. Elija Comenzar a descubrir sus servicios.
4. Seleccione la casilla de verificación y elija Empezar a descubrir servicios.

   Al completar este paso por primera vez en la cuenta, se crea el rol vinculado al servicio AWSServiceRoleForCloudWatchApplicationSignals. Este rol otorga a Application Signals los siguientes permisos:

- `xray:GetServiceGraph`
- `logs:StartQuery`
- `logs:GetQueryResults`
- `cloudwatch:GetMetricData`
- `cloudwatch:ListMetrics`
- `tag:GetResources`

Para obtener más información acerca de este rol, consulte Permisos de roles vinculados a un servicio para CloudWatch Application Signals.

Paso 2: crear roles de IAM

Debe crear dos Roles de IAM. Si ya ha creado estos roles, es posible que deba añadirles permisos.

- ECS task role— (Función de tarea de ECS) Los contenedores utilizan esta función para ejecutarse. Los permisos deben ser los que necesiten las aplicaciones, además de CloudWatchAgentServerPolicy y AWSXRayWriteOnlyAccess.
- ECS task execution role (Rol de ejecución de tarea de ECS): Amazon ECS utiliza este rol para lanzar y ejecutar los contenedores. Si ya ha creado este rol, asóciele las políticas AmazonSSMReadOnlyAccess, AmazonECSTaskExecutionRolePolicy y CloudWatchAgentServerPolicy.

  Si necesita almacenar más datos confidenciales para que Amazon ECS los use, consulte Specifying Sensitive Data (Especificación de información confidencial) para obtener más información.

Para obtener más información acerca de cómo crear Roles de IAM, consulte Creating IAM Roles (Creación de Roles de IAM).

Paso 3: preparar la configuración del agente de CloudWatch

En primer lugar, prepare la configuración del agente con Application Signals habilitada. Para ello, cree un archivo local denominado `/tmp/ecs-cwagent.json`.

```
{
  "traces": {
```

```
    "traces_collected": {
      "app_signals": {}
    }
  },
  "logs": {
    "metrics_collected": {
      "app_signals": {}
    }
  }
}
```

Luego, cargue esta configuración en el almacén de parámetros SSM. Para ello, ejecute el siguiente comando. En el archivo, sustituya *$REGION* por el nombre de la región actual.

```
aws ssm put-parameter \
--name "ecs-cwagent" \
--type "String" \
--value "`cat /tmp/ecs-cwagent.json`" \
--region "$REGION"
```

Paso 4: instrumentar la aplicación con el agente de CloudWatch

El siguiente paso es configurar la aplicación para CloudWatch Application Signals.

Java

Para instrumentar la aplicación en Amazon ECS con el agente CloudWatch

1. En primer lugar, especifique un montaje de enlace. El volumen se utilizará para compartir archivos entre contenedores en los siguientes pasos. Usará este montaje de enlace más tarde en este procedimiento.

   ```
   "volumes": [
     {
       "name": "opentelemetry-auto-instrumentation"
     }
   ]
   ```

2. Añada una definición de sidecar de agente de CloudWatch. Para ello, añada un nuevo contenedor llamado `ecs-cwagent` a la definición de tareas de la aplicación. Sustituya *$REGION* por el nombre de la región actual. Sustitúyala por la ruta a la imagen más reciente

del contenedor de CloudWatch en Amazon Elastic Container Registry. Para obtener más información, consulte cloudwatch-agent en Amazon ECR.

```
{
  "name": "ecs-cwagent",
  "image": "$IMAGE",
  "essential": true,
  "secrets": [
    {
      "name": "CW_CONFIG_CONTENT",
      "valueFrom": "ecs-cwagent"
    }
  ],
  "logConfiguration": {
    "logDriver": "awslogs",
    "options": {
      "awslogs-create-group": "true",
      "awslogs-group": "/ecs/ecs-cwagent",
      "awslogs-region": "$REGION",
      "awslogs-stream-prefix": "ecs"
    }
  }
}
```

3. Añada un nuevo contenedor `init` a la definición de tareas de la aplicación. Sustituya *$IMAGE* por la imagen más reciente del repositorio de imágenes AWS Distro para OpenTelemetry de Amazon ECR.

```
{
  "name": "init",
  "image": "$IMAGE",
  "essential": false,
  "command": [
    "cp",
    "/javaagent.jar",
    "/otel-auto-instrumentation/javaagent.jar"
  ],
  "mountPoints": [
    {
      "sourceVolume": "opentelemetry-auto-instrumentation",
      "containerPath": "/otel-auto-instrumentation",
      "readOnly": false
    }
```

```
  ]
}
```

4. Añada las siguientes variables de entorno al contenedor de aplicaciones. Para obtener más información, consulte

Variable de entorno	Configuración para habilitar Application Signals
`OTEL_RESOURCE_ATTRIBUTES`	Sustituya `$SVC_NAME` por el nombre de la aplicación. Se mostrará como el nombre de la aplicación en los paneles de Application Signals. Sustituya `$HOST_ENV` por el entorno host en el que se ejecuta la aplicación. Se mostrará como el entorno alojado en de la aplicación en los paneles de Application Signals.
`OTEL_AWS_APP_SIGNALS_ENABLED`	Configúrelo en `true` para habilitar el procesador SpanMetricsProcessor de Application Signals.
`OTEL_METRICS_EXPORTER`	Configúrelo en `none` para deshabilitar otros exportadores de métricas.
`OTEL_AWS_APP_SIGNALS_EXPORTER_ENDPOINT`	Configúrelo en `http://127.0.0.1:4315` para enviar las métricas al sidecar de CloudWatch.
`OTEL_EXPORTER_OTLP_TRACES_ENDPOINT`	Configúrelo en `http://127.0.0.1:4315` para enviar seguimientos al sidecar de CloudWatch.
`OTEL_TRACES_SAMPLER`	Defina X-Ray como el muestreador de seguimientos.

Variable de entorno	Configuración para habilitar Application Signals
`OTEL_PROPAGATORS`	Añada X-Ray como uno de los propagadores.
`JAVA_TOOL_OPTIONS`	Inyecte el agente Java AWS Distro para OpenTelemetry.

5. Monte el volumen `opentelemetry-auto-instrumentation` que definió en el paso 1 de este procedimiento.

   Para una aplicación Java, utilice lo siguiente.

```
{
  "name": "app",
   ...
  "environment": [
    {
      "name": "OTEL_RESOURCE_ATTRIBUTES",
      "value": "aws.hostedin.environment=$HOST_ENV,service.name=$SVC_NAME"
    },
    {
      "name": "OTEL_AWS_APP_SIGNALS_ENABLED",
      "value": "true"
    },
    {
      "name": "OTEL_METRICS_EXPORTER",
      "value": "none"
    },
    {
      "name": "JAVA_TOOL_OPTIONS",
      "value": " -javaagent:/otel-auto-instrumentation/javaagent.jar"
    },
    {
      "name": "OTEL_AWS_APP_SIGNALS_EXPORTER_ENDPOINT",
      "value": "http://127.0.0.1:4315"
    },
    {
      "name": "OTEL_TRACES_SAMPLER",
      "value": "xray"
    },
```

```
    {
      "name": "OTEL_EXPORTER_OTLP_TRACES_ENDPOINT",
      "value": "http://127.0.0.1:4315"
    },
    {
      "name": "OTEL_PROPAGATORS",
      "value": "tracecontext,baggage,b3,xray"
    }
  ],
  "mountPoints": [
    {
      "sourceVolume": "opentelemetry-auto-instrumentation",
      "containerPath": "/otel-auto-instrumentation",
      "readOnly": false
    }
  ]
}
```

Python

Antes de activar las señales de aplicación para sus aplicaciones Python, debe tener en cuenta las consideraciones siguientes.

- En algunas aplicaciones en contenedores, la falta de una variable de entorno `PYTHONPATH` a veces puede provocar que la aplicación no se inicie. Para solucionar este problema, asegúrese de configurar la variable de entorno `PYTHONPATH` en la ubicación del directorio de trabajo de la aplicación. Esto se debe a un problema conocido con la instrumentación automática de OpenTelemetry. Para obtener más información sobre este problema, consulte Python autoinstrumentation setting of PYTHONPATH is not compliant.
- Para las aplicaciones de Django, se requieren configuraciones adicionales, que se describen en la documentación de Python de OpenTelemetry.
  - Use el indicador `--noreload` para evitar la recarga automática.
  - Establezca la variable de entorno `DJANGO_SETTINGS_MODULE` en la ubicación del archivo `settings.py` de su aplicación Django. Esto garantiza que OpenTelemetry pueda acceder correctamente a la configuración de Django e integrarse correctamente con ella.

Instrumentación de la aplicación Python en Amazon ECS con el agente de CloudWatch

1. En primer lugar, especifique un montaje de enlace. El volumen se utilizará para compartir archivos entre contenedores en los siguientes pasos. Usará este montaje de enlace más tarde en este procedimiento.

```
"volumes": [
  {
    "name": "opentelemetry-auto-instrumentation-python"
  }
]
```

2. Añada una definición de sidecar de agente de CloudWatch. Para ello, añada un nuevo contenedor llamado `ecs-cwagent` a la definición de tareas de la aplicación. Sustituya *$REGION* por el nombre de la región actual. Sustitúyala por la ruta a la imagen más reciente del contenedor de CloudWatch en Amazon Elastic Container Registry. Para obtener más información, consulte cloudwatch-agent en Amazon ECR.

```
{
  "name": "ecs-cwagent",
  "image": "$IMAGE",
  "essential": true,
  "secrets": [
    {
      "name": "CW_CONFIG_CONTENT",
      "valueFrom": "ecs-cwagent"
    }
  ],
  "logConfiguration": {
    "logDriver": "awslogs",
    "options": {
      "awslogs-create-group": "true",
      "awslogs-group": "/ecs/ecs-cwagent",
      "awslogs-region": "$REGION",
      "awslogs-stream-prefix": "ecs"
    }
  }
}
```

3. Añada un nuevo contenedor `init` a la definición de tareas de la aplicación. Sustituya *$IMAGE* por la imagen más reciente del repositorio de imágenes AWS Distro para OpenTelemetry de Amazon ECR.

```
{
    "name": "init",
    "image": "$IMAGE",
    "essential": false,
    "command": [
        "cp",
        "-a",
        "/autoinstrumentation/.",
        "/otel-auto-instrumentation-python"
    ],
    "mountPoints": [
        {
            "sourceVolume": "opentelemetry-auto-instrumentation-python",
            "containerPath": "/otel-auto-instrumentation-python",
            "readOnly": false
        }
    ]
}
```

4. Añada las siguientes variables de entorno al contenedor de aplicaciones. Para obtener más información, consulte

Variable de entorno	Configuración para habilitar Application Signals
`OTEL_RESOURCE_ATTRIBUTES`	Sustituya `$SVC_NAME` por el nombre de la aplicación. Se mostrará como el nombre de la aplicación en los paneles de Application Signals. Sustituya `$HOST_ENV` por el entorno host en el que se ejecuta la aplicación. Se mostrará como el entorno alojado en de la aplicación en los paneles de Application Signals.
`OTEL_AWS_APP_SIGNALS_ENABLED`	Configúrelo en `true` para habilitar el procesador SpanMetricsProcessor de Application Signals.

Variable de entorno	Configuración para habilitar Application Signals
`OTEL_METRICS_EXPORTER`	Configúrelo en `none` para deshabilitar otros exportadores de métricas.
`OTEL_EXPORTER_OTLP_PROTOCOL`	Establézcalo en `http/protobuf` para enviar métricas y seguimientos a CloudWatch mediante HTTP.
`OTEL_AWS_APP_SIGNALS_EXPORTER_ENDPOINT`	Configúrelo en `http://127.0.0.1:4316/v1/metrics` para enviar las métricas al sidecar de CloudWatch.
`OTEL_EXPORTER_OTLP_TRACES_ENDPOINT`	Configúrelo en `http://127.0.0.1:4316/v1/traces` para enviar seguimientos al sidecar de CloudWatch.
`OTEL_TRACES_SAMPLER`	Defina X-Ray como el muestreador de seguimientos.
`OTEL_PROPAGATORS`	Añada X-Ray como uno de los propagadores.
`OTEL_PYTHON_DISTRO`	Establézcalo en `aws_distro` para usar la instrumentación de Python de ADOT.
`OTEL_PYTHON_CONFIGURATOR`	Establézcalo en `aws_configuration` para usar la configuración de Python de ADOT.
`PYTHONPATH`	Sustituya `$APP_PATH` por la ubicación del directorio de trabajo de la aplicación dentro del contenedor. Esto es necesario para que el intérprete Python encuentre los módulos de la aplicación.

Variable de entorno	Configuración para habilitar Application Signals
`DJANGO_SETTINGS_MODULE`	Necesario solo para las aplicaciones Django. Configúrelo en la ubicación del archivo `settings.py` de su aplicación Django. Sustituya `$PATH_TO_SETTINGS` .

5. Monte el volumen `opentelemetry-auto-instrumentation-python` que definió en el paso 1 de este procedimiento.

   Para una aplicación Python, use lo siguiente.

```
{
    "name": "app",
    ...
    "environment": [
        {
            "name": "PYTHONPATH",
            "value": "/otel-auto-instrumentation-python/opentelemetry/
instrumentation/auto_instrumentation:$APP_PATH:/otel-auto-instrumentation-
python"
        },
        {
            "name": "OTEL_EXPORTER_OTLP_PROTOCOL",
            "value": "http/protobuf"
        },
        {
            "name": "OTEL_TRACES_SAMPLER",
            "value": "xray"
        },
        {
            "name": "OTEL_TRACES_SAMPLER_ARG",
            "value": "endpoint=http://localhost:2000"
        },
        {
            "name": "OTEL_LOGS_EXPORTER",
            "value": "none"
        },
        {
            "name": "OTEL_PYTHON_DISTRO",
            "value": "aws_distro"
```

```
        },
        {
            "name": "OTEL_PYTHON_CONFIGURATOR",
            "value": "aws_configurator"
        },
        {
            "name": "OTEL_EXPORTER_OTLP_TRACES_ENDPOINT",
            "value": "http://localhost:4316/v1/traces"
        },
        {
            "name": "OTEL_AWS_APP_SIGNALS_EXPORTER_ENDPOINT",
            "value": "http://localhost:4316/v1/metrics"
        },
        {
            "name": "OTEL_METRICS_EXPORTER",
            "value": "none"
        },
        {
            "name": "OTEL_AWS_APP_SIGNALS_ENABLED",
            "value": "true"
        },
        {
            "name": "OTEL_RESOURCE_ATTRIBUTES",
            "value": "aws.hostedin.environment=$HOST_ENV,service.name=$SVC_NAM"
        },
        {
            "name": "DJANGO_SETTINGS_MODULE",
            "value": "$PATH_TO_SETTINGS.settings"
        }
    ],
    "mountPoints": [
        {
            "sourceVolume": "opentelemetry-auto-instrumentation-python",
            "containerPath": "/otel-auto-instrumentation-python",
            "readOnly": false
        }
    ]
}
```

Paso 5: implementar la aplicación

Cree una nueva revisión de la definición de la tarea e impleméntela en el clúster de aplicaciones. Debería ver tres contenedores en la tarea recién creada:

- `init`
- `ecs-cwagent`
- `app`

Use una configuración personalizada para habilitar Application Signals en Amazon EC2 y otras plataformas

⚠ Application Signals está en versión de prueba. Si tiene algún comentario sobre esta característica, puede ponerse en contacto con nosotros en app-signals-feedback@amazon.com.

Para las aplicaciones que se ejecutan en Amazon EC2 y otras arquitecturas que no son Amazon EKS, debe instalar y configurar usted mismo el agente CloudWatch y AWS Distro para OpenTelemetry. En estas arquitecturas habilitadas con una configuración personalizada de Application Signals, Application Signals no descubre automáticamente los nombres de los servicios ni los hosts o clústeres en los que se ejecutan. Debe especificar estos nombres durante la configuración personalizada, y los nombres que especifique son los que aparecen en los paneles de Application Signals.

Los siguientes pasos se han probado en instancias de Amazon EC2, pero también se espera que funcionen en otras arquitecturas compatibles con AWS Distro para OpenTelemetry.

Requisitos

- Para obtener soporte para Application Signals, debe usar la versión más reciente del agente CloudWatch y del agente AWS Distro para OpenTelemetry.
- Debe tener instalado la AWS CLI en la instancia. Recomendamos la AWS CLI versión 2, pero la versión 1 también debería funcionar. Para obtener información sobre cómo instalar AWS CLI, consulte Instalar o actualizar la versión más reciente de la AWS CLI.

> **Important**
>
> Si ya utiliza OpenTelemetry con una aplicación que pretende habilitar para Application Signals, consulte Consideraciones sobre la compatibilidad de OpenTelemetry antes de activar Application Signals.

Paso 1: habilitar Application Signals en la cuenta

Si aún no ha activado Application Signals en esta cuenta, debe conceder a Application Signals los permisos que necesita para descubrir los servicios. Para ello, haga lo siguiente. Solo es necesario hacerlo una vez para la cuenta.

Para habilitar Application Signals para las aplicaciones

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Servicios.
3. Elija Comenzar a descubrir sus servicios.
4. Seleccione la casilla de verificación y elija Empezar a descubrir servicios.

   Al completar este paso por primera vez en la cuenta, se crea el rol vinculado al servicio AWSServiceRoleForCloudWatchApplicationSignals. Este rol otorga a Application Signals los siguientes permisos:

   - `xray:GetServiceGraph`
   - `logs:StartQuery`
   - `logs:GetQueryResults`
   - `cloudwatch:GetMetricData`
   - `cloudwatch:ListMetrics`
   - `tag:GetResources`

   Para obtener más información acerca de este rol, consulte Permisos de roles vinculados a un servicio para CloudWatch Application Signals.

Paso 2: descargar e iniciar el agente de CloudWatch

Para instalar el agente de CloudWatch como parte de la activación de Application Signals en una instancia de Amazon EC2

1. Descargue la última versión del agente de CloudWatch en la instancia. Si la instancia ya tiene el agente de CloudWatch instalado, es posible que tenga que actualizarlo. Solo las versiones del agente publicadas el 30 de noviembre de 2023 o después son compatibles con CloudWatch Application Signals.

   Para obtener información sobre la descarga del agente de CloudWatch, consulte Descargue del paquete de del agente de CloudWatch.
2. Antes de iniciar el agente de CloudWatch, configúrelo para habilitar Application Signals. El siguiente ejemplo es una configuración de agente de CloudWatch que habilita Application Signals para métricas y seguimientos en un host EC2.

   Puede crear este archivo mediante el siguiente comando:

   ```
   vim amazon-cloudwatch-agent.json
   ```

   Añada lo siguiente como contenido del archivo.

   ```
   {
     "traces": {
       "traces_collected": {
         "app_signals": {}
       }
     },
     "logs": {
       "metrics_collected": {
         "app_signals": {}
       }
     }
   }
   ```
3. Adjunte las políticas de IAM de CloudwatchAgentServerPolicy y AWSXRayWriteOnlyAccess al rol de IAM de la instancia de Amazon EC2.

   a. Inicie sesión en la AWS Management Console y abra la consola de IAM en https://console.aws.amazon.com/iam/.

b. Elija Roles y busque el rol que usa la instancia de Amazon EC2. A continuación, elija el nombre de esa función.

c. En la pestaña Permisos, elija Añadir permisos, Asociar políticas.

d. Busque CloudWatchAgentServerPolicy. Use el cuadro de búsqueda si es necesario. Seleccione la casilla de verificación de la política y elija Añadir permisos.

e. Busque AWSXrayWriteOnlyAccess. Use el cuadro de búsqueda si es necesario. Seleccione la casilla de verificación de la política y elija Añadir permisos.

4. Para iniciar el agente de CloudWatch, introduzca uno de los siguientes comandos. Reemplace *agent-config-file-path* por la ruta al archivo de configuración del agente CloudWatch, como `./amazon-cloudwatch-agent.json`. Debe incluir el prefijo `file:` tal y como se muestra.

```
export CONFIG_FILE_PATH=./amazon-cloudwatch-agent.json
```

```
sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl \
-a fetch-config \
-m ec2 -s -c file:$CONFIG_FILE_PATH
```

Paso 3: instrumentar la aplicación e iníciela

El siguiente paso es configurar la aplicación para CloudWatch Application Signals.

Java

Instrumentación de aplicaciones Java como parte de la activación de Application Signals en una instancia de Amazon EC2

1. Descargue la última versión del agente de autoinstrumentación de AWS Distro para OpenTelemetry de Java. Puede descargar la última versión mediante este enlace. Puede ver información sobre todas las versiones publicadas en aws-otel-java-instrumentation Lanzamientos.
2. Para optimizar las ventajas de Application Signals, utilice variables de entorno para proporcionar información adicional antes de iniciar la aplicación. Esta información se mostrará en los paneles de Application Signals.

a. Para la variable `OTEL_RESOURCE_ATTRIBUTES`, especifique la siguiente información como pares clave-valor:

   - `aws.hostedin.environment` establece el entorno en el que se ejecuta la aplicación. Se mostrará como el entorno alojado en de la aplicación en los paneles de Application Signals. Esta clave de atributo solo la usa Application Signals y se convierte en anotaciones de trazas de X-Ray y dimensiones métricas de CloudWatch. Si no proporciona ningún valor para esta clave, se utiliza el valor predeterminado de `Generic`.
   - `service.name` establece el nombre del servicio. Se mostrará como el nombre del servicio de la aplicación en los paneles de Application Signals. Si no proporciona ningún valor para esta clave, se utiliza el valor predeterminado de `unknown_service`.

b. Para la variable `OTEL_EXPORTER_OTLP_TRACES_ENDPOINT`, especifique la URL del punto de conexión base a la que se van a exportar los seguimientos. El agente CloudWatch expone el 4315 como el puerto OLTP. En Amazon EC2, dado que las aplicaciones se comunican con el agente local de CloudWatch, debe establecer este valor en `OTEL_EXPORTER_OTLP_TRACES_ENDPOINT=http://localhost:4315`

c. Para la variable `OTEL_AWS_APP_SIGNALS_EXPORTER_ENDPOINT`, especifique la URL del punto de conexión base a la que se van a exportar las métricas. El agente CloudWatch expone el 4315 como el puerto OLTP. En Amazon EC2, dado que las aplicaciones se comunican con el agente local de CloudWatch, debe establecer este valor en `OTEL_AWS_APP_SIGNALS_EXPORTER_ENDPOINT=http://localhost:4315`

d. Para la variable `JAVA_TOOL_OPTIONS`, especifique la ruta en la que se almacena el agente de autoinstrumentación AWS Distro para OpenTelemetry de Java.

```
export JAVA_TOOL_OPTIONS=' -javaagent:$ADOT_AGENT_PATH'
```

   Por ejemplo:

```
export ADOT_AGENT_PATH=./aws-opentelemetry-agent.jar
```

e. Para la variable `OTEL_METRICS_EXPORTER`, se recomienda establecer el valor en `none`. Esto deshabilita otros exportadores de métricas para que solo se utilice el exportador de Application Signals.

f. Para la variable `OTEL_AWS_APP_SIGNALS_ENABLED`, habilite el SpanMetricProcessor (SMP) configurándolo en `OTEL_AWS_APP_SIGNALS_ENABLED true`. Esto genera métricas de Application Signals a partir de los seguimientos.

3. Inicie la aplicación con las variables de entorno descritas en el paso anterior. A continuación se muestra un ejemplo de un script de inicio.

```
JAVA_TOOL_OPTIONS=' -javaagent:$ADOT_AGENT_PATH' \
OTEL_METRICS_EXPORTER=none \
OTEL_AWS_APP_SIGNALS_ENABLED=true \
OTEL_AWS_APP_SIGNALS_EXPORTER_ENDPOINT=http://localhost:4315 \
OTEL_EXPORTER_OTLP_TRACES_ENDPOINT=http://localhost:4315 \
OTEL_RESOURCE_ATTRIBUTES=aws.hostedin.environment=$YOUR_HOST_ENV,service.name=
$YOUR_SVC_NAME \
java -jar $MY_JAVA_APP.jar
```

Python

Instrumentación de aplicaciones Python como parte de la activación de Application Signals en una instancia de Amazon EC2

1. Descargue la última versión del agente de autoinstrumentación de AWS Distro para OpenTelemetry de Python. Instálelo ejecutando el siguiente comando de .

```
pip install aws-opentelemetry-distro
```

Puede ver información sobre todas las versiones publicadas en AWS Distro for OpenTelemetry Python instrumentation.

2. Para optimizar las ventajas de Application Signals, utilice variables de entorno para proporcionar información adicional antes de iniciar la aplicación. Esta información se mostrará en los paneles de Application Signals.

a. Para la variable `OTEL_RESOURCE_ATTRIBUTES`, especifique la siguiente información como pares clave-valor:

- `aws.hostedin.environment` establece el entorno en el que se ejecuta la aplicación. Se mostrará como el entorno alojado en de la aplicación en los paneles de Application Signals. Esta clave de atributo solo la usa Application Signals y se convierte en anotaciones de trazas de X-Ray y dimensiones métricas de CloudWatch.

Si no proporciona ningún valor para esta clave, se utiliza el valor predeterminado de `Generic`.

- `service.name` establece el nombre del servicio. Se mostrará como el nombre del servicio de la aplicación en los paneles de Application Signals. Si no proporciona ningún valor para esta clave, se utiliza el valor predeterminado de `unknown_service`.

b. Para la variable `OTEL_EXPORTER_OTLP_PROTOCOL`, especifique `http/protobuf` para exportar datos de telemetría a través de HTTP a los puntos de conexión del agente de CloudWatch que se enumeran en los siguientes pasos.

c. Para la variable `OTEL_EXPORTER_OTLP_TRACES_ENDPOINT`, especifique la URL del punto de conexión base a la que se van a exportar los seguimientos. El agente de CloudWatch expone 4316 como su puerto OLTP a través de HTTP. En Amazon EC2, dado que las aplicaciones se comunican con el agente local de CloudWatch, debe establecer este valor en `OTEL_EXPORTER_OTLP_TRACES_ENDPOINT=http://localhost:4316/v1/traces`

d. Para la variable `OTEL_AWS_APP_SIGNALS_EXPORTER_ENDPOINT`, especifique la URL del punto de conexión base a la que se van a exportar las métricas. El agente de CloudWatch expone 4316 como su puerto OLTP a través de HTTP. En Amazon EC2, dado que las aplicaciones se comunican con el agente local de CloudWatch, debe establecer este valor en `OTEL_AWS_APP_SIGNALS_EXPORTER_ENDPOINT=http://localhost:4316/v1/metrics`

e. Para la variable `OTEL_METRICS_EXPORTER`, se recomienda establecer el valor en `none`. Esto deshabilita otros exportadores de métricas para que solo se utilice el exportador de Application Signals.

f. Para la variable `OTEL_AWS_APP_SIGNALS_ENABLED`, habilite el SpanMetricProcessor configurándo `OTEL_AWS_APP_SIGNALS_ENABLED` en `true`. Esto genera métricas de Application Signals a partir de los seguimientos.

3. Inicie la aplicación con las variables de entorno descritas en el paso anterior. A continuación se muestra un ejemplo de un script de inicio.

- Sustituya `$HOST_ENV` por el entorno host en el que se ejecuta la aplicación. Se mostrará como el entorno Alojado en de la aplicación en los paneles de Application Signals.
- Sustituya `$SVC_NAME` por el nombre de la aplicación. Esto se mostrará como el nombre de la aplicación, en los paneles de Application Signals.
- Sustituya `$PYTHON_APP` por la ubicación y el nombre de la aplicación.

```
OTEL_METRICS_EXPORTER=none \
OTEL_LOGS_EXPORTER=none \
OTEL_AWS_APP_SIGNALS_ENABLED=true \
OTEL_PYTHON_DISTRO=aws_distro \
OTEL_PYTHON_CONFIGURATOR=aws_configurator \
OTEL_EXPORTER_OTLP_PROTOCOL=http/protobuf \
OTEL_TRACES_SAMPLER=xray \
OTEL_TRACES_SAMPLER_ARG="endpoint=http://localhost:2000" \
OTEL_AWS_APP_SIGNALS_EXPORTER_ENDPOINT=http://localhost:4316/v1/metrics \
OTEL_EXPORTER_OTLP_TRACES_ENDPOINT=http://localhost:4316/v1/traces \
OTEL_RESOURCE_ATTRIBUTES=aws.hostedin.environment=$HOST_ENV,service.name=
$SVC_NAME \
opentelemetry-instrument python $PYTHON_APP.py
```

Antes de activar las señales de aplicación para sus aplicaciones Python, debe tener en cuenta las consideraciones siguientes.

- En algunas aplicaciones en contenedores, la falta de una variable de entorno `PYTHONPATH` a veces puede provocar que la aplicación no se inicie. Para solucionar este problema, asegúrese de configurar la variable de entorno `PYTHONPATH` en la ubicación del directorio de trabajo de la aplicación. Esto se debe a un problema conocido con la instrumentación automática de OpenTelemetry. Para obtener más información sobre este problema, consulte Python autoinstrumentation setting of PYTHONPATH is not compliant.
- Para las aplicaciones de Django, se requieren configuraciones adicionales, que se describen en la documentación de Python de OpenTelemetry.
  - Use el indicador `--noreload` para evitar la recarga automática.
  - Establezca la variable de entorno `DJANGO_SETTINGS_MODULE` en la ubicación del archivo `settings.py` de su aplicación Django. Esto garantiza que OpenTelemetry pueda acceder correctamente a la configuración de Django e integrarse correctamente con ella.

Solución de problemas de instalación de Application Signals

⚠ Application Signals está en versión de prueba. Si tiene algún comentario sobre esta característica, puede ponerse en contacto con nosotros en app-signals-feedback@amazon.com.

Esta sección contiene sugerencias para solucionar los problemas de CloudWatch Application Signals.

Temas

La aplicación no se inicia después de activar Application Signals

Si la aplicación en un clúster de Amazon EKS no se inicia después de habilitar Application Signals en el clúster, compruebe lo siguiente:

- Compruebe si la aplicación ha sido instrumentada por otra solución de supervisión. Application Signals no admite la coexistencia con otras soluciones de instrumentación.
- Confirme que la aplicación cumpla con los requisitos de compatibilidad para utilizar Application Signals. Para obtener más información, consulte Sistemas compatibles con Application Signals .
- Si la aplicación no pudo extraer los artefactos de Application Signals, como las imágenes del agente Java o Python de AWS Distro para OpenTelemetery y del agente de CloudWatch, podría deberse a un problema de red.

Para mitigar el problema, elimine la anotación `instrumentation.opentelemetry.io/inject-java: "true"` o `instrumentation.opentelemetry.io/inject-python: "true"` del manifiesto de implementación de la aplicación y vuelva a implementarla. A continuación, compruebe si la aplicación funciona.

La aplicación Python no se inicia después de activar Application Signals

Es un problema conocido en la instrumentación automática de OpenTelemetry que una variable de entorno `PYTHONPATH` faltante a veces puede provocar que la aplicación no se inicie. Para resolver esto, asegúrese de configurar la variable de entorno `PYTHONPATH` en la ubicación del directorio de trabajo de su aplicación. Para obtener más información sobre este problema, consulte Python autoinstrumentation setting of PYTHONPATH is not compliant with Python's module resolution behavior, breaking Django applications.

Para las aplicaciones de Django, se requieren configuraciones adicionales, que se describen en la documentación de Python de OpenTelemetry.

- Use el indicador `--noreload` para evitar la recarga automática.
- Establezca la variable de entorno `DJANGO_SETTINGS_MODULE` en la ubicación del archivo `settings.py` de su aplicación Django. Esto garantiza que OpenTelemetry pueda acceder correctamente a la configuración de Django e integrarse correctamente con ella.

Faltan datos de telemetría en CloudWatch y X-Ray

Si faltan métricas o seguimientos en los paneles de Application Signals, las causas podrían ser las siguientes. Investigue estas causas solo si ha esperado 15 minutos para que Application Signals recopile y muestre datos desde la última actualización.

- Asegúrese de que la biblioteca y el marco que está utilizando sean compatibles con el agente Java de ADOT. Para obtener más información, consulte Bibliotecas/Marcos.
- Asegúrese de que el agente de CloudWatch esté en ejecución. En primer lugar, compruebe el estado de los pods de agentes de CloudWatch y asegúrese de que todos estén en estado `Running`.

  ```
  kubectl -n amazon-cloudwatch get pods.
  ```

  Añada lo siguiente al archivo de configuración del agente de CloudWatch para habilitar los registros de depuración y, a continuación, reinicie el agente.

  ```
  "agent": {
  >>>>>>> streams
    "region": "${REGION}",
    "debug": true
  ```

```
},
```

A continuación, compruebe si hay errores en los pods de agentes de CloudWatch.

- Compruebe si hay problemas de configuración con el agente de CloudWatch. Confirme que lo siguiente sigue en el archivo de configuración del agente de CloudWatch y que el agente se ha reiniciado desde que se añadió.

```
"agent": {
  "region": "${REGION}",
  "debug": true
},
```

A continuación, consulte los registros de depuración de OpenTelemetry para ver si hay mensajes de error como `ERROR io.opentelemetry.exporter.internal.grpc.OkHttpGrpcExporter - Failed to export ...`. Estos mensajes pueden indicar el problema.

Si eso no resuelve el problema, vuelque y compruebe las variables de entorno con nombres que comiencen por `OTEL_` describiendo el pod con el comando `kubectl describe pod`.

- Para habilitar el registro de depuración de OpenTelemetry Python, establezca la variable de entorno `OTEL_PYTHON_LOG_LEVEL` en `debug` y vuelva a implementar la aplicación.
- Compruebe si hay permisos incorrectos o insuficientes para exportar datos desde el agente de CloudWatch. Si ve mensajes `Access Denied` en los registros del agente de CloudWatch, puede que este sea el problema. Es posible que los permisos aplicados al instalar el agente de CloudWatch se hayan modificado o revocado posteriormente.
- Compruebe si hay un problema de AWS Distro para OpenTelemetry (ADOT) al generar datos de telemetría.

  Asegúrese de que las anotaciones de la instrumentación `instrumentation.opentelemetry.io/inject-java` y `sidecar.opentelemetry.io/inject-java` se apliquen a la implementación de la aplicación y que el valor sea `true`. Sin ellas, los pods de aplicaciones no se instrumentarán aunque el complemento ADOT esté instalado correctamente.

  A continuación, compruebe si el contenedor `Init` está aplicado a la aplicación y su estado `Ready` sea `True`. Si el contenedor `init` no está listo, consulte el estado para ver el motivo.

Si el problema persiste, haga lo siguiente para habilitar el registro de depuración en el Java SDK de OpenTelemetry. A continuación, busque los mensajes que comiencen por `ERROR io.telemetry`.

Para habilitar el registro de depuración, defina la variable de entorno `OTEL_JAVAAGENT_DEBUG` en true y vuelva a implementar la aplicación.

- Es posible que el exportador de métricas o intervalos esté descartando datos. Para averiguarlo, consulte el registro de la aplicación para ver si hay mensajes que incluyan `Failed to export...`
- Es posible que el agente de CloudWatch se esté viendo limitado al enviar métricas o intervalos a Application Signals. Compruebe si hay mensajes que indiquen una limitación en los registros del agente de CloudWatch.

Las métricas de dependencia tienen valores desconocidos

Si ve UnknownOperation, UnknownRemoteService o UnknownRemoteOperation como nombre de dependencia u operación en los paneles de Application Signals, compruebe si la aparición de puntos de datos para el servicio remoto desconocido y la operación remota desconocida coinciden con su implementación. Es un problema conocido en Application Signals y está previsto que se corrija en una versión futura.

Gestión de un conflicto de configuración al administrar el complemento de observabilidad de Amazon CloudWatch EKS

Al instalar o actualizar el complemento de observabilidad de Amazon CloudWatch EKS, si observa un error provocado por un `Health Issue` del tipo `ConfigurationConflict` con una descripción que comienza por `Conflicts found when trying to apply. Will not continue due to resolve conflicts mode`, probablemente se deba a que ya tiene el agente de CloudWatch y sus componentes asociados, como ServiceAccount, ClusterRole y ClusterRoleBinding instalados en el clúster. Cuando el complemento intenta instalar el agente de CloudWatch y sus componentes asociados, si detecta algún cambio en el contenido, por defecto no se realiza la instalación o la actualización para evitar sobrescribir el estado de los recursos del clúster.

Si está intentando incorporar al complemento de observabilidad de ECS de Amazon CloudWatch y observa este error, le recomendamos que elimine la configuración de agente de CloudWatch existente que haya instalado anteriormente en el clúster y, a continuación, instale el complemento

EKS. Asegúrese de hacer una copia de seguridad de las personalizaciones que haya realizado en la configuración original del agente de CloudWatch, como una configuración de agente personalizada, y envíelas al complemento de observabilidad de Amazon CloudWatch EKS la próxima vez que lo instale o actualice. Si ya había instalado el agente de CloudWatch para incorporar Información de contenedores, consulte Eliminación del agente de CloudWatch y Fluen Bit para Información de contenedores para obtener más información.

Como alternativa, el complemento admite una opción de configuración de resolución de conflictos que puede especificar `OVERWRITE`. Puede usar esta opción para continuar con la instalación o actualización del complemento sobrescribiendo los errores en el clúster. Si utiliza la consola de Amazon EKS, encontrará el método de resolución de errores al elegir los ajustes de configuración opcionales al crear o actualizar el complemento. Si está utilizando la AWS CLI, puede proporcionar `--resolve-conflicts OVERWRITE` al comando para crear o actualizar el complemento.

Configuración de Application Signals

> ⚠ Application Signals está en versión de prueba. Si tiene algún comentario sobre esta característica, puede ponerse en contacto con nosotros en app-signals-feedback@amazon.com.

Esta sección contiene información sobre la configuración de CloudWatch Application Signals.

Frecuencia de muestreo de trazas

De forma predeterminada, al activar Application Signals, el muestreo centralizado de X-Ray se habilita con los ajustes de frecuencia de muestreo predeterminados de `reservoir=1/s` y `fixed_rate=5%`. Las variables de entorno del agente SDK de AWS Distro para OpenTelemetry (ADOT) se configuran de la siguiente manera.

Variable de entorno	Valor	Nota
`OTEL_TRACES_SAMPLER`	`xray`	
`OTEL_TRACES_SAMPLER_ARG`	`endpoint=http://cloudwatch-agent.amazon-cloudwatch:2000`	Punto de conexión del agente de CloudWatch

Para obtener información sobre cómo cambiar la configuración de muestreo, consulte lo siguiente:

- Para cambiar el muestreo de X-Ray, consulte Personalización de las reglas de muestreo.
- Para cambiar el muestreo de ADOT, consulte Configuración del recopilador OpenTelemetry para el muestreo remoto de X-Ray.

Si desea deshabilitar el muestreo centralizado de X-Ray y utilizar el muestreo local en su lugar, defina los siguientes valores para el agente Java del SDK de ADOT como se indica a continuación. El siguiente ejemplo establece la frecuencia de muestreo en un 5 %.

Variable de entorno	Valor
`OTEL_TRACES_SAMPLER`	`parentbased_traceidratio`
`OTEL_TRACES_SAMPLER_ARG`	`0.05`

Para obtener información sobre ajustes de muestreo más avanzados, consulte OTEL_TRACES_SAMPLER.

Administración de operaciones de alta cardinalidad

Application Signals incluye configuraciones en el agente de CloudWatch que puede usar para administrar la cardinalidad de sus operaciones y administrar la exportación de métricas para optimizar los costos. De forma predeterminada, la función de limitación de métricas se activa cuando el número de operaciones distintas de un servicio a lo largo del tiempo supera el umbral predeterminado de 500. Puede ajustar el comportamiento ajustando los ajustes de configuración.

Comprobación de si la limitación de métricas está activada

Puede utilizar uno de los métodos siguientes para comprobar si se está aplicando el límite de métricas predeterminado. Si es así, debe considerar la posibilidad de optimizar el control de cardinalidad siguiendo los pasos que se indican en la siguiente sección.

- En la consola de CloudWatch, elija Application Signals, Servicios. Si ve una Operación llamada AllOtherOperations o una RemoteOperation llamada AllOtherRemoteOperations, significa que se está limitando la métrica.
- Si alguna métrica recopilada por Application Signals tiene el valor `AllOtherOperations` de su dimensión `Operation`, entonces se está produciendo una limitación de la métrica.

- Si alguna métrica recopilada por Application Signals tiene el valor `AllOtherRemoteOperations` de su dimensión `RemoteOperation`, entonces se está produciendo una limitación de la métrica.

Optimización del control de cardinalidad

Para optimizar el control de cardinalidad, puede hacer lo siguiente:

- Cree reglas personalizadas para agregar operaciones.
- Configure su política de limitación de métricas.

Creación de reglas personalizadas para agregar operaciones

A veces, las operaciones de alta cardinalidad pueden deberse a valores únicos inapropiados extraídos del contexto. Por ejemplo, enviar solicitudes HTTP/S que incluyan ID de usuario o ID de sesión en la ruta puede provocar cientos de operaciones dispares. Para resolver estos problemas, recomendamos configurar el agente de CloudWatch con reglas de personalización para volver a escribir estas operaciones.

En los casos en los que se generen numerosas métricas diferentes a través de llamadas `RemoteOperation` individuales, por ejemplo `PUT /api/customer/owners/123`, `PUT /api/customer/owners/456` y solicitudes similares, le recomendamos que consolide estas operaciones en una sola `RemoteOperation`. Un enfoque es estandarizar todas las llamadas `RemoteOperation` que comienzan con `PUT /api/customer/owners/` a un formato uniforme, específicamente `PUT /api/customer/owners/{ownerId}`. En el siguiente ejemplo, se ilustra este caso. Para obtener información sobre otras reglas de personalización, consulte Habilitación de CloudWatch Application Signals.

```
{
   "logs":{
      "metrics_collected":{
         "app_signals":{
            "rules":[
               {
                  "selectors":[
                     {
                        "dimension":"RemoteOperation",
                        "match":"PUT /api/customer/owners/*"
                     }
                  ],
```

```
                "replacements":[
                   {
                      "target_dimension":"RemoteOperation",
                      "value":"PUT /api/customer/owners/{ownerId}"
                   }
                ],
                "action":"replace"
             }
          ]
       }
    }
 }
}
```

En otros casos, es posible que las métricas de alta cardinalidad se hayan agregado a `AllOtherRemoteOperations` y es posible que no esté claro qué métricas específicas se incluyen. El agente de CloudWatch puede registrar las operaciones interrumpidas. Para identificar las operaciones interrumpidas, utilice la configuración del siguiente ejemplo para activar el registro hasta que el problema vuelva a surgir. A continuación, inspeccione los registros del agente de CloudWatch (a los que se puede acceder mediante un contenedor `stdout` o archivos de registro de EC2) y busque la palabra clave `drop metric data`.

```
{
  "agent": {
    "config": {
      "agent": {
        "debug": true
      },
      "traces": {
        "traces_collected": {
          "app_signals": {
          }
        }
      },
      "logs": {
        "metrics_collected": {
          "app_signals": {
            "limiter": {
              "log_dropped_metrics": true
            }
          }
        }
```

```
      }
    }
  }
}
```

Creación de su política de limitación de métricas

Si la configuración de limitación de métricas predeterminada no aborda la cardinalidad de su servicio, puede personalizar la configuración del limitador de métricas. Para configurar esta acción, agregue una sección `limiter` en la sección `logs/metrics_collected/app_signals` del archivo de configuración del agente de CloudWatch.

El siguiente ejemplo reduce el umbral del límite de métricas de 500 métricas distintas a 100.

```
{
  "logs": {
    "metrics_collected": {
      "app_signals": {
        "limiter": {
          "drop_threshold": 100
        }
      }
    }
  }
}
```

Objetivos de nivel de servicio (SLO)

⚠ Application Signals está en versión de prueba. Si tiene algún comentario sobre esta característica, puede ponerse en contacto con nosotros en app-signals-feedback@amazon.com.

Puede usar Application Signals para crear objetivos de nivel de servicio para los servicios de las operaciones comerciales críticas. Al crear SLO para estos servicios, podrá realizar un seguimiento de los mismos en el panel de control de los SLO, lo que le permitirá tener una vista rápida de sus operaciones más importantes.

Además de crear una vista rápida que sus operadores pueden utilizar para ver el estado actual de las operaciones críticas, puede utilizar los SLO para realizar un seguimiento del rendimiento a largo plazo de los servicios y asegurarse de que cumplen sus expectativas. Si tiene acuerdos de nivel de servicio con los clientes, los SLO son una excelente herramienta para asegurar su cumplimiento.

La evaluación del estado de sus servicios con los SLO comienza con el establecimiento de objetivos claros y medibles basados en métricas de rendimiento clave: indicadores de nivel de servicio (SLI). Un SLO hace un seguimiento del rendimiento del SLI en relación con el umbral y el objetivo que establezca, e informa qué tan lejos o cerca está el rendimiento de la aplicación del umbral.

Application Signals lo ayuda a establecer los SLO en sus métricas de rendimiento clave. Application Signals recopila de forma automática las métricas de `Latency` y `Availability` de cada servicio y operación que detecta, y estas métricas suelen ser ideales para utilizarlas como SLI. Con el asistente de creación de SLO, puede utilizar estas métricas para sus SLO. A continuación, puede realizar un seguimiento del estado de todos los SLO con los paneles de Application Signals.

Puede configurar los SLO para operaciones específicas a las que llame o utilice su servicio. Puede usar cualquier métrica o expresión métrica de CloudWatch como un SLI, además de usar las métricas de `Latency` y `Availability`.

La creación de SLO es muy importante para aprovechar al máximo CloudWatch Application Signals. Después de crear los SLO, puede ver su estado en la consola de Application Signals para ver rápidamente cuáles de estos servicios y operaciones esenciales funcionan de forma correcta y cuáles no. El seguimiento con los SLO ofrece las siguientes ventajas principales:

- A los operadores de servicios les resulta más fácil comparar el estado operativo actual de los servicios críticos en comparación con el SLI. Luego, pueden clasificar e identificar con rapidez los servicios y las operaciones que no funcionen de forma correcta.
- Puede realizar un seguimiento del rendimiento de sus servicios en relación con objetivos empresariales cuantificables durante períodos de tiempo más largos.

Al elegir en qué aspectos establecer los SLO, prioriza lo que es importante para usted. Los paneles de Application Signals presentan automáticamente información sobre lo que ha priorizado.

Al crear un SLO, también puede optar por crear alarmas de CloudWatch al mismo tiempo para supervisar los SLO. Puede establecer alarmas que supervisen los incumplimientos del umbral y también los niveles de advertencia. Estas alarmas pueden notificarle automáticamente si las métricas de los SLO superan el umbral que estableció o si se acercan a un umbral de advertencia. Por

ejemplo, un SLO que se acerca a su umbral de advertencia puede indicarle que su equipo podría necesitar reducir la pérdida de la aplicación para asegurarse de que se cumplen los objetivos de rendimiento a largo plazo.

Temas

Conceptos del SLO

Un SLO incluye los siguientes componentes:

- Un indicador de nivel de servicio (SLI), que es una métrica de rendimiento clave que se especifica. Representa el nivel de rendimiento deseado para su aplicación. Application Signals recopila de forma automática las métricas de `Latency` y `Availability` para los servicios y operaciones que detecta, y estas métricas suelen ser ideales para establecer los SLO.

  Usted elige el umbral que desea usar para su SLI. Por ejemplo, 200 ms de latencia.
- Un objetivo o un objetivo de rendimiento, que es el porcentaje de tiempo que se espera para que el SLI alcance el umbral en cada intervalo de tiempo. Los intervalos de tiempo pueden ser tan cortos como horas o tan largos como un año.

  Los intervalos pueden ser intervalos de calendario o intervalos continuos.
  - Los intervalos de calendario se alinean con el calendario, como un SLO del que se hace un seguimiento por mes. CloudWatch ajusta automáticamente las cifras del estado, presupuesto y rendimiento en función de la cantidad de días de un mes. Los intervalos de calendario son más adecuados para los objetivos empresariales que se miden de forma alineada con el calendario.
  - Los intervalos continuos se calculan de forma continua. Los intervalos continuos son más adecuados para realizar un seguimiento de la experiencia reciente de los usuarios en la aplicación.
- El período es un período de tiempo más corto y muchos períodos forman un intervalo. El rendimiento de la aplicación se compara con el del SLI durante cada período del intervalo. Para cada período, se determina que la aplicación ha alcanzado o no el rendimiento necesario.

Por ejemplo, un objetivo del 99 % con un intervalo de calendario de un día y un período de 1 minuto significa que la aplicación debe cumplir o alcanzar el umbral de éxito durante el 99 % de los períodos de 1 minuto del día. Si es así, se cumplen los SLO de ese día. Al día siguiente hay un nuevo intervalo de evaluación y la aplicación debe cumplir o alcanzar el umbral de éxito durante el 99 % de los períodos de 1 minuto del segundo día para cumplir con los SLO correspondientes a ese segundo día.

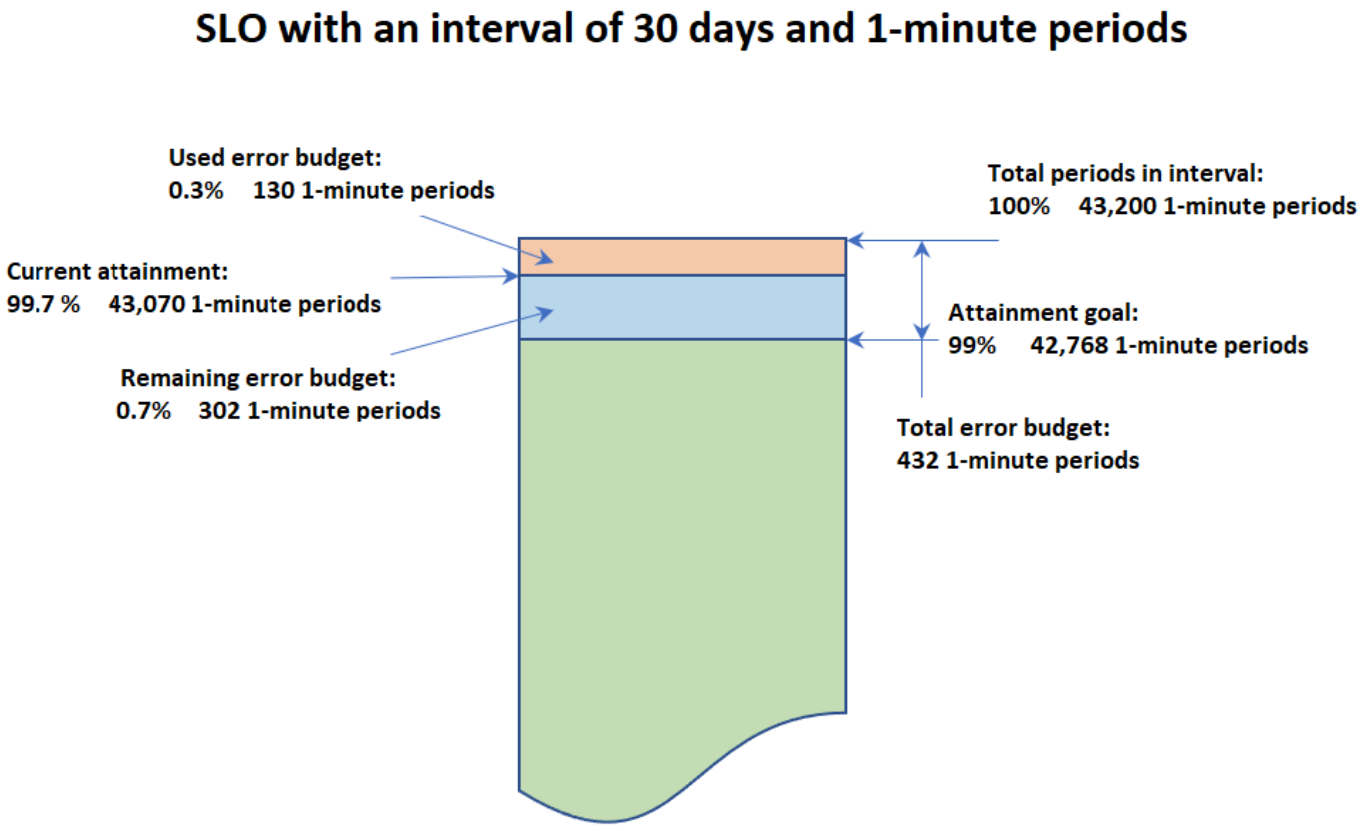
Un SLI puede basarse en una de las nuevas métricas de aplicación estándar recopiladas por Application Signals. Alternativamente, puede ser cualquier métrica o expresión métrica de CloudWatch. Las métricas de aplicación estándar que puede usar para un SLI son `Latency` y `Availability`. `Availability` representa las respuestas satisfactorias divididas entre el total de solicitudes. Se calcula como (1: tasa de errores)*100, donde las respuestas a los errores son errores `5xx`. Las respuestas correctas son respuestas sin errores `5XX`. Las respuestas `4XX` se consideran satisfactorias.

> **Note**
>
> Actualmente, solo se admiten los cálculos basados en períodos. En futuras versiones se prevé la posibilidad de realizar cálculos basados en volúmenes o solicitudes.

Cálculo del presupuesto de errores y rendimiento

Cuando consulta la información sobre un SLO, ve su estado de funcionamiento actual y el presupuesto de errores. El presupuesto de errores es la cantidad de tiempo dentro del intervalo que puede superar el umbral y, aun así, permitir que se cumpla el SLO. El presupuesto total de errores es la cantidad total de tiempo de incumplimiento que se puede tolerar durante todo el intervalo. El presupuesto de errores restante es la cantidad restante de tiempo de incumplimiento que se puede tolerar durante el intervalo actual. Esto ocurre después de restar del total del presupuesto de errores el tiempo de incumplimiento que ya transcurrió.

La siguiente figura ilustra los conceptos del rendimiento y el presupuesto de errores para un objetivo con un intervalo de 30 días, períodos de 1 minuto y un objetivo de rendimiento del 99 %. 30 días incluyen 43 200 períodos de 1 minuto. El 99 % de 43 200 es 42 768, por lo que 42 768 minutos durante el mes deben funcionar de forma correcta para cumplir con el SLO. En lo que va del intervalo actual, 130 de los períodos de 1 minuto no funcionaban de forma correcta.



Determinación del éxito dentro de cada período

Dentro de cada período, los datos del SLI se añaden en un único punto de datos en función de la estadística utilizada para el SLI. Este punto de datos representa la duración total del período. Ese único punto de datos se compara con el umbral del SLI para determinar si el período es correcto. Si aparecen períodos incorrectos durante el intervalo de tiempo actual en el panel de control, los operadores de servicio pueden avisar que es necesario clasificar el servicio.

Si se determina que el período es incorrecto, toda la duración del período se considera fallida y se tiene en cuenta en el presupuesto de errores. El seguimiento del presupuesto de errores le permite saber si el servicio logra el rendimiento que desea durante un período de tiempo más prolongado.

Creación de un SLO

Le recomendamos que establezca los SLO de latencia y disponibilidad en sus aplicaciones críticas. Estas métricas recopiladas por Application Signals se alinean con los objetivos empresariales comunes.

También puede establecer los SLO en cualquier métrica de CloudWatch o en cualquier expresión matemática métrica que dé como resultado una serie temporal individual.

La primera vez que cree un SLO en su cuenta, CloudWatch creará automáticamente el rol vinculado al servicio AWSServiceRoleForCloudwatchApplicationSignals en su cuenta, si aún no existe. Este rol vinculado al servicio permite a CloudWatch recopilar datos de Registros de CloudWatch, datos de registros de seguimiento de X-Ray, datos de métricas de CloudWatch y datos de etiquetado de las aplicaciones de su cuenta. Para obtener más información acerca de los roles vinculados a servicio de CloudWatch, consulte Uso de roles vinculados a servicios para CloudWatch.

Cómo crear un SLO

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Objetivos de nivel de servicio (SLO).
3. Seleccione Crear SLO.
4. Escriba un nombre para el SLO. Incluir el nombre de un servicio u operación, junto con las palabras clave adecuadas, como latencia o disponibilidad, lo ayudará a identificar rápidamente lo que indica el estado del SLO durante la clasificación.
5. En Establecer el indicador de nivel de servicio (SLI), realice una de las siguientes operaciones:
   - Para establecer el SLO en cualquiera de las métricas de la aplicación estándar `Latency` o `Availability`:
     a. Seleccione Operación de servicio.
     b. Seleccione el servicio que monitoreará este SLO.
     c. Seleccione la operación que monitoreará este SLO.

        Los menús desplegables Seleccionar servicio y Seleccionar operación se rellenan con los servicios y operaciones que han estado activos en las últimas 24 horas.
     d. Seleccione Disponibilidad o Latencia y, a continuación, establezca el umbral.
   - Para establecer el SLO en cualquier métrica de CloudWatch o en una expresión matemática métrica de CloudWatch:
     a. Seleccione Métrica de CloudWatch.
     b. Haga clic en Seleccionar métrica de CloudWatch.

        Aparece la pantalla Seleccionar métrica. Utilice las pestañas Examinar o Consulta para buscar la métrica que desee o cree una expresión matemática métrica.

Después de seleccionar la métrica que desee, seleccione la pestaña Métricas diagramadas y seleccione la Estadística y el Período que desee usar para el SLO. A continuación, elija Select metric (Seleccionar métrica).

Para obtener más información sobre estas pantallas, consulte Representar gráficamente una métrica y Añadir una expresión matemática a un gráfico de CloudWatch.

c. En Establecer condiciones, seleccione un operador de comparación y un umbral para que el SLO lo utilice como indicador de éxito.

6. Si seleccionó Operación de servicio en el paso 5, si lo desea, puede seleccionar Ajustes adicionales y, a continuación, ajustar la duración del período de este SLO.

7. Establezca el intervalo y el objetivo de rendimiento del SLO. Para obtener más información sobre los intervalos y los objetivos de rendimiento y cómo funcionan de forma conjunta, consulte Conceptos del SLO.

8. (Opcional) Establezca una o más alarmas de CloudWatch o un umbral de advertencia para el SLO.

   a. Las alarmas de CloudWatch pueden utilizar Amazon SNS para notificarle de forma proactiva si una aplicación no funciona de forma correcta en función del rendimiento del SLI.

   Para crear una alarma, seleccione una de las casillas de verificación de la alarma e introduzca o cree el tema de Amazon SNS para usarlo en las notificaciones cuando la alarma entre en estado `ALARM`. Para obtener más información acerca de las alarmas de CloudWatch, consulte Uso de las alarmas de Amazon CloudWatch. La creación de alarmas genera cargos. Para obtener más información sobre los precios de CloudWatch, consulte Precios de Amazon CloudWatch.

   b. Si establece un umbral de advertencia, este aparecerá en las pantallas de Application Signals para ayudarlo a identificar los SLO que corren el riesgo de no cumplirse, incluso si actualmente son correctos.

   Para establecer un umbral de advertencia, introduzca el valor del umbral en Umbral de advertencia. Cuando el presupuesto de error del SLO es inferior al umbral de advertencia, el SLO se marca con una Advertencia en varias pantallas de Application Signals. Los umbrales de advertencia también aparecen en los gráficos del presupuesto de errores. También puede crear una alarma de advertencia de SLO que se base en el umbral de advertencia.

9. Para añadir etiquetas a este SLO, seleccione la pestaña Etiquetas y, a continuación, seleccione Añadir nueva etiqueta. Las etiquetas pueden ayudarle a administrar, identificar, organizar, buscar y filtrar recursos. Para obtener más información sobre el etiquetado, consulte Etiquetado de los recursos de AWS.

   > **Note**
   >
   > Si la aplicación con la que se relaciona este SLO está registrada en AWS Service Catalog AppRegistry, puede usar la etiqueta `awsApplication` para asociar este SLO a esa aplicación en AppRegistry. Para más información, consulte ¿Qué es AppRegistry?

10. Seleccione Crear SLO. Si también opta por crear una o más alarmas, el nombre del botón cambiará para reflejarlo.

Visualización y clasificación del estado del SLO

Puede ver rápidamente el estado de sus SLO mediante los Objetivos de nivel de servicio o las opciones de Servicios en la consola de CloudWatch. La vista de Servicios ofrece una vista rápida de la proporción de servicios que no funcionan de forma correcta, calculada en función de los SLO que haya establecido. Para obtener más información acerca del uso de la opción Servicios, consulte Monitoreo del estado operativo de sus aplicaciones con Application Signals.

La vista de los Objetivos de nivel de servicio proporciona una vista global de su organización. Puede ver los SLO cumplidos y no cumplidos en su conjunto. Esto le permite ver cuántos servicios y operaciones están cumpliendo sus expectativas durante los períodos de tiempo más largos, en función de los SLI que haya elegido.

Cómo ver todos los SLO mediante la vista Objetivos de nivel de servicio

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Objetivos de nivel de servicio (SLO).

   Aparece la lista de Objetivos de nivel de servicio (SLO).

   Puede ver rápidamente el estado actual de sus SLO en la columna Estado del SLI. Para ordenar los SLO de manera que todos los SLO que no funcionen de forma correcta estén al principio de la lista, seleccione la columna Estado del SLI hasta que todos los SLO que no funcionen de forma correcta estén en la parte superior.

La tabla de SLO tiene las siguientes columnas predeterminadas. Puede ajustar las columnas que se muestran al elegir el icono de engranaje que aparece en la parte superior de la lista. Para obtener más información sobre los objetivos, los SLI, el rendimiento y los intervalos, consulte Conceptos del SLO.

- El nombre del SLO.
- La columna Objetivo muestra el porcentaje de períodos durante cada intervalo que deben alcanzar de forma satisfactoria el umbral del SLI para cumplir el objetivo del SLO. También muestra la duración del intervalo del SLO.
- El estado del SLI muestra si el estado operativo actual de la aplicación es correcto o no. Si algún período del intervalo de tiempo seleccionado actualmente no era correcto para el SLO, el estado del SLI mostrará Incorrecto.
- El rendimiento final es el nivel de rendimiento alcanzado al final del intervalo de tiempo seleccionado. Ordene según esta columna para ver los SLO que corren mayor riesgo de no cumplirse.
- El Delta de rendimiento es la diferencia en el nivel de rendimiento entre el inicio y el final del intervalo de tiempo seleccionado. Un delta negativo significa que la métrica tiene una tendencia descendente. Ordene según esta columna para ver las últimas tendencias de los SLO.
- El Presupuesto de errores finales (%) es el porcentaje del tiempo total del período que puede tener períodos incorrectos y, aun así, lograr los SLO con éxito. Si lo establece en un 5 % y el SLI es incorrecto en un 5 % o por debajo de los períodos restantes del intervalo, los SLO todavía se cumplen de forma correcta.
- El Delta del presupuesto de errores es la diferencia en el presupuesto de errores entre el inicio y el final del intervalo de tiempo seleccionado. Un delta negativo significa que la métrica tiende a fallar.
- El Presupuesto de errores finales (tiempo) es la cantidad de tiempo real del intervalo que puede resultar incorrecto y, aun así, permitir que el SLO se cumpla de forma correcta. Por ejemplo, si es de 14 minutos, y el SLI es incorrecto durante menos de 14 minutos en el intervalo restante, el SLO se seguirá cumpliendo de forma correcta.
- Las columnas Servicio, Operación y Tipo muestran información sobre el servicio y la operación para los que se ha establecido este SLO.

3. Para ver los gráficos del rendimiento y el presupuesto de errores de un SLO, seleccione el botón de opción situado junto al nombre del SLO.

Los gráficos en la parte superior de la página muestran el estado del rendimiento y el presupuesto de errores del SLO. También se muestra un gráfico sobre la métrica del SLI asociada a este SLO.

4. Para seguir clasificando un SLO que no cumple su objetivo, elija el nombre del servicio o el nombre de la operación correspondiente a ese SLO. Se lo redirigirá a la página de detalles, donde podrá seguir clasificando. Para obtener más información, consulte Visualización de la actividad detallada del servicio y el estado operativo en la página de detalles del servicio.
5. Para cambiar el intervalo de tiempo de los gráficos y las tablas de la página, seleccione un nuevo intervalo de tiempo cerca de la parte superior de la pantalla.

Edición de un SLO existente

Siga los siguientes pasos para editar un SLO existente. Al editar un SLO, solo puede cambiar el umbral, el intervalo, el objetivo de rendimiento y las etiquetas. Para cambiar otros aspectos, como el servicio, la operación o la métrica, cree un SLO nuevo en lugar de editar uno existente.

Al cambiar parte de la configuración básica de un SLO, como el período o el umbral, se invalidan todos los puntos de datos y las evaluaciones anteriores sobre el rendimiento y el estado. Elimina y vuelve a crear de forma efectiva el SLO.

> **Note**
>
> Si edita un SLO, las alarmas asociadas a ese SLO no se actualizan de forma automática. Es posible que tenga que actualizar las alarmas para mantenerlas sincronizadas con el SLO.

Cómo editar un SLO existente

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Objetivos de nivel de servicio (SLO).
3. Seleccione el botón de opción situado junto al SLO que desea editar y haga clic en Acciones, Editar SLO.
4. Realice los cambios y, a continuación, elija Guardar cambios.

Eliminación de un SLO

Siga los siguientes pasos para eliminar un SLO existente.

> **Note**
>
> Al eliminar un SLO, las alarmas asociadas a ese SLO no se eliminan de forma automática. Tendrá que eliminarlas usted mismo. Para obtener más información, consulte Administración de alarmas.

Cómo eliminar un SLO

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Objetivos de nivel de servicio (SLO).
3. Elija el botón de opción situado junto al SLO que desea editar y elija Acciones, Eliminar SLO.
4. Elija Confirmar.

Monitoreo del estado operativo de sus aplicaciones con Application Signals

> **Application Signals está en versión preliminar. Si tiene algún comentario sobre esta característica, puede ponerse en contacto con nosotros en app-signals-feedback@amazon.com.**

Utilice Application Signals en la consola de CloudWatch para monitorear y solucionar problemas del estado operativo de sus aplicaciones:

- Monitoree los servicios de sus aplicaciones: como parte del monitoreo operativo diario, utilice la página Servicios para ver un resumen de todos los servicios. Consulte los servicios con la mayor tasa de errores o latencia y compruebe qué servicios tienen indicadores de nivel de servicio (SLI) que no funcionan de forma correcta. Seleccione un servicio para abrir la página Detalles del servicio y ver las métricas detalladas, las operaciones del servicio, los valores controlados de Synthetics y las solicitudes de los clientes. Esto le permite solucionar problemas e identificar la causa raíz de los problemas operativos.

- Inspeccione la topología de su aplicación: utilice el Mapa de servicio para comprender y monitorear la topología de su aplicación a lo largo del tiempo, incluidas las relaciones entre los clientes, los valores controlados de Synthetics, los servicios y las dependencias. Vea al instante el estado del indicador de nivel de servicio (SLI) y consulte las métricas clave, como el volumen de llamadas, la tasa de errores y la latencia. Para obtener información más detallada, desplácese a la página Detalles del servicio.

Explore un ejemplo de escenario que demuestra cómo se pueden utilizar estas páginas para solucionar con rapidez un problema de estado del servicio operativo, desde la detección inicial hasta la identificación de la causa raíz.

Cómo Application Signals permite el monitoreo del estado operativo

Después de activar su aplicación para Application Signals, los servicios de la aplicación, las API y sus dependencias se detectan de manera automática y se muestran en las páginas Servicios, Detalles del servicio y Mapa de servicios. Application Signals recopila información de varios orígenes para poder activar la detección de servicios y la supervisión del estado operativo:

- AWS Distro para OpenTelemetry (ADOT): como parte de la activación de Application Signals, se configura una biblioteca de autoinstrumentación Java de OpenTelemetry para emitir métricas y seguimientos recopilados por el agente de CloudWatch. Las métricas y los seguimientos se utilizan para activar la detección de servicios, operaciones, dependencias y otra información de servicio.
- Objetivos de nivel de servicio (SLO): después de crear objetivos de nivel de servicio para sus servicios, las páginas Servicios, Detalles del servicio y Mapa de servicios muestran el estado del indicador de nivel de servicio (SLI). Los SLI pueden monitorear la latencia, la disponibilidad y otras métricas operativas.
- Valores controlados de CloudWatch: cuando configura el rastreo de rayos X en los valores controlados, las llamadas a sus servicios desde los scripts de los valores controlados se asocian a su servicio y se muestran en la página de detalles del servicio.
- Monitoreo de usuarios reales (RUM) de CloudWatch: cuando el rastreo de rayos X está activado en el cliente web RUM de CloudWatch RUM, las solicitudes a los servicios se asocian de manera automática y se muestran en la página de detalles del servicio.
- AWS Service Catalog AppRegistry: Application Signals descubre automáticamente los recursos de AWS de su cuenta y le permite agruparlos en aplicaciones lógicas creadas en AppRegistry. El nombre de la aplicación que aparece en la página de servicios se basa en el recurso informático subyacente en el que se ejecutan los servicios.

> **Note**
>
> Application Signals muestra los servicios y operaciones en función de las métricas y los rastros emitidos en el filtro de tiempo actual que haya elegido. (De forma predeterminada, son las últimas tres horas). Si no hay actividad en el filtro de tiempo actual para un servicio, operación, dependencia, valor controlado de Synthetics o página de cliente, no se mostrará. Actualmente, se pueden mostrar hasta 1000 servicios. La detección de los servicios y su topología puede demorar hasta 10 minutos. La evaluación del estado del indicador de nivel de servicio (SLI) puede demorar hasta 15 minutos.

Visualización de la actividad general del servicio y el estado operativo en la página del Servicio

> **Application Signals se encuentra en versión preliminar para Amazon CloudWatch y está sujeto a cambios.**

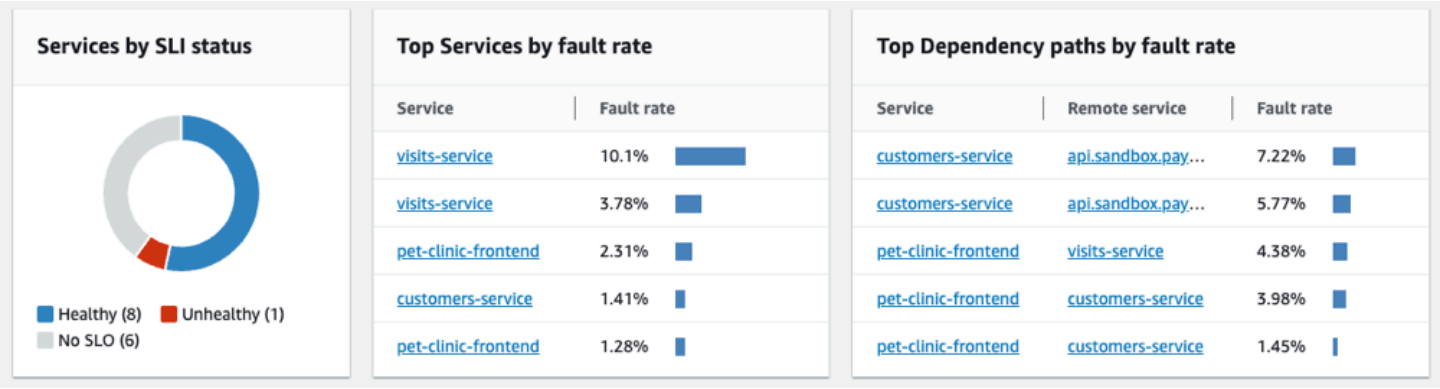
Utilice la página Servicios para ver una lista de los servicios que están habilitados para Application Signals. También puede consultar las métricas operativas y ver rápidamente qué servicios tienen indicadores de nivel de servicio (SLI) incorrectos. Desplácese hacia abajo para detectar anomalías de rendimiento a medida que identifica la causa raíz de los problemas operativos. Para ver esta página, abra la consola de CloudWatch y elija Servicios en la sección Application Signals del panel de navegación izquierdo.

Exploración de las métricas de estado operativo de sus servicios

La parte superior de la página Servicios incluye un gráfico general del estado operativo del servicio y varias tablas que muestran los principales servicios y dependencias de los servicios por tasa de errores. El gráfico de servicios de la izquierda muestra un desglose del número de servicios que tienen indicadores de nivel de servicio (SLI) correctos e incorrectos durante el filtro de tiempo actual a nivel de página. Los SLI pueden monitorear la latencia, la disponibilidad y otras métricas operativas.

Las dos tablas situadas junto al gráfico muestran una lista de los principales servicios por tasa de errores. Elija cualquier nombre de servicio en cualquiera de las tablas para abrir una página de

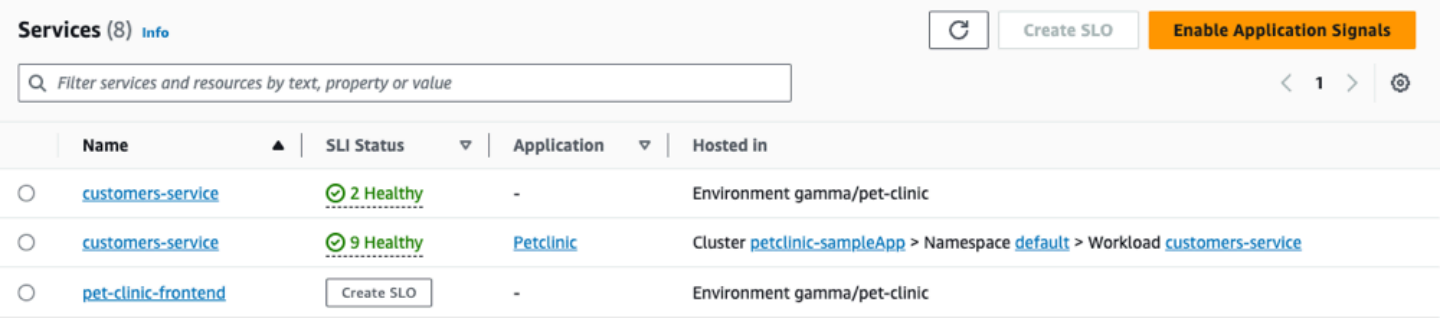
detalles del servicio y ver los detalles de la operación del servicio. Elija una ruta de dependencia para abrir la página de detalles y ver los detalles de dependencia del servicio. Ambas tablas muestran información de las últimas tres horas, incluso si se selecciona un filtro de períodos de tiempo más largos en la parte superior derecha de la página.



Supervisión del estado operativo con la tabla Servicios

La tabla Servicios muestra una lista de los servicios que se han activado para Application Signals. Elija Habilitar Application Signals para abrir una página de configuración y empezar a configurar los servicios. Para obtener más información, consulte Habilitar Application Signals.

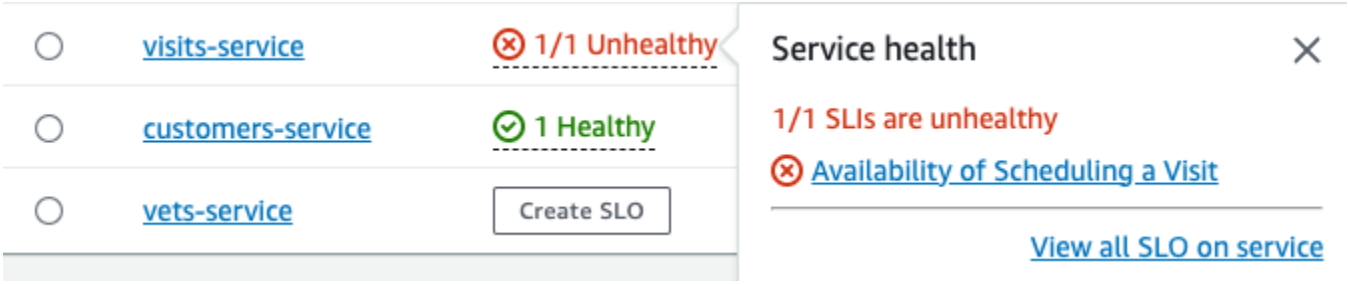
Para que le resulte más fácil encontrar lo que busca, filtre la tabla Servicios al seleccionar una o más propiedades del cuadro de texto del filtro. Al elegir cada propiedad, se lo guiará por los criterios del filtro. Verá el filtro completo debajo del cuadro de texto del filtro. Elija Borrar filtros en cualquier momento para eliminar el filtro de la tabla.



Elija el nombre de cualquier servicio de la tabla para ver una página de detalles del servicio que contiene métricas de nivel de servicio, operaciones y detalles adicionales. Si ha asociado el recurso informático subyacente del servicio a una aplicación de AppRegistry o a la tarjeta Aplicaciones de la página de inicio de la AWS Management Console, elija el nombre de la aplicación para mostrar los detalles de la aplicación en la página de la consola de myApplications. En el caso de los servicios alojados en Amazon EKS, elija cualquier enlace de la columna Alojado en para ver el clúster, el

espacio de nombres o la carga de trabajo en la información de contenedores de CloudWatch. Para los servicios que se ejecutan en Amazon ECS o Amazon EC2, se muestra el valor Entorno.

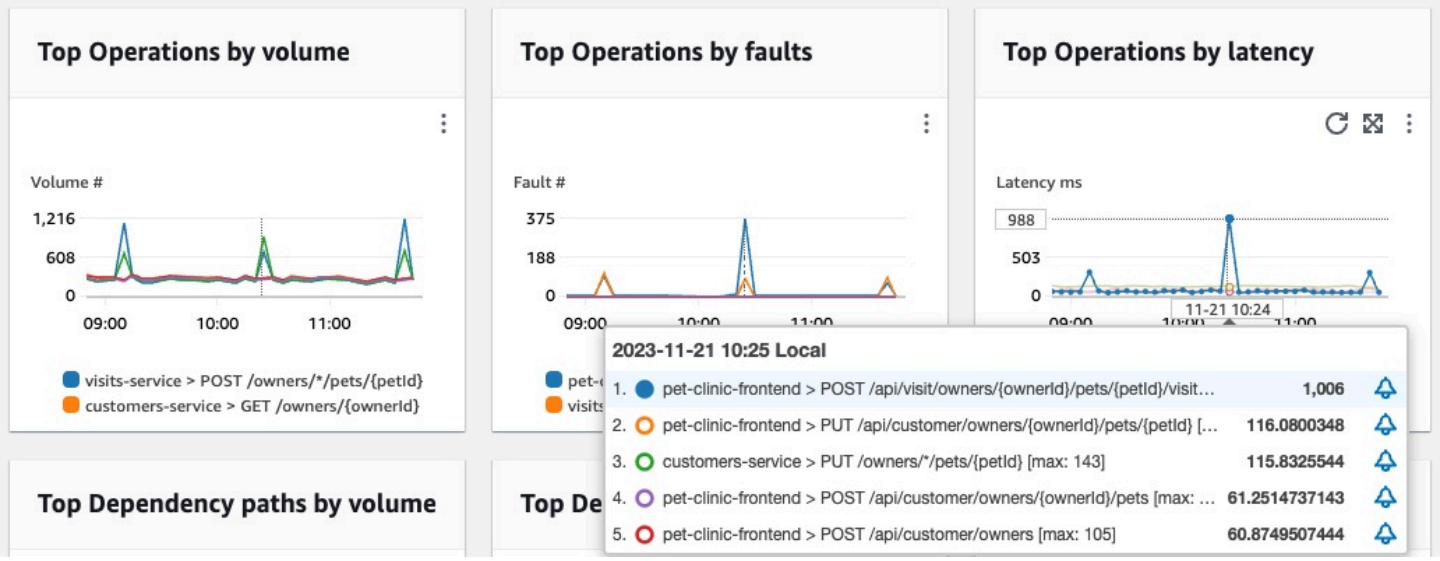
Se muestra el estado del Indicador de nivel de servicio (SLI) para cada servicio de la tabla. Elija el estado del SLI de un servicio para que aparezca una ventana emergente con un enlace a los SLI incorrectos y un enlace para ver todos los SLO del servicio.



Si no se ha creado ningún SLO para un servicio, elija el botón Crear SLO en la columna Estado del SLI. Para crear SLO adicionales para cualquier servicio, seleccione el botón de opción situado junto al nombre del servicio y, a continuación, elija Crear SLO en la parte superior derecha de la tabla. Al crear los SLO, puede ver de inmediato cuáles de sus servicios y operaciones funcionan de forma correcta y cuáles no. Para obtener más información, consulte Objetivos de nivel de servicio (SLO).

Visualización de las principales métricas de operación y dependencia

Debajo de la tabla Servicios, puede ver las principales operaciones y dependencias de todos los servicios por volumen de llamadas, errores y latencia. Este conjunto de gráficos proporciona información fundamental sobre qué operaciones o dependencias pueden no funcionar de forma correcta en todos los servicios. Seleccione cualquier punto del gráfico para ver una ventana emergente con información de serie más detallada. Coloque el cursor sobre las descripciones de las series en la parte inferior de un gráfico para ver una ventana emergente que contiene métricas detalladas de una operación o ruta de dependencia específica. Seleccione el botón del menú contextual en la esquina superior derecha de un gráfico para ver opciones adicionales, incluida la visualización de las métricas o las páginas de registros de CloudWatch.



Visualización de la actividad detallada del servicio y el estado operativo en la página de detalles del servicio

⚠ Application Signals se encuentra en versión preliminar para Amazon CloudWatch y está sujeto a cambios.

Al instrumentar la aplicación, Amazon CloudWatch Application Signals asigna todos los servicios que descubre la aplicación. Utilice la página de detalles del servicio para ver una descripción general de sus servicios, las operaciones, dependencias, canarios y solicitudes de los clientes para un solo servicio. Para ver la página de detalles del servicio, haga lo siguiente:

- Abra la consola de CloudWatch.
- Elija Servicios en la sección Application Signals del panel de navegación izquierdo.
- Elija el nombre de cualquier servicio desde Servicios, Servicios principales o de las tablas de dependencias.

La página de detalles del servicio se organiza en las siguientes pestañas:

- Descripción general: utilice esta pestaña para ver una descripción general de un solo servicio, incluida la cantidad de operaciones, las dependencias, los datos sintéticos y las páginas de clientes. La pestaña muestra las métricas clave de todo el servicio, las principales operaciones y

las dependencias. Estas métricas incluyen datos de series temporales sobre la latencia, las fallas y los errores en todas las operaciones de servicio para ese servicio.

- Operaciones de servicio: use esta pestaña para ver una lista de las operaciones a las que llama su servicio y gráficos interactivos con las métricas clave que miden el estado de cada operación. Puede seleccionar un punto de datos en un gráfico para obtener información sobre los rastros, los registros o las métricas asociadas a ese punto de datos.
- Dependencias: use esta pestaña para ver una lista de las dependencias a las que llama su servicio y una lista de métricas de esas dependencias.
- Canarios de Synthetics: use esta pestaña para ver una lista de canarios de Synthetics que simulan las llamadas de los usuarios a su servicio y las métricas de rendimiento clave para ver cómo funcionan esos canarios.
- Páginas de clientes: use esta pestaña para ver una lista de las páginas de clientes que llaman a su servicio y las métricas que miden la calidad de las interacciones de los clientes con su aplicación.

Visualización de la descripción general de su servicio

Utilice la página de descripción general del servicio para ver un resumen detallado de las métricas de todas las operaciones de servicio en una única ubicación. Compruebe el rendimiento de todas las operaciones, dependencias, páginas de clientes y canarios de Synthetics que interactúan con su aplicación. Utilice esta información como ayuda para determinar dónde concentrar los esfuerzos para identificar problemas, solucionar errores y encontrar oportunidades de optimización.

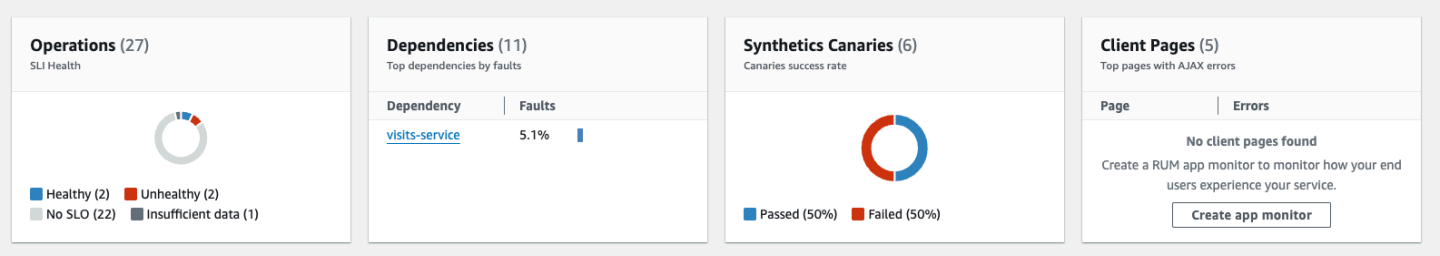
Elija cualquier enlace en Detalles del servicio para ver la información relacionada con un servicio específico. Por ejemplo, en el caso de los servicios alojados en Amazon EKS, la página de detalles del servicio muestra información sobre el clúster, el espacio de nombres y la carga de trabajo. Para los servicios alojados en Amazon ECS o Amazon EC2, la página de detalles del servicio muestra el valor del entorno.

En Servicios, la pestaña Descripción general muestra un resumen de lo siguiente:

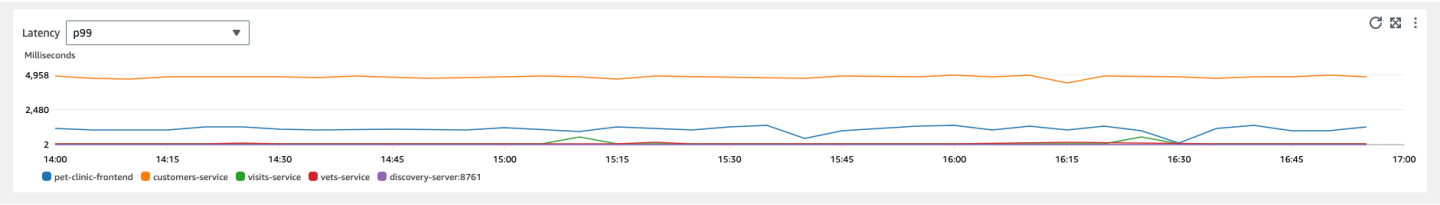
- Operaciones: utilice esta pestaña para ver el estado de las operaciones de servicio. El estado está determinado por los indicadores de nivel de servicio (SLI), que se definen como parte de un objetivo de nivel de servicio (SLO).
- Dependencias: utilice esta tabla para ver las principales dependencias de los servicios a los que llama su aplicación, ordenadas por tasa de errores.
- Canarios de Synthetics: utilice esta pestaña para ver el resultado de las llamadas simuladas a puntos de conexión o API asociadas a su servicio y el número de canarios fallidos.

- Páginas de clientes: utilice esta pestaña para ver las páginas principales a las que llaman los clientes que tienen errores asíncronos de JavaScript y XML (AJAX).

La siguiente ilustración muestra una descripción general de los servicios:



La pestaña Descripción general también muestra un gráfico de las dependencias con la latencia más alta en todos los servicios. Utilice las métricas de latencia p99, p90 y p50 para evaluar rápidamente qué dependencias contribuyen a la latencia total del servicio, de la siguiente manera:

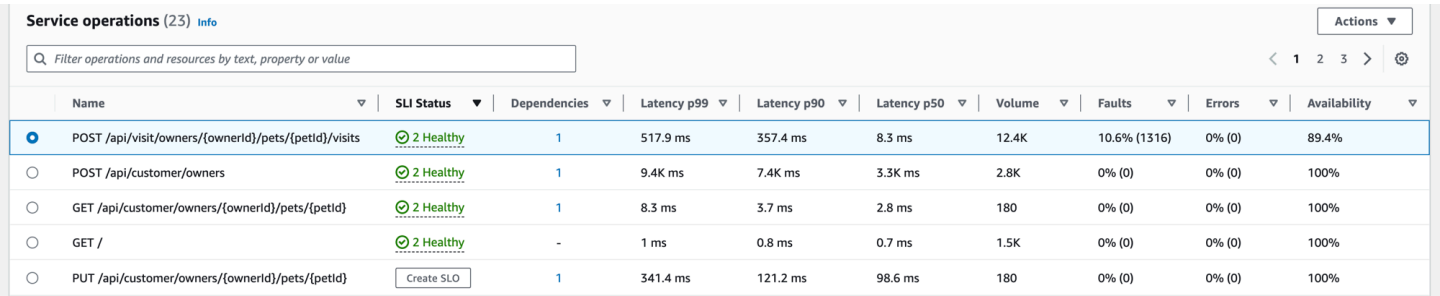


Por ejemplo, el gráfico anterior muestra que el 99 % de las solicitudes realizadas a la dependencia customer-service se completaron en aproximadamente 4950 milisegundos. Las demás dependencias tardaron menos tiempo.

Los gráficos que muestran las cuatro operaciones de servicio principales por latencia muestran el volumen de solicitudes, la disponibilidad, la tasa de fallos y la tasa de errores de esos servicios, como se muestra en la siguiente imagen:

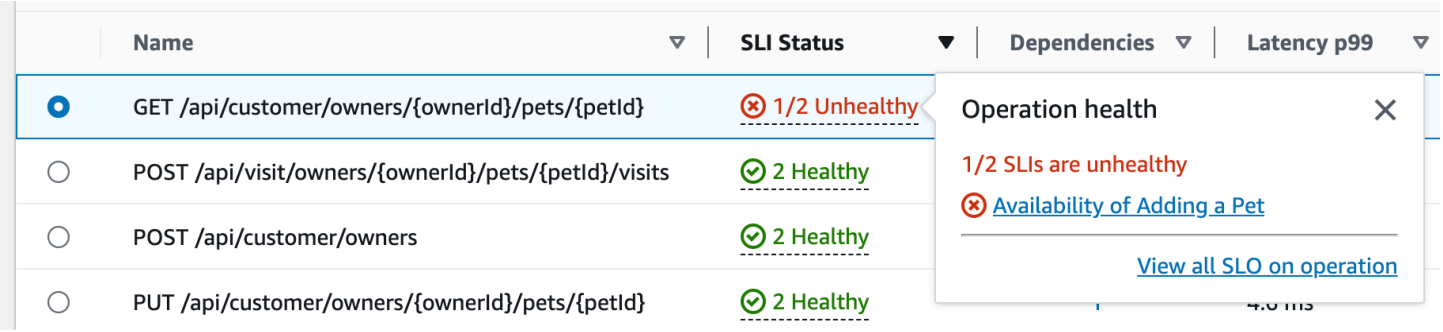
Visualización de las operaciones de servicio

Al instrumentar la aplicación, Application Signals descubre todas las operaciones de servicio a las que llama la aplicación. Utilice la pestaña Operaciones del servicio para ver una tabla que contiene las operaciones del servicio y un conjunto de métricas que miden el rendimiento de una operación seleccionada. Estas métricas incluyen el estado del indicador de nivel de servicio (SLI), el número de dependencias, la latencia, el volumen, los fallos, los errores y la disponibilidad, como se muestra en la siguiente imagen:



Filtre la tabla para que le resulte más fácil encontrar una operación de servicio al seleccionar una o más propiedades del cuadro de texto del filtro. Al elegir cada propiedad, se lo guiará por los criterios de filtro y verá el filtro completo debajo del cuadro de texto del filtro. Elija Borrar filtros en cualquier momento para eliminar el filtro de la tabla.

Elija el estado del SLI para una operación para mostrar una ventana emergente que contiene un vínculo a cualquier SLI en mal estado y un vínculo para ver todos los SLO de la operación, como se muestra en la siguiente tabla:



La tabla de operaciones del servicio muestra el estado del SLI, el número de SLI en buen estado o en mal estado y el número total de SLO para cada operación.

Utilice los SLI para supervisar la latencia, la disponibilidad y otras métricas operativas que miden el estado operativo de un servicio. Utilice un SLO para comprobar el rendimiento y el estado de sus servicios y operaciones.

Para crear un SLO, realice el siguiente procedimiento:

- Si una operación no tiene ningún SLO, elija el botón Crear SLO en la columna Estado del SLI.
- Si una operación ya tiene un SLO, haga lo siguiente:
  - Seleccione el botón de opción situado junto al nombre de la operación.
  - Seleccione Crear SLO en la flecha desplegable Acciones situada en la parte superior derecha de la tabla.

Para obtener más información, consulte Objetivos de nivel de servicio (SLO).

La columna Dependencias muestra el número de dependencias a las que llama esta operación. Elija este número para abrir la pestaña Dependencias filtrada según la operación seleccionada.

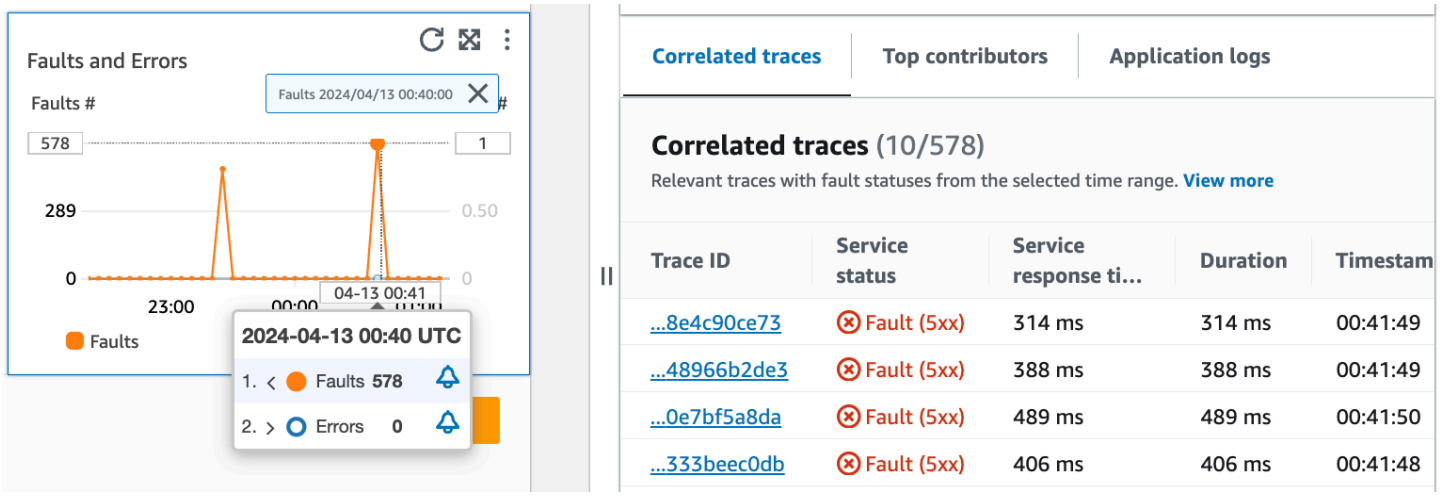
Visualización de métricas de operaciones de servicio, seguimientos correlacionados y registros de aplicaciones

Application Signals correlaciona las métricas de operación de servicio con los seguimientos de AWS X-Ray, CloudWatch Container Insights y los registros de las aplicaciones. Utilice estas métricas para solucionar problemas de estado operativo. Para ver las métricas como información gráfica, haga lo siguiente:

1. Elija una opción de servicio en la tabla Operaciones de servicio para ver un conjunto de gráficos para la operación seleccionada encima de la tabla con métricas de volumen y disponibilidad, latencia y fallas y errores.

2. Coloque el cursor sobre un punto en un gráfico para ver más información.
3. Seleccione un punto para abrir un panel de diagnóstico que muestra seguimientos, métricas y registros de aplicaciones correlacionados para el punto seleccionado en el gráfico.

La siguiente imagen muestra la información sobre herramientas que aparece después de pasar el ratón sobre un punto del gráfico y el panel de diagnóstico que aparece al hacer clic en un punto. La información sobre herramientas contiene información sobre el punto de datos asociado en el gráfico de fallas y errores. El panel contiene los seguimientos correlacionados, los colaboradores principales y los registros de aplicaciones asociados al punto seleccionado.



Seguimientos correlacionados

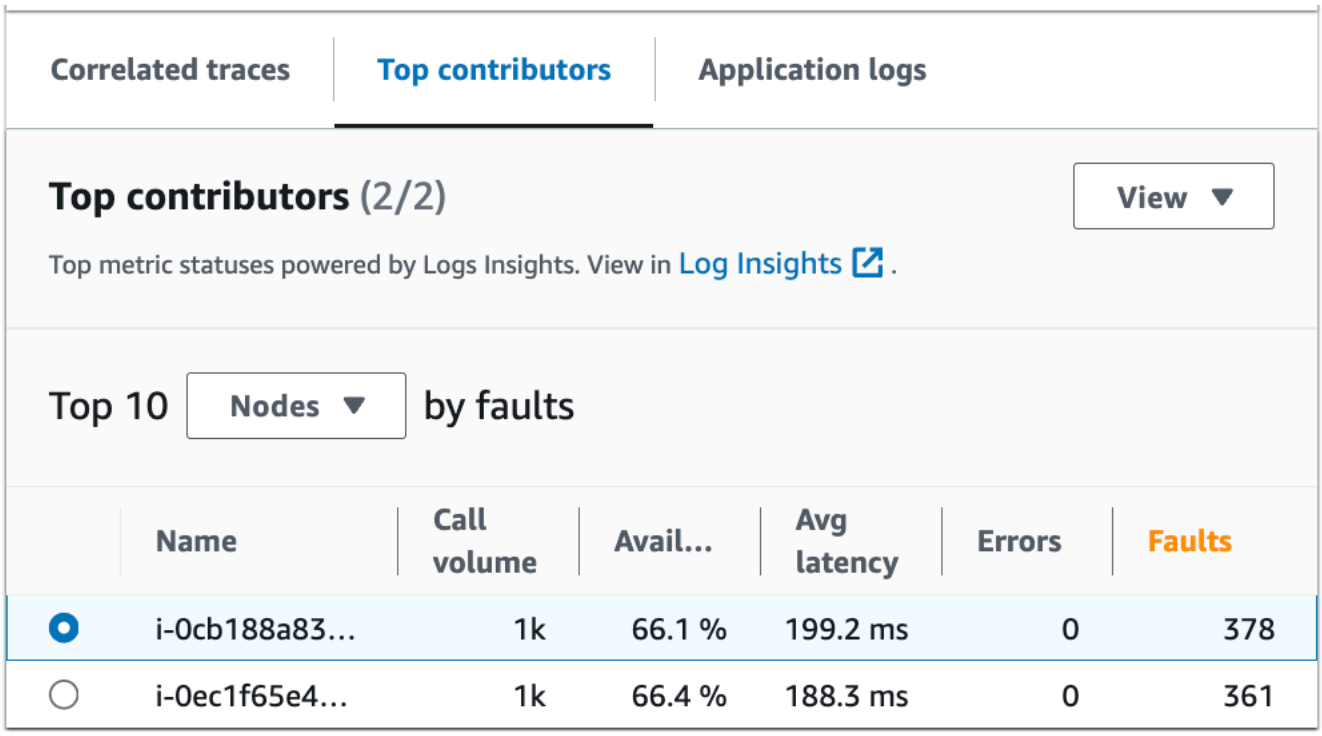
Observe los seguimientos relacionados para comprender un problema subyacente con un seguimiento. Puede comprobar si los seguimientos correlacionados o cualquier nodo de servicio asociado a ellos se comportan de forma similar. Para examinar los seguimientos correlacionados, elija un ID de seguimiento de la tabla Seguimientos correlacionados para abrir la página Detalles del seguimiento de X-Ray para el seguimiento elegido. La página de detalles del seguimiento contiene un mapa de nodos de servicio asociados con el seguimiento seleccionado y una línea de tiempo de los segmentos del seguimiento.

Colaboradores principales

Consulte los principales colaboradores para encontrar las principales fuentes de entrada a una métrica. Agrupe a los colaboradores por diferentes componentes para buscar similitudes dentro del grupo y comprender en qué se diferencia el comportamiento de seguimiento entre ellos.

En la pestaña Colaboradores principales se muestran las métricas del volumen de llamadas, la disponibilidad, la latencia media, los errores y las fallas de cada grupo. La siguiente imagen de

ejemplo muestra los principales colaboradores a un conjunto de métricas para una aplicación implementada en una plataforma Amazon EKS:



Los principales colaboradores contienen las siguientes métricas:

- Volumen de llamadas: utilice el volumen de llamadas para conocer el número de solicitudes por intervalo de tiempo de un grupo.
- Disponibilidad: utilice la disponibilidad para ver el porcentaje de tiempo que no se detectó ningún error en un grupo.
- Latencia media: utilice la latencia para comprobar el tiempo medio durante el que se enviaron las solicitudes de un grupo durante un intervalo de tiempo que depende de cuánto tiempo hace que se hicieron las solicitudes que está investigando. Las solicitudes que se realizaron menos de 15 días antes se evalúan en intervalos de 1 minuto. Las solicitudes que se realizaron entre 15 y 30 días antes, inclusive, se evalúan en intervalos de 5 minutos. Por ejemplo, si está investigando las solicitudes que provocaron un error hace 15 días, la métrica del volumen de llamadas es igual al número de solicitudes por intervalo de 5 minutos.
- Errores: el número de errores por grupo medidos durante un intervalo de tiempo.
- Fallos: el número de fallos por grupo durante un intervalo de tiempo.

Colaboradores principales que utilizan Amazon EKS o Kubernetes

Utilice información acerca de los colaboradores principales para aplicaciones implementadas en Amazon EKS o Kubernetes para ver las métricas de estado operativo agrupadas por Nodo, Pod y PodTemplateHash. Se aplican las siguientes definiciones:

- Un pod es un grupo de uno o más contenedores de Docker que comparten almacenamiento y recursos. Un pod es la unidad más pequeña que se puede implementar en una plataforma de Kubernetes. Agrupe por pods para comprobar si los errores están relacionados con limitaciones específicas del pod.
- Un nodo es un servidor que ejecuta pods. Agrupe por nodos para comprobar si los errores están relacionados con limitaciones específicas del nodo.
- El hash de una plantilla de pod se utiliza para buscar una versión concreta de una implementación. Agrupe por hash de plantilla de pod para comprobar si los errores están relacionados con una implementación en particular.

Principales colaboradores que utilizan Amazon EC2

Utilice la información sobre los principales colaboradores para aplicaciones implementadas en Amazon EKS para ver las métricas de estado operativo agrupadas por ID de instancia y grupo de escalado automático. Se aplican las siguientes definiciones:

- Un ID de instancia es un identificador único de la instancia de Amazon EC2 que ejecuta su servicio. Agrupe por ID de instancia para comprobar si los errores están relacionados con una instancia de Amazon EC2 específica.
- Un grupo de escalado automático es un conjunto de instancias de Amazon EC2 que le permiten escalar o reducir verticalmente los recursos que necesita para atender las solicitudes de sus aplicaciones. Agrupe por grupo de escalado automático si quiere comprobar si el alcance de los errores se limita a las instancias del grupo.

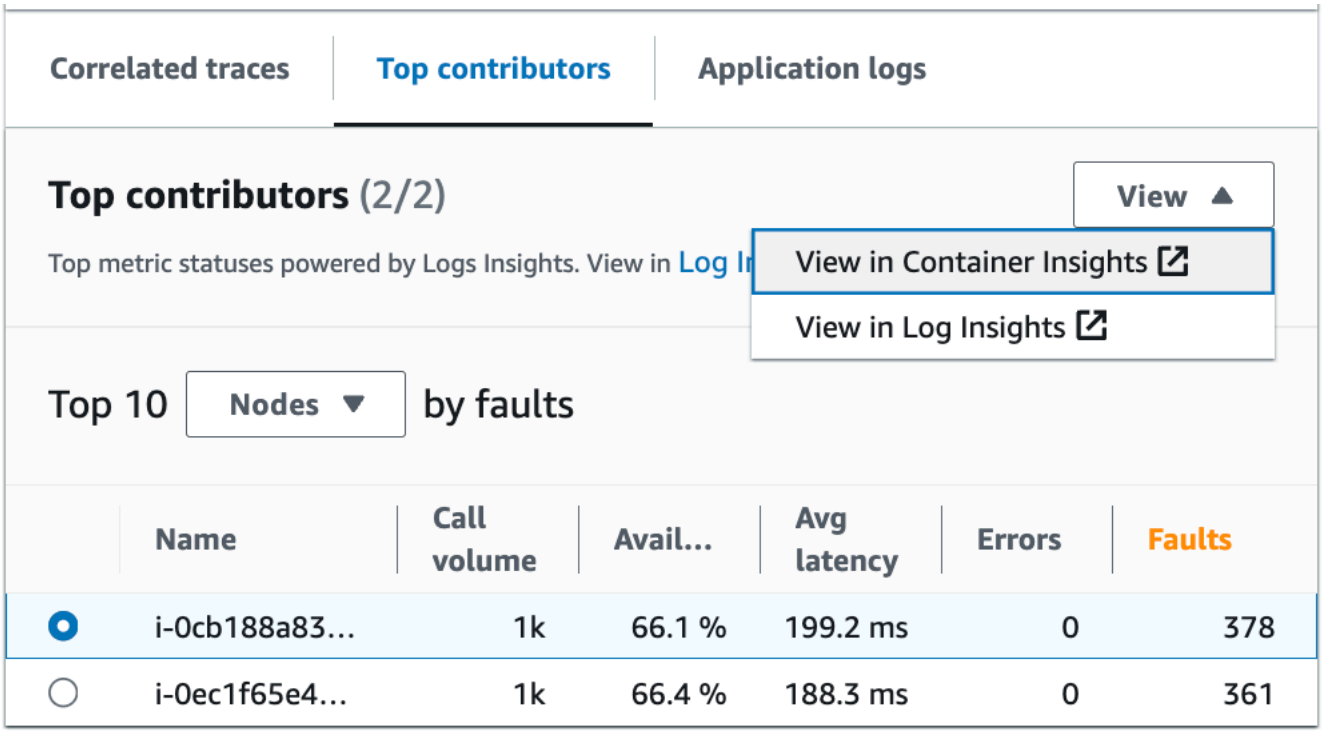
Principales colaboradores que utilizan una plataforma personalizada

Utilice la información sobre los principales colaboradores para aplicaciones implementadas con instrumentación personalizada para ver las métricas del estado operativo agrupadas por nombre de host. Se aplican las siguientes definiciones:

- Un nombre de host identifica un dispositivo, como un punto de conexión o una instancia de Amazon EC2, que está conectado a una red. Agrupe por nombre de host para comprobar si los errores están relacionados con un dispositivo físico o virtual específico.

Visualización de los principales colaboradores en Log Insights y Container Insights

Vea y modifique la consulta automática que generó las métricas para sus principales colaboradores en Información de registros. Vea las métricas de rendimiento de la infraestructura por grupos específicos, como pods o nodos, en Información de contenedores. Puede ordenar los clústeres, los nodos o las cargas de trabajo por consumo de recursos e identificar rápidamente las anomalías o mitigar los riesgos de forma proactiva antes de que la experiencia del usuario final se vea afectada. A continuación, se muestra una imagen que muestra cómo seleccionar estas opciones:



En Información de contenedores, puede ver las métricas de su contenedor de Amazon EKS o Amazon ECS que son específicas de la agrupación de sus principales colaboradores. Por ejemplo, si ha agrupado un contenedor de EKS por grupo para generar los principales colaboradores, la información sobre el contenedor mostrará las métricas y estadísticas filtradas para su pod.

En Información de registros, puede modificar la consulta que generó las métricas en Colaboradores principales siguiendo estos pasos:

1. Seleccione Ver en Información de registros. La página Información de registros que se abre contiene una consulta que se genera automáticamente y contiene la siguiente información:
   - El nombre del grupo de clústeres de registros.
   - La operación que estaba investigando con CloudWatch.

- El agregado de la métrica de estado operativo con la que interactuó en el gráfico.

Los resultados del registro se filtran automáticamente para mostrar los datos de los últimos cinco minutos antes de seleccionar el punto de datos en el gráfico de servicio.

2. Para editar la consulta, sustituya el texto generado por sus cambios. También puede usar el Generador de consultas como ayuda para generar una consulta nueva o actualizar la consulta existente.

Registros de aplicaciones

Utilice la consulta de la pestaña Registros de aplicaciones para generar información registrada para su grupo de registros y servicio actuales e inserte una marca de tiempo. Un grupo de registro es un grupo de flujos de registro que puede definir al configurar la aplicación.

Utilice un grupo de registros para organizar los registros con características similares, incluidas las siguientes:

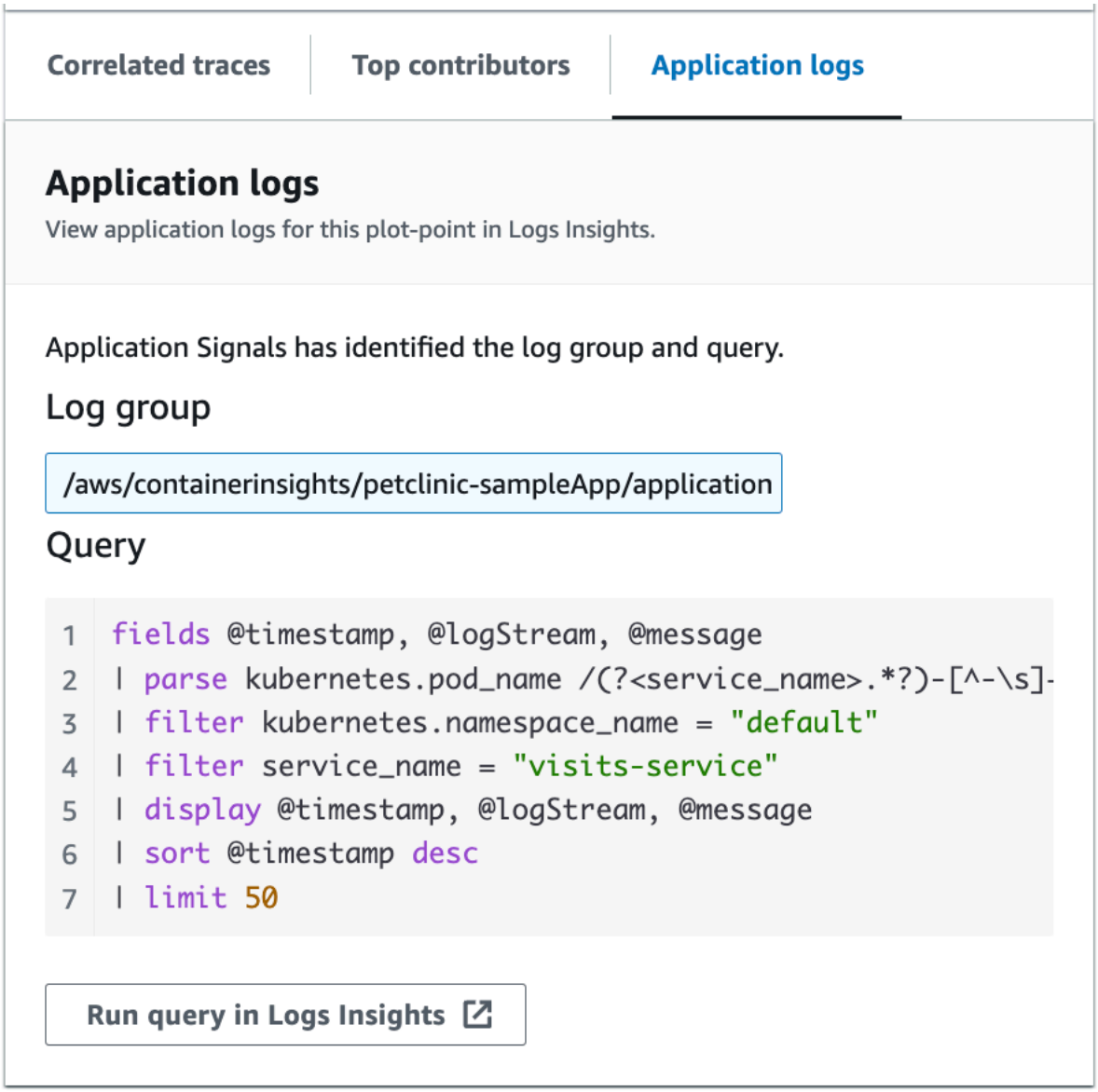
- Capture los registros de una organización, fuente o función específicas.
- Capture los registros a los que accede un usuario en particular.
- Capture registros para un periodo de tiempo específico.

Utilice estos flujos de registro para realizar un seguimiento de grupos o períodos de tiempo específicos. También puede configurar reglas de supervisión, alarmas y notificaciones para estos grupos de registros. Para obtener información general acerca de los grupos de registros, consulte Working with log groups and log streams.

La consulta de registros de la aplicación devuelve los registros, los patrones de texto recurrentes y las visualizaciones gráficas de sus grupos de registros.

Para ejecutar la consulta, seleccione Ejecutar consulta en Información de registros para ejecutar la consulta generada automáticamente o modificarla. Para editar la consulta, sustituya el texto generado automáticamente por sus cambios. También puede usar el Generador de consultas como ayuda para generar una consulta nueva o actualizar la consulta existente.

La siguiente imagen muestra la consulta de ejemplo que se genera automáticamente en función del punto seleccionado en el gráfico de operaciones del servicio:



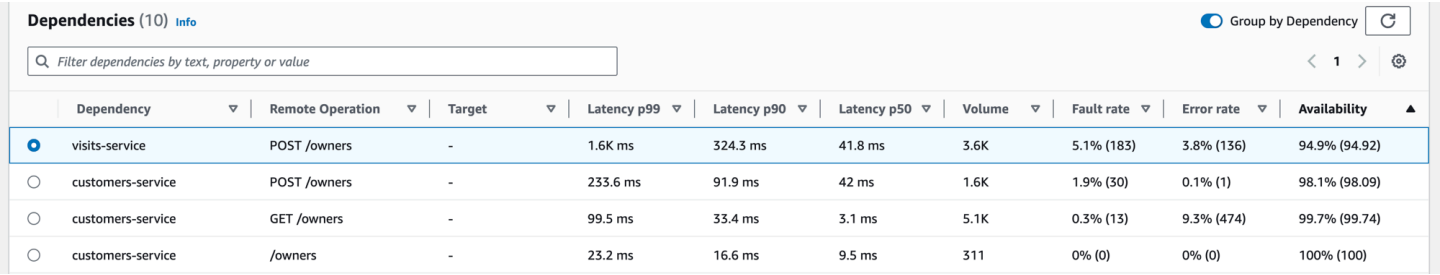
En la imagen anterior, CloudWatch detectó automáticamente el grupo de registro asociado al punto seleccionado y lo incluyó en una consulta generada.

Visualización de las dependencias de servicio

Elija la pestaña Dependencias para ver la tabla de Dependencias y un conjunto de métricas para las dependencias de todas las operaciones del servicio o de una sola operación. La tabla contiene una lista de las dependencias detectadas por Application Signals, incluidas las métricas de latencia, volumen de llamadas, tasa de fallos, tasa de errores y disponibilidad.

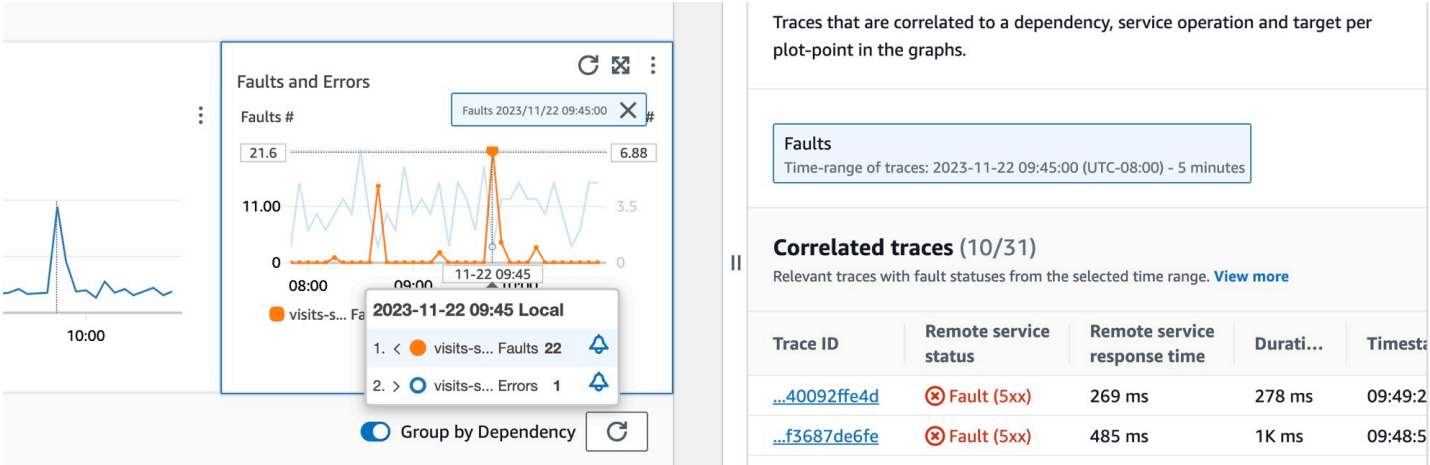
En la parte superior de la página, elija una operación de la lista desplegable para ver sus dependencias o elija Todo para ver las dependencias de todas las operaciones.

Filtre la tabla para que le resulte más fácil encontrar lo que busca, al seleccionar una o más propiedades del cuadro de texto del filtro. Al elegir cada propiedad, se lo guiará por los criterios de filtro y verá el filtro completo debajo del cuadro de texto del filtro. Elija Borrar filtros en cualquier momento para eliminar el filtro de la tabla. Seleccione Agrupar por dependencia en la parte superior derecha de la tabla para agrupar las dependencias por nombre de servicio y operación. Cuando la agrupación esté activada, expanda o contraiga un grupo de dependencias con el icono + situado junto al nombre de la dependencia.



La columna Dependencia muestra el nombre del servicio de dependencia, mientras que la columna Operación remota muestra el nombre de la operación del servicio. Al llamar a los servicios de AWS, la columna Destino muestra el recurso de AWS, como una tabla de DynamoDB o una cola de Amazon SNS.

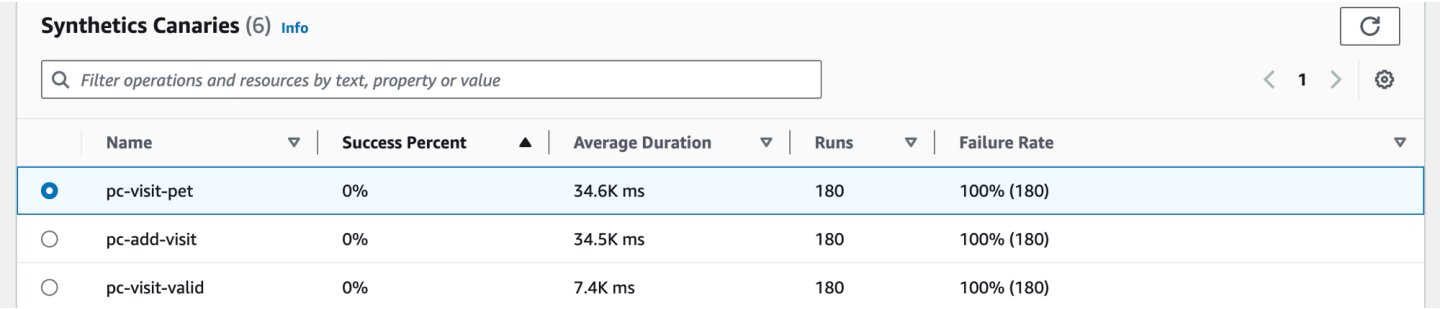
Para seleccionar una dependencia, seleccione la opción situada junto a una dependencia en la tabla de Dependencias. Esto muestra un conjunto de gráficos que muestran métricas detalladas de volumen de llamadas, disponibilidad, fallas y errores. Coloque el cursor sobre un punto en un gráfico para ver una ventana emergente con más información. Seleccione un punto de un gráfico para abrir un panel de diagnóstico que muestre los seguimientos correlacionados del punto seleccionado en el gráfico. Elija un ID de seguimiento de la tabla Seguimientos correlacionados para abrir la página Detalles del seguimiento de X-Ray para el seguimiento elegido.



Visualización de los valores controlados de Synthetics

Seleccione la pestaña Valores controlados de Synthetics para ver la tabla Valores controlados de Synthetics y un conjunto de métricas para cada valor controlado de la tabla. La tabla incluye métricas del porcentaje de éxito, la duración promedio, las ejecuciones y la tasa de fallos. Solo se muestran los valores controlados que están habilitados para el seguimiento de AWS X-Ray.

Utilice el cuadro de texto del filtro de la tabla de canarios de Synthetics para encontrar el canario que le interese. Cada filtro que crea aparece debajo del cuadro de texto del filtro. Elija Borrar filtros en cualquier momento para eliminar el filtro de la tabla.



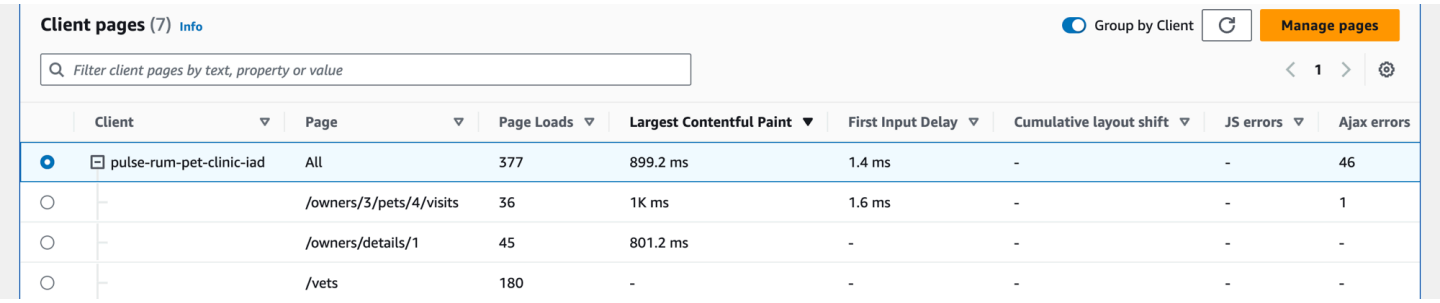
Seleccione el botón de radio situado junto al nombre del canario para ver un conjunto de pestañas que contienen métricas detalladas con gráficos, como el porcentaje de éxito, los errores y la duración. Coloque el cursor sobre un punto en un gráfico para ver una ventana emergente con más información. Seleccione un punto de un gráfico para abrir un panel de diagnóstico que muestre las ejecuciones de canario que se correlacionan con el punto seleccionado. Seleccione una ejecución de canario y elija el tiempo de ejecución para ver los artefactos de la ejecución de canario que seleccionó, incluidos los registros, los archivos HTTP (HAR), las capturas de pantalla y los pasos sugeridos para ayudarlo a solucionar problemas. Seleccione Más información para abrir la página Canarios de CloudWatch Synthetics junto a Ejecuciones de canarios.

Visualización de las páginas de sus clientes

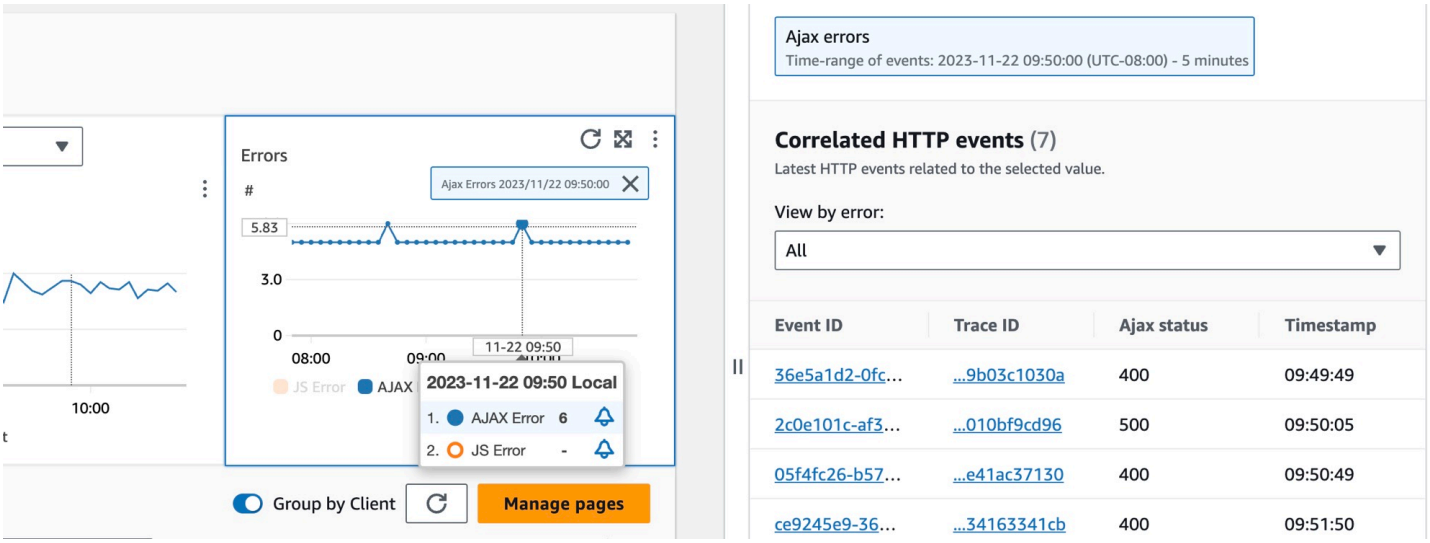
Seleccione la pestaña Páginas de clientes para ver una lista de las páginas web de clientes que llaman a su servicio. Utilice el conjunto de métricas de la página de cliente seleccionada para medir la calidad de la experiencia de su cliente al interactuar con un servicio o una aplicación. Estas métricas incluyen las cargas de página, las métricas esenciales de la web y los errores.

Para mostrar las páginas de sus clientes en la tabla, debe configurar su cliente web de CloudWatch RUM para el seguimiento de X-Ray y activar las métricas de Application Signals para las páginas de los clientes. Elija Administrar páginas para seleccionar qué páginas están habilitadas para las métricas de Application Signals.

Utilice el cuadro de texto del filtro para buscar la página del cliente o el monitor de aplicaciones que le interese debajo del cuadro de texto del filtro. Elija Borrar filtros para eliminar el filtro de la tabla. Seleccione Agrupar por cliente para agrupar las páginas de los clientes por cliente. Cuando estén agrupadas, seleccione el icono + situado junto al nombre del cliente para expandir la fila y ver todas las páginas de ese cliente.



Para seleccionar una página de cliente, seleccione la opción situada junto a una página del cliente en la tabla Páginas de clientes. Verá un conjunto de gráficos que muestran métricas detalladas. Coloque el cursor sobre un punto en un gráfico para ver una ventana emergente con más información. Seleccione un punto de un gráfico para abrir un panel de diagnóstico que muestre los eventos de navegación de rendimiento correlacionados del punto seleccionado en el gráfico. Elija un ID de evento de la lista de eventos de navegación para abrir la vista de la página de CloudWatch RUM para el evento elegido.



Note

Para ver los errores AJAX en las páginas de sus clientes, utilice la versión 1.15 de cliente web de CloudWatch RUM o posterior.
Actualmente, se pueden mostrar hasta 100 operaciones, valores controlados y páginas de clientes, y hasta 250 dependencias por servicio.

Visualización de la topología de su aplicación y supervisión del estado operativo con la asignación de servicios de CloudWatch

⚠ Application Signals se encuentra en versión preliminar para Amazon CloudWatch y está sujeto a cambios.

Note

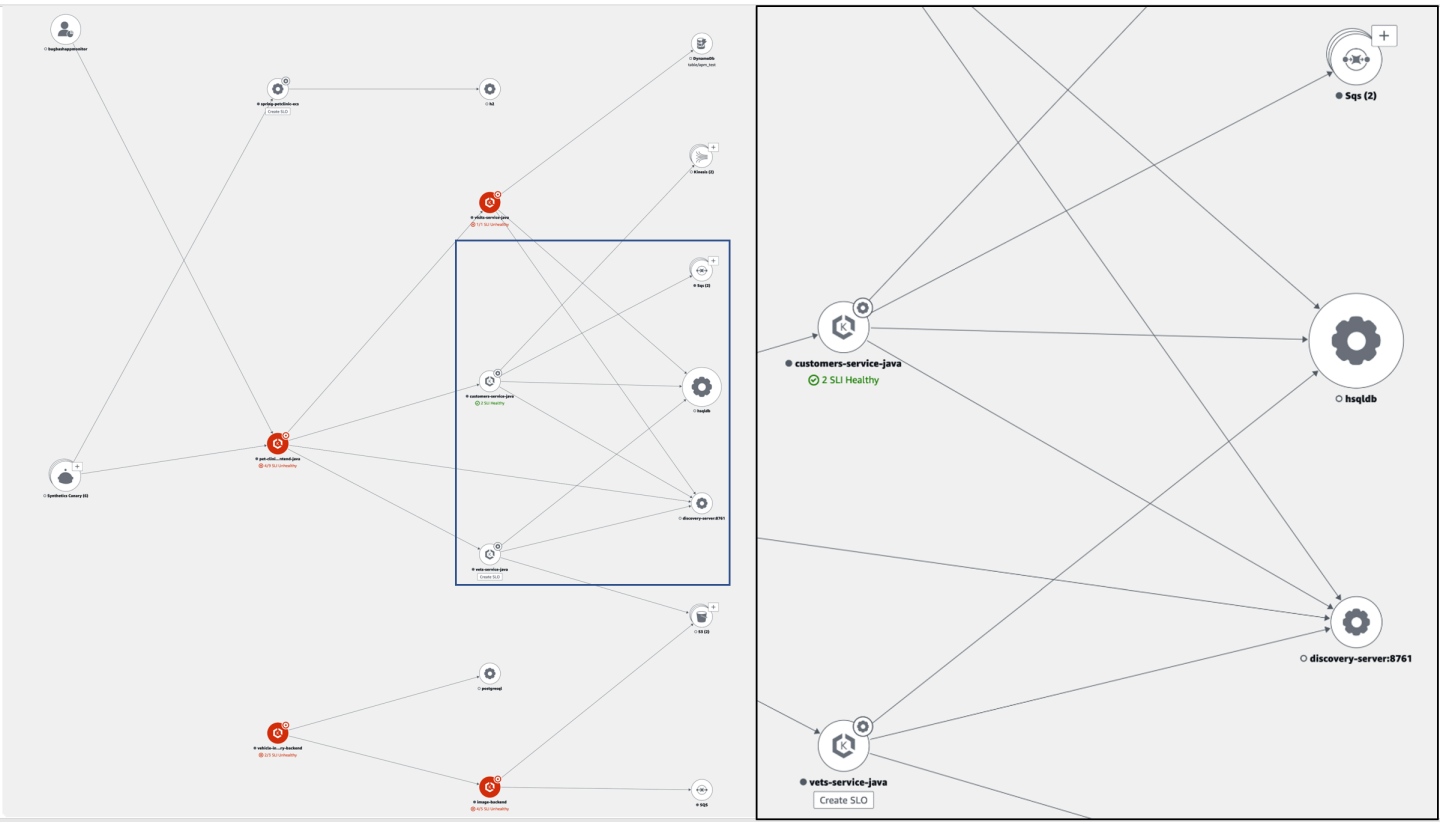
La asignación de servicios de CloudWatch reemplaza a la asignación de ServiceLens. Para ver un mapa de su aplicación basado en seguimientos de AWS X-Ray, abra el Mapa de seguimiento de X-Ray. Seleccione Mapa de servicio en la sección X-Ray en el panel de navegación izquierdo de la consola de CloudWatch.

Utilice la asignación de servicios para ver la topología de los clientes de su aplicación, los canarios sintéticos, los servicios y las dependencias y para supervisar el estado operativo. Para ver la asignación de servicios, abra la Consola de CloudWatch y elija Asignación de servicio en la sección Application Signals del panel de navegación izquierdo.

Después de habilitar la aplicación para Application Signals, utilice la asignación de servicios para facilitar el monitoreo del estado operativo de la aplicación:

- Observe las conexiones entre los nodos de clientes, valor controlado, servicio y dependencia para comprender la topología de la aplicación y el flujo de ejecución. Esto es útil sobre todo si los operadores de servicio no son su equipo de desarrollo.
- Observe qué servicios cumplen o no sus objetivos de nivel de servicio (SLO). Cuando un servicio no cumple con los SLO, puede identificar de inmediato si un servicio o una dependencia descendente podría estar contribuyendo al problema o afectando a varios servicios ascendentes.
- Seleccione un nodo individual de cliente, canario sintético, servicio o dependencia para visualizar las métricas asociadas. La página Detalles del servicio muestra información más detallada sobre las operaciones, las dependencias, los canarios sintéticos y las páginas de clientes.
- Filtre y amplíe la asignación de servicio para centrarse con más facilidad en una parte de la topología de la aplicación o para ver la asignación completa. Cree un filtro al elegir una o más propiedades del cuadro de texto del filtro. Al elegir cada propiedad, se lo guiará por los criterios de los filtros. Verá el filtro completo debajo del cuadro de texto del filtro. Elija Borrar filtros en cualquier momento para eliminar el filtro.

El siguiente ejemplo de asignación de servicios muestra los servicios con periferia que los conectan a los componentes con los que interactúan. Si se define un SLO, la asignación de servicios también muestra el estado de funcionamiento.

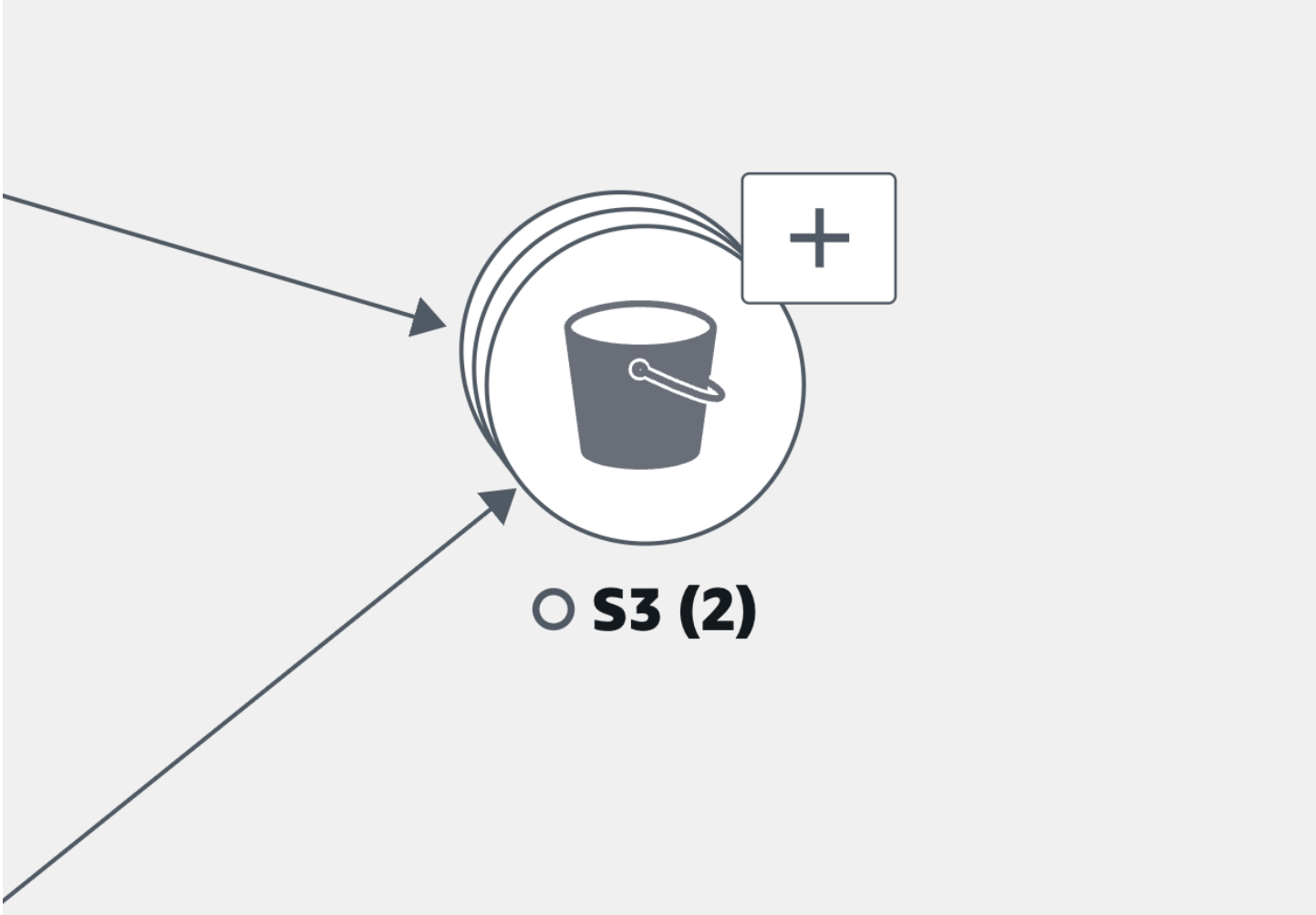


Exploración de la asignación de servicios

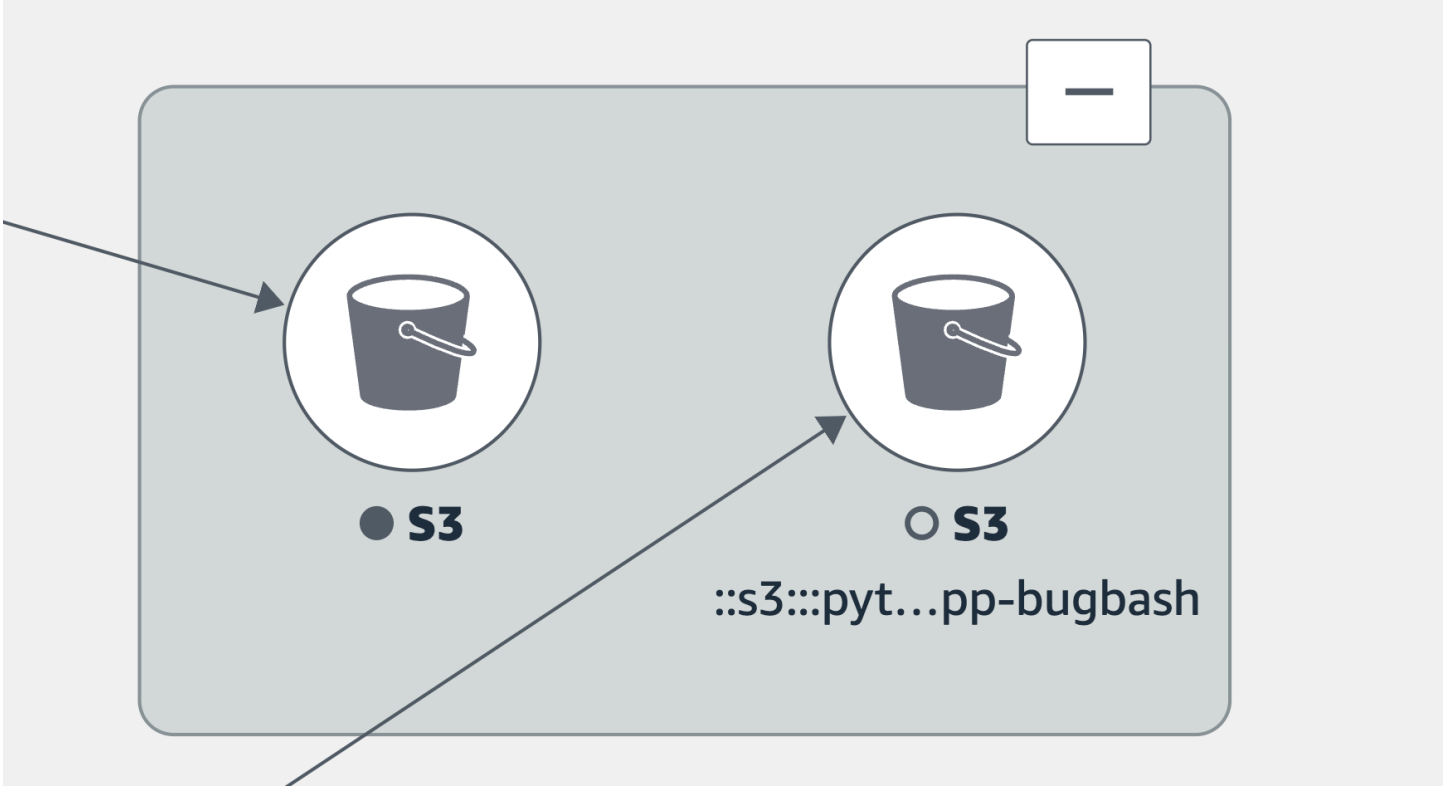
Una vez que haya habilitado la aplicación para Application Signals, la asignación de servicios muestra los nodos que representan sus servicios y dependencias.

Habilite el rastreo activo para sus clientes de CloudWatch RUM y canarios sintéticos para ver los nodos de cliente y canario en la asignación.

De forma predeterminada, los canarios, los clientes de RUM y las dependencias de servicios de AWS del mismo tipo se agrupan en un único icono expandible en la asignación de servicios. Las dependencias de servicio externas a AWS no se agrupan de forma predeterminada. Por ejemplo, en la siguiente imagen, todos los buckets de Amazon S3 están agrupados en un icono expandible:



En la imagen anterior, la etiqueta entre la agrupación de Amazon S3 y el servicio de origen muestra el número de periferias del grupo entre paréntesis debajo del icono de la dependencia. Seleccione el icono (+) para expandir el grupo y ver sus elementos individuales, como se muestra en la siguiente imagen.

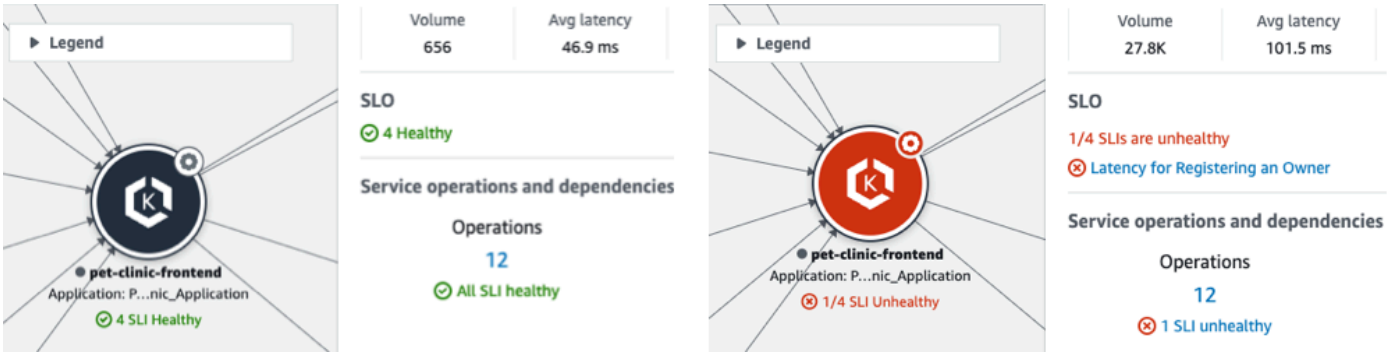


Elija la siguiente pestaña para obtener información sobre cómo explorar cada tipo de nodo y las periferias (conexiones) entre ellos.

View your application services

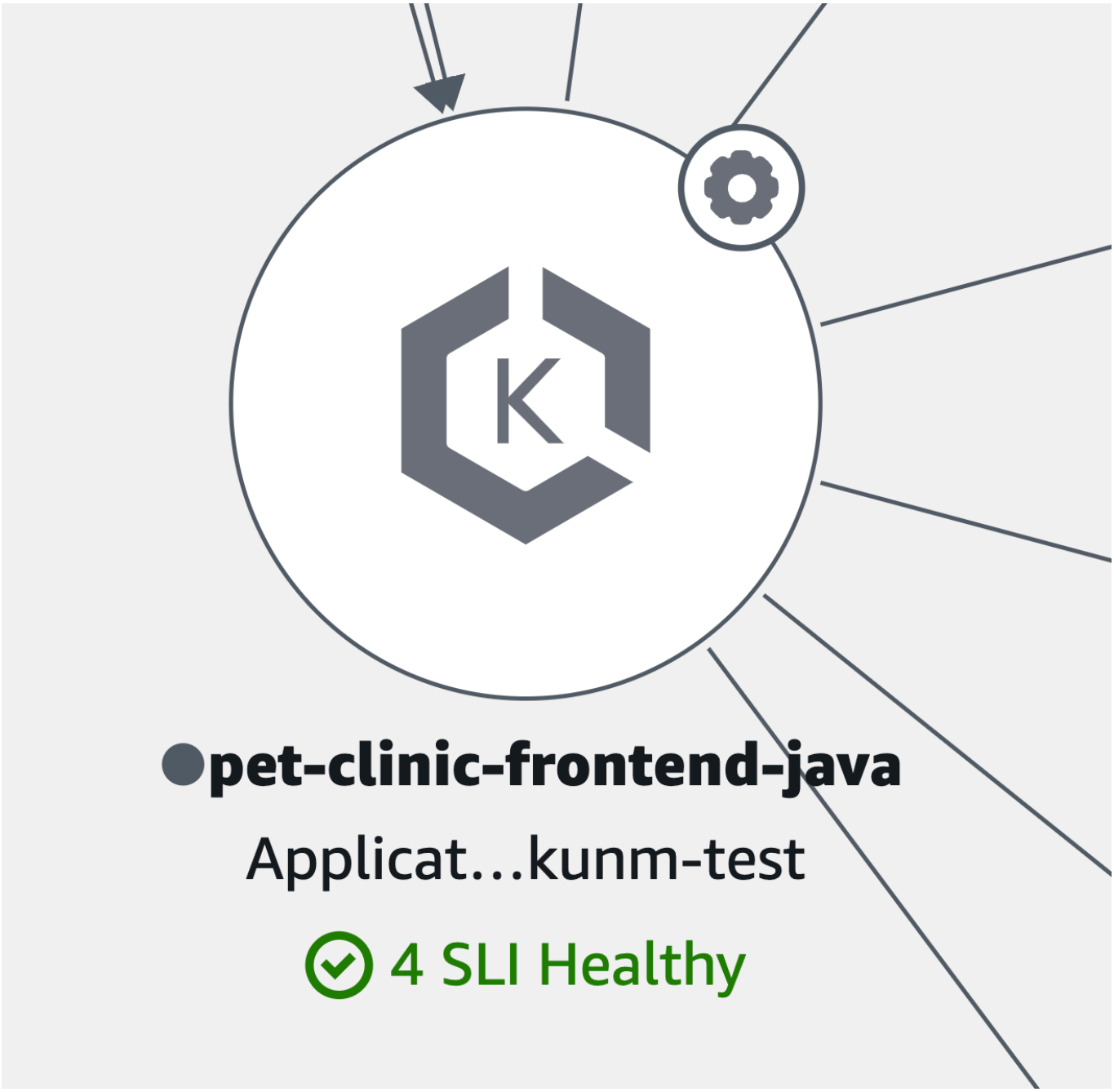
Puede ver los servicios de sus aplicaciones y el estado de sus SLO e indicadores de nivel de servicio (SLI) en la Asignación de servicio. Si no se ha creado ningún SLO para un servicio, elija el botón Crear SLO situado debajo del nodo de servicio.

La Asignación de servicios muestra todos sus servicios. También muestra los clientes y canarios que consumen el servicio y las dependencias a las que llaman sus servicios, como se muestra en la siguiente imagen:



Los siguientes iconos representan ejemplos de servicios de aplicaciones en la asignación de servicios:

- Amazon Elastic Kubernetes Service:



- Un contenedor de Kubernetes:

- Amazon Elastic Compute Cloud (Amazon EC2):

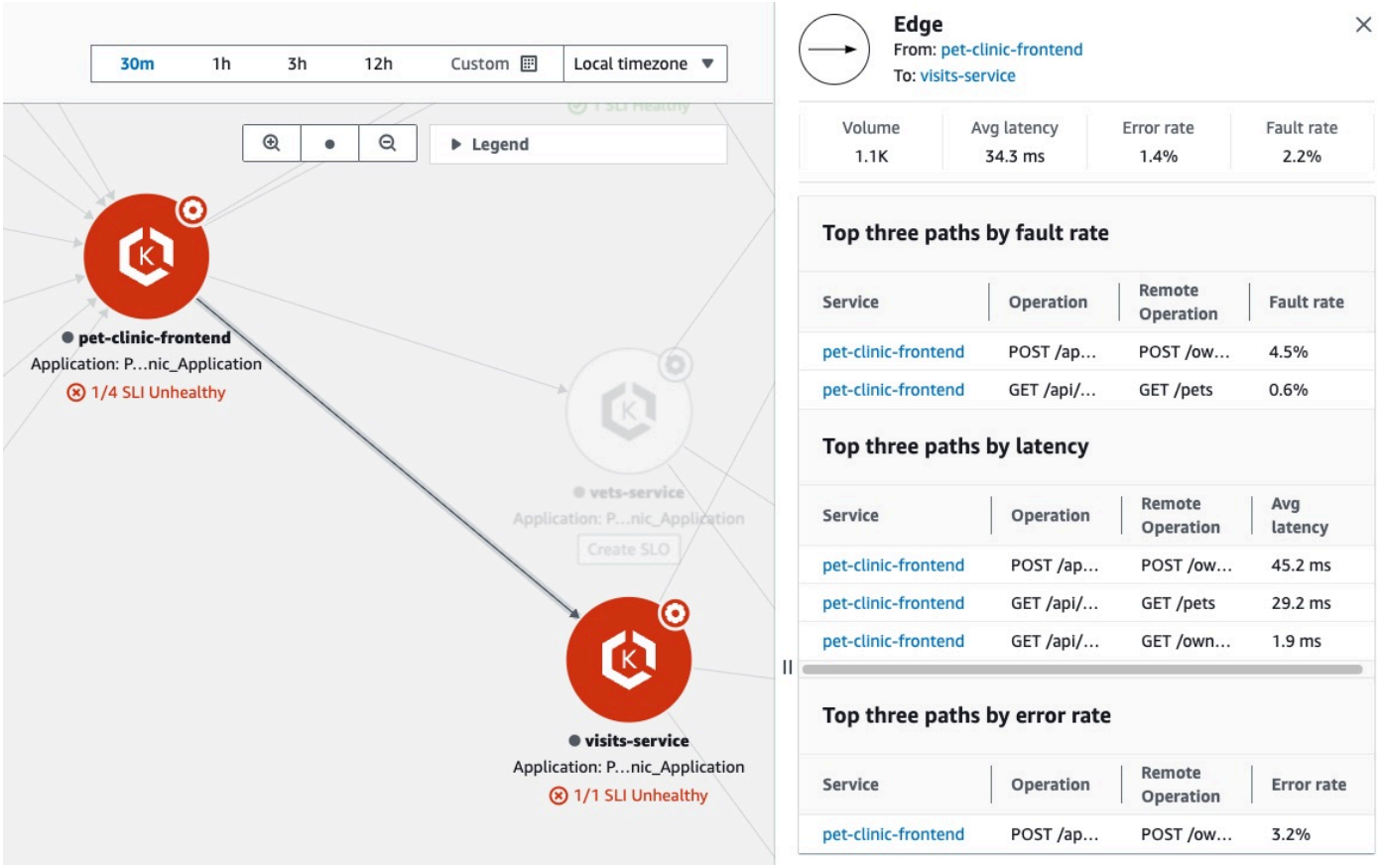
- Otros tipos de servicios de aplicaciones que no se mencionaron anteriormente:

Al seleccionar un nodo de servicio, se abre un panel con la siguiente información detallada del servicio:

- Métricas del volumen de llamadas, la latencia, los errores y la tasa de fallos.
- El número de SLI y SLO que están en `healthy` o `unhealthy`.
- La opción para ver más información acerca de un SLO.
- El número de operaciones de servicio, dependencias, canarios sintéticos y páginas de clientes.
- La opción de seleccionar cada número para abrir la página Detalles del servicio correspondiente.

- El nombre de la aplicación, si ha asociado el recurso de cómputo subyacente a una aplicación mediante AppRegistry o la tarjeta Aplicaciones en la página de inicio de la AWS Management Console.
  - Elija el nombre de la aplicación para mostrar los detalles de la aplicación en la página de la consola de myApplications.
- El `Cluster`, `Namespace` y `Workload` de los servicios alojados en Amazon EKS o el `Environment` de los servicios alojados en Amazon ECS o Amazon EC2. Para los servicios alojados en Amazon EKS, elija cualquier enlace para abrir la Información de contenedores de CloudWatch.

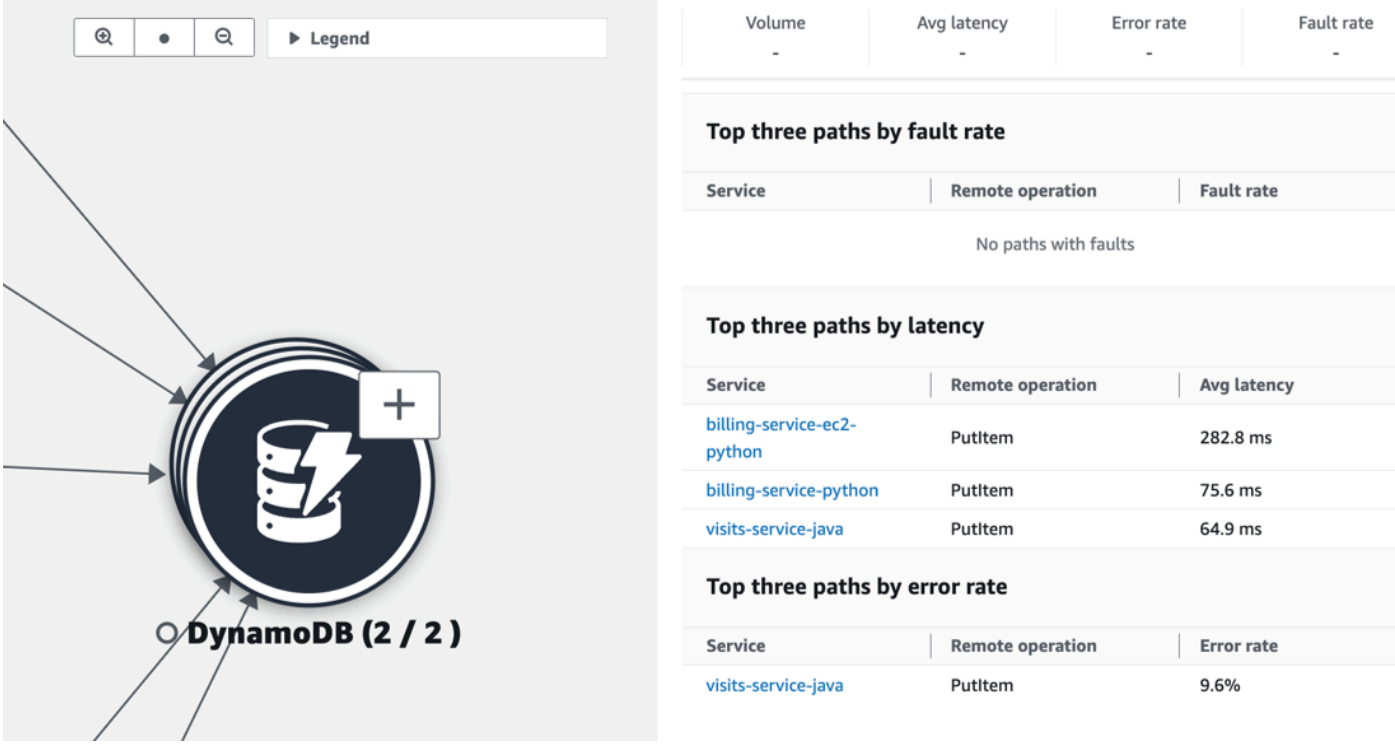
Seleccione una periferia o conexión entre un nodo de servicio y un nodo de servicio o dependencia posterior. Esto abre un panel que contiene las rutas principales por tasa de fallos, latencia y tasa de errores, como se muestra en la siguiente imagen de ejemplo. Elija cualquier enlace del panel para abrir la página Detalles del servicio y ver la información detallada del servicio o dependencia elegida.



View dependencies

Las dependencias de sus aplicaciones se muestran en la asignación de servicio, conectadas a los servicios que las llaman.

Seleccione un nodo de dependencia para abrir el panel que contiene las rutas principales por tasa de fallos, latencia y tasa de errores. Elija cualquier servicio o enlace de destino para abrir la página Detalles del servicio y ver la información detallada sobre el servicio o dependencia elegida, como se muestra en la siguiente imagen de ejemplo:

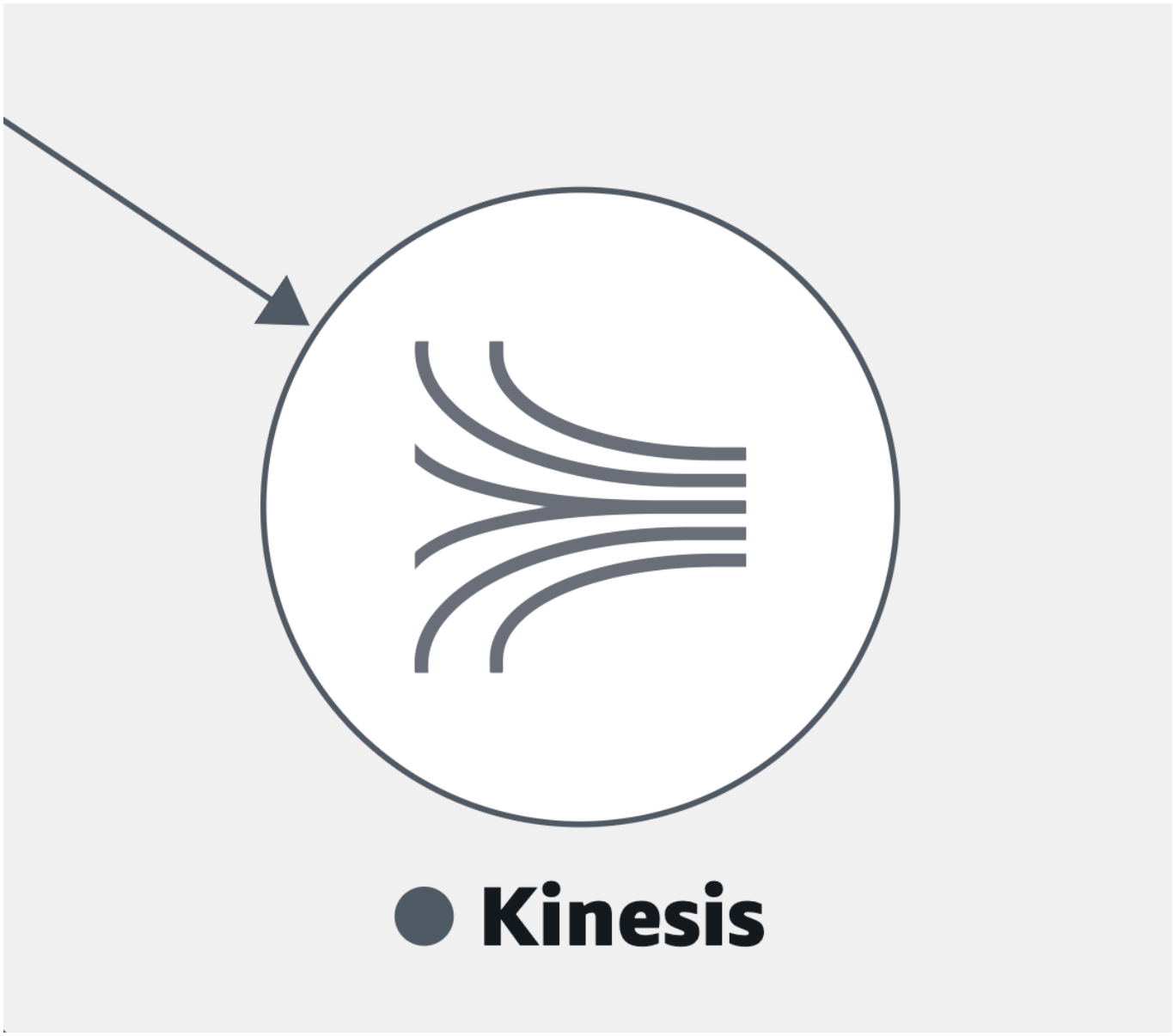


Las dependencias de los servicios se agrupan de forma predeterminada en un único icono expandible. Seleccione el ícono (+), como se muestra en la imagen anterior, para expandir el grupo y ver sus elementos individuales.

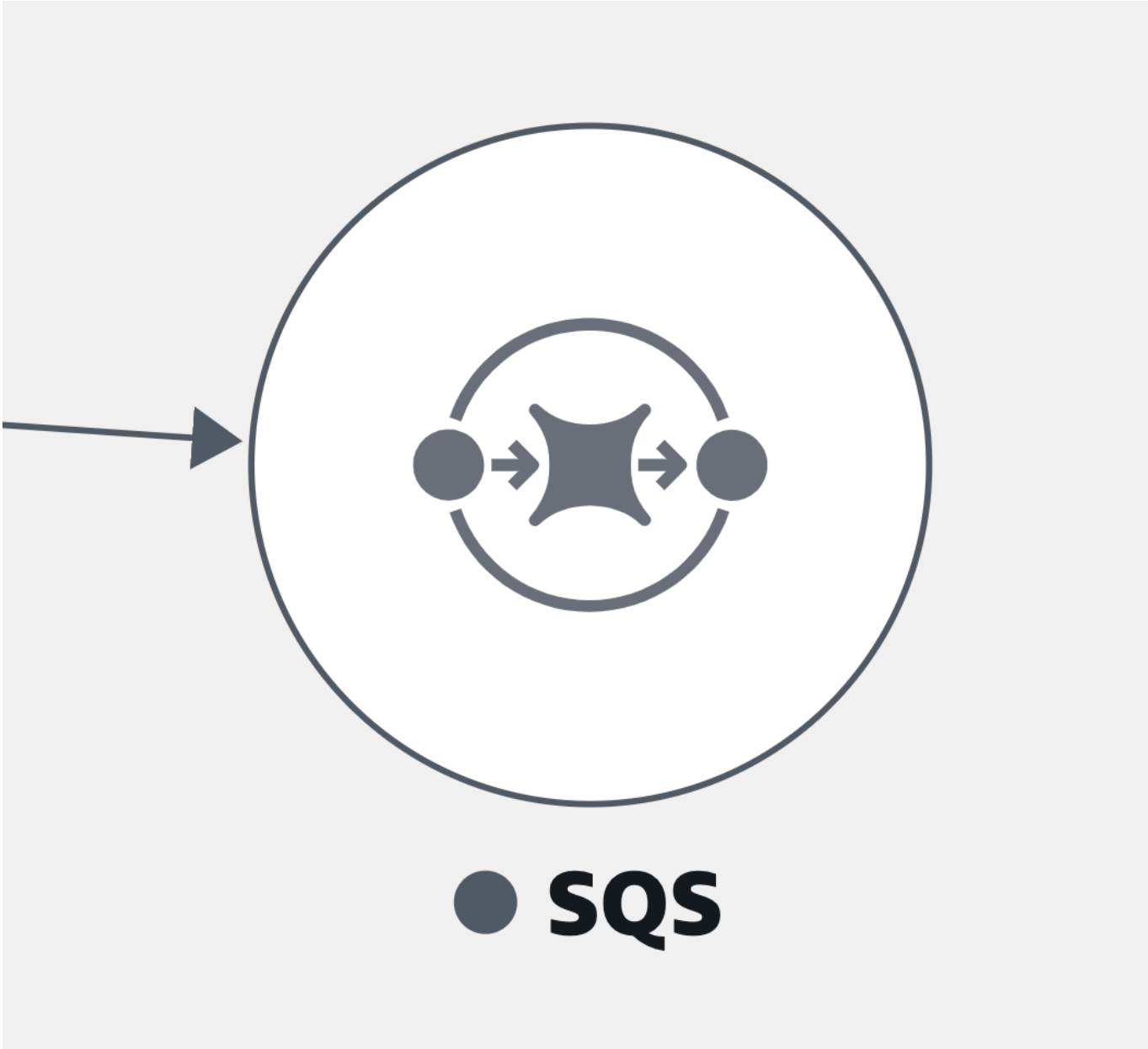
Los siguientes iconos representan ejemplos de nodos de dependencia en la asignación de servicios:

- Un bucket de Amazon S3:

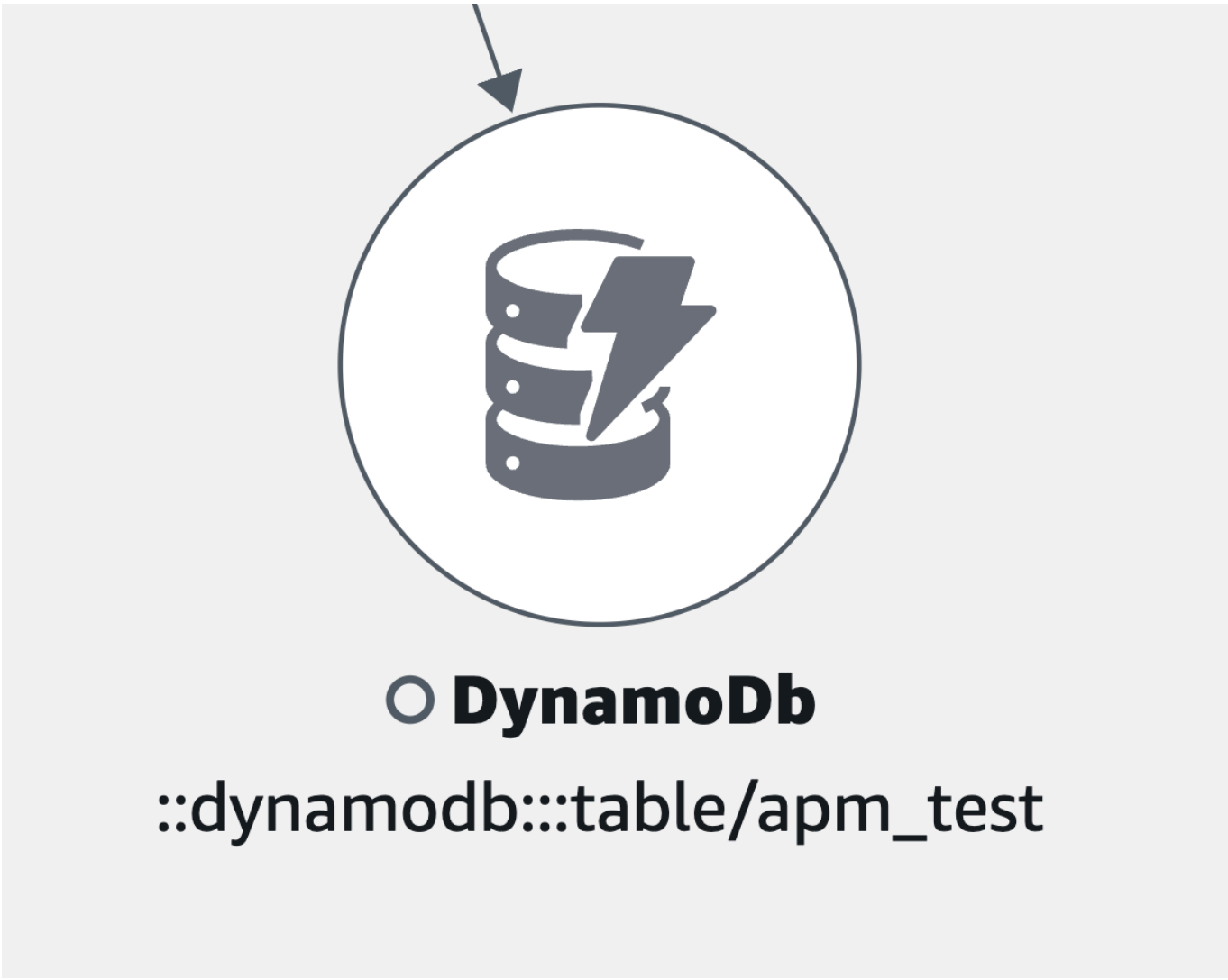
- Un flujo de Amazon Kinesis:



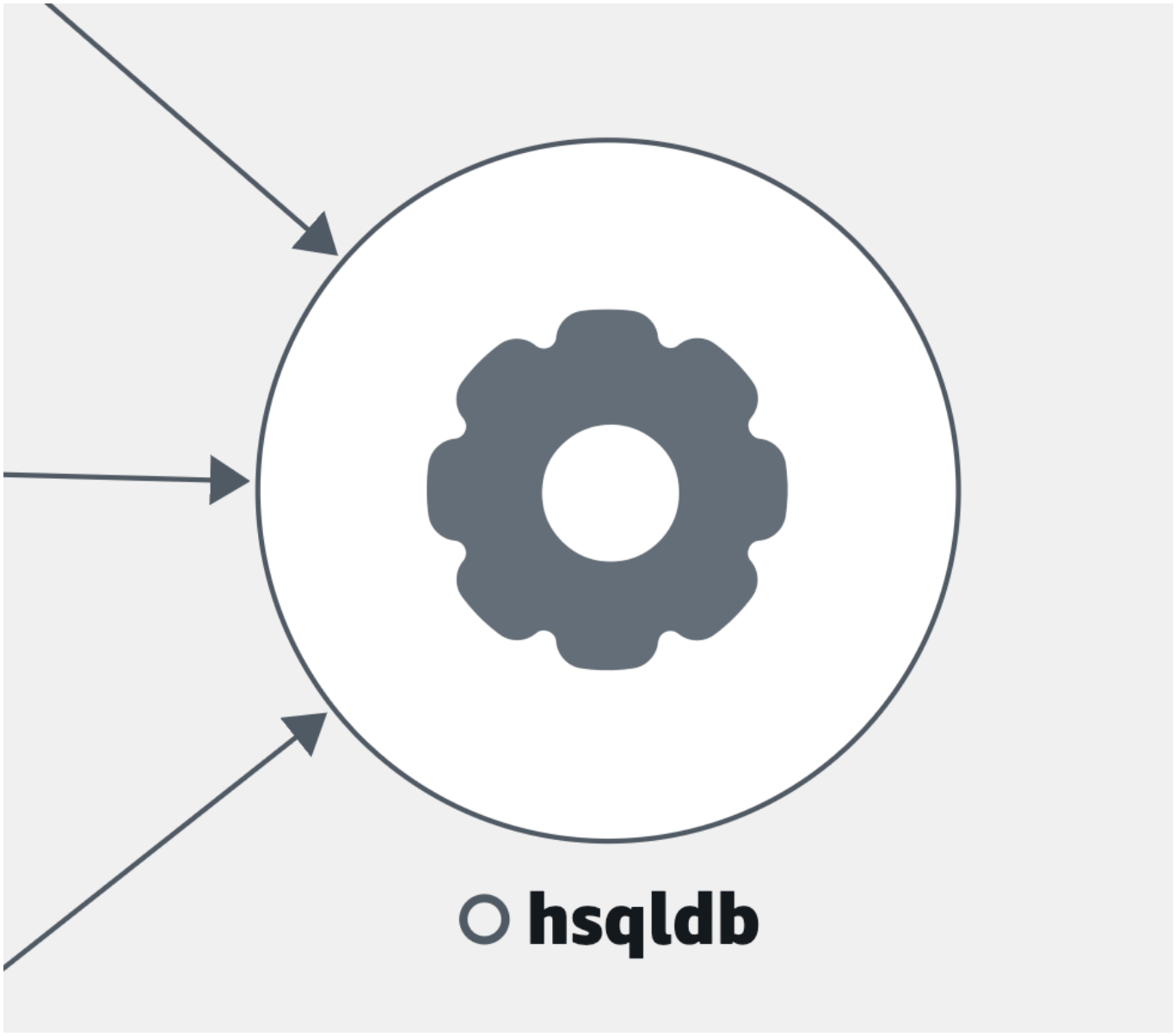
- Amazon Simple Queue Service (Amazon SQS):



- Una tabla de Amazon DynamoDB:



- Otros tipos de dependencias que no se mencionaron anteriormente:



View clients

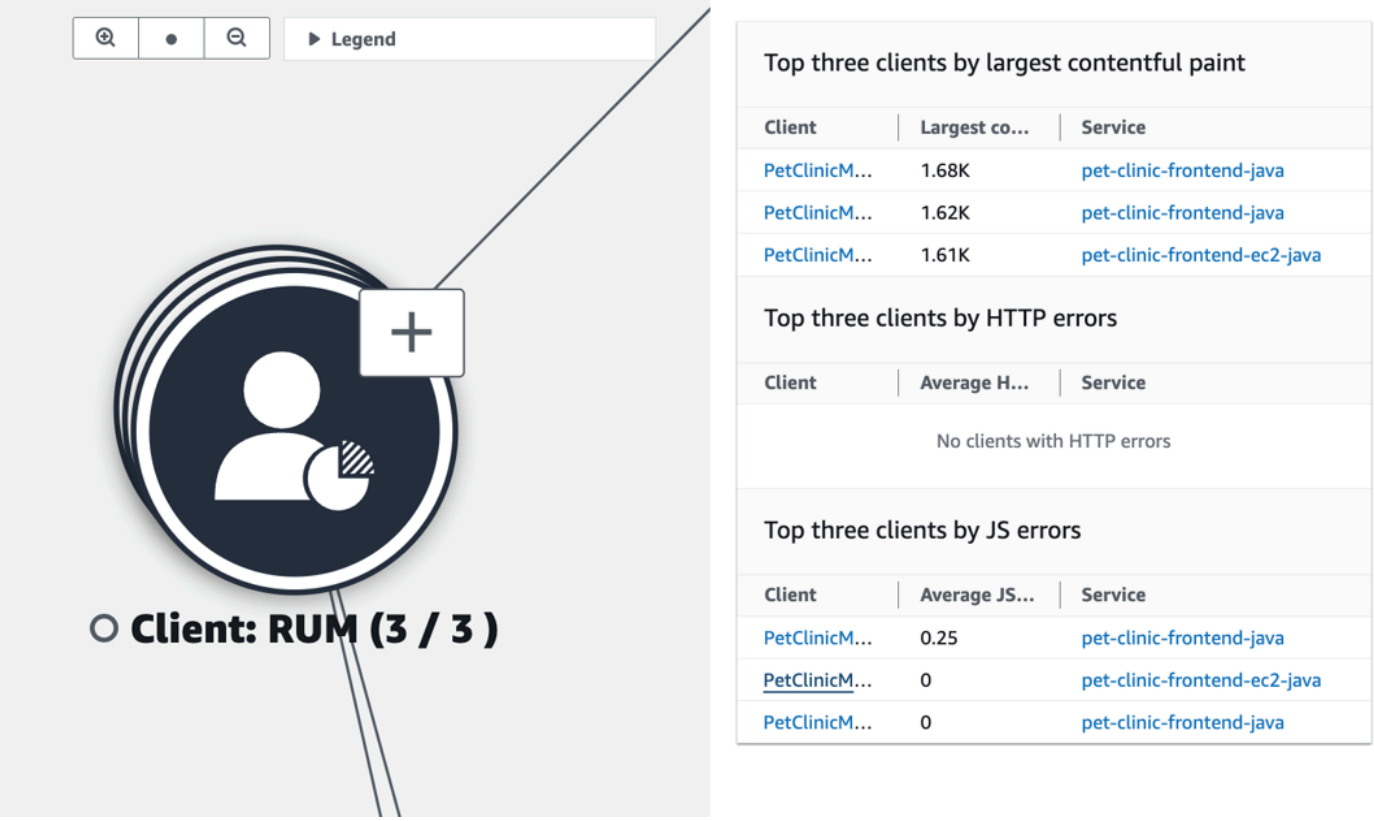
Después de activar el seguimiento de X-Ray para los clientes web de CloudWatch RUM, estos se muestran en la asignación de servicio conectada a los servicios a los que llaman.

Elija un nodo de cliente para abrir el panel que muestre la siguiente información detallada del cliente:

- Métricas de las cargas de página, tiempo promedio de carga, errores y los elementos vitales web promedio.
- Un gráfico que muestra el desglose de los errores.

- Un enlace para mostrar los detalles del cliente en CloudWatch RUM.

Las dependencias de los servicios se agrupan de forma predeterminada en un único icono expandible. Seleccione el icono (+) para expandir el grupo y ver sus elementos individuales, como se muestra en la siguiente imagen.



El siguiente icono representa un ejemplo de un cliente de RUM en la asignación de servicios:

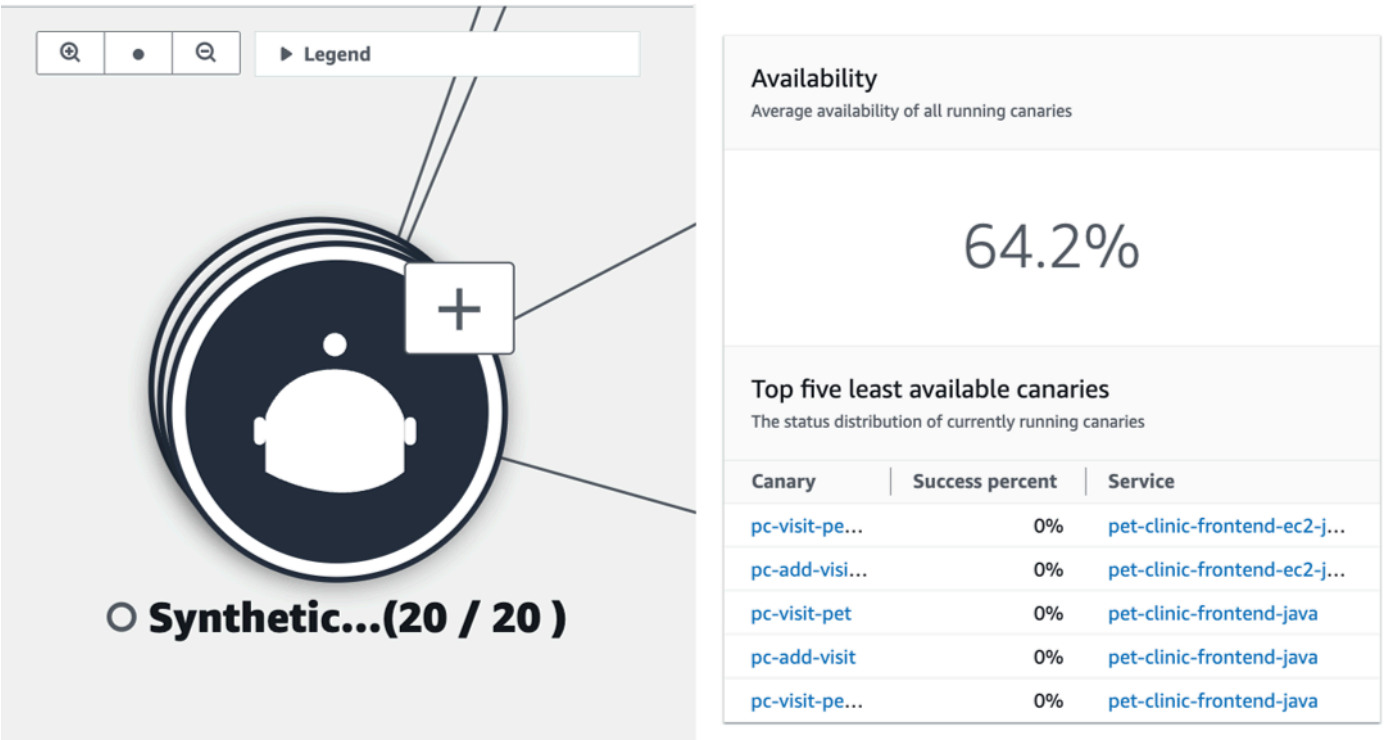
- Un cliente de RUM:

> **Note**
>
> Para ver los errores AJAX en las páginas de sus clientes, utilice la versión 1.15 de cliente web de CloudWatch RUM o posterior.

View synthetics canaries

Después de activar el seguimiento de AWS X-Ray para sus canarios de CloudWatch Synthetics, estos se muestran en la asignación de servicio conectada a los servicios a los que llaman, como se muestra en la siguiente imagen de ejemplo:

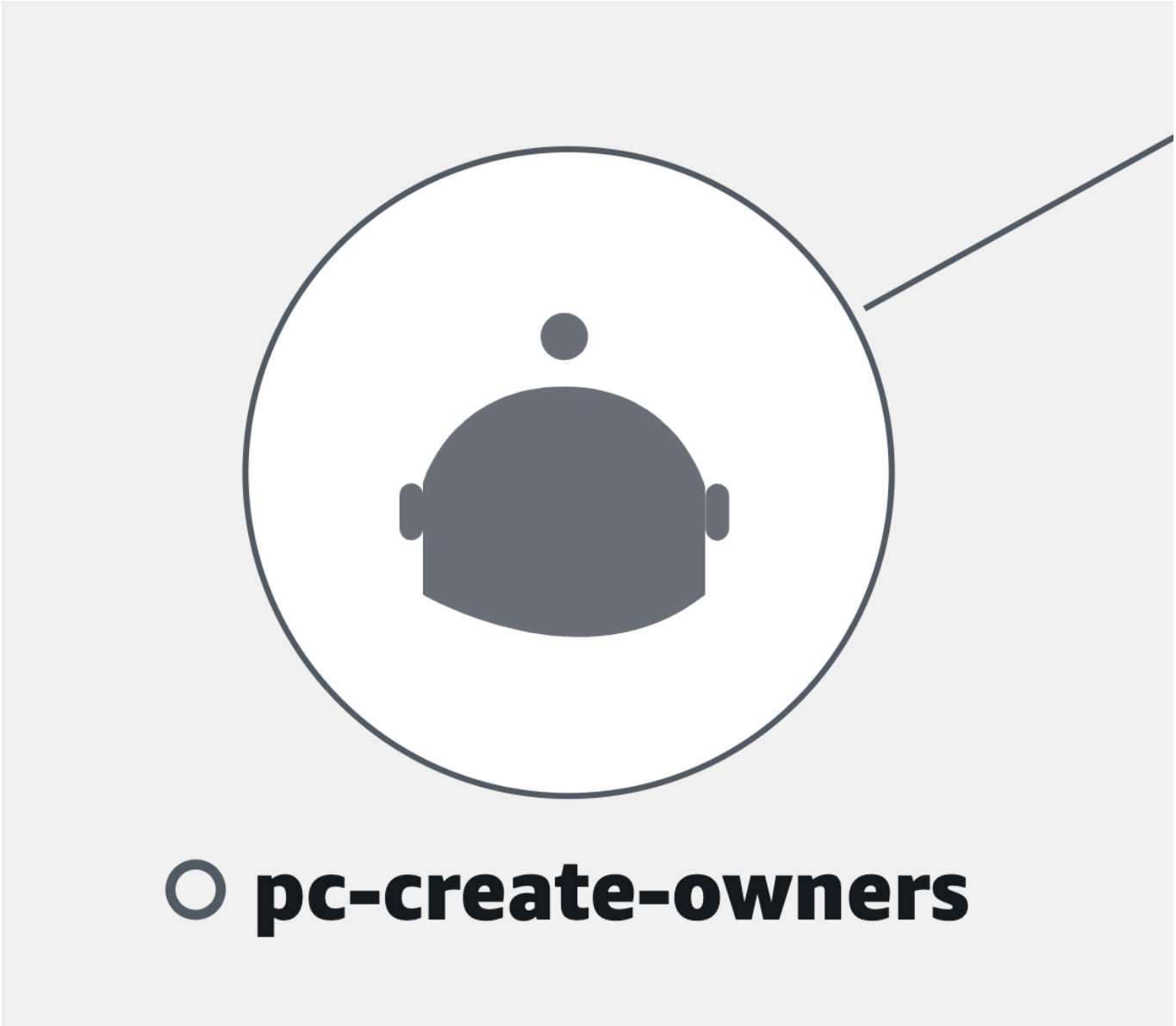
Seleccione un nodo de canario para abrir el panel que muestra la siguiente información detallada del canario, tal como se muestra en la siguiente imagen:

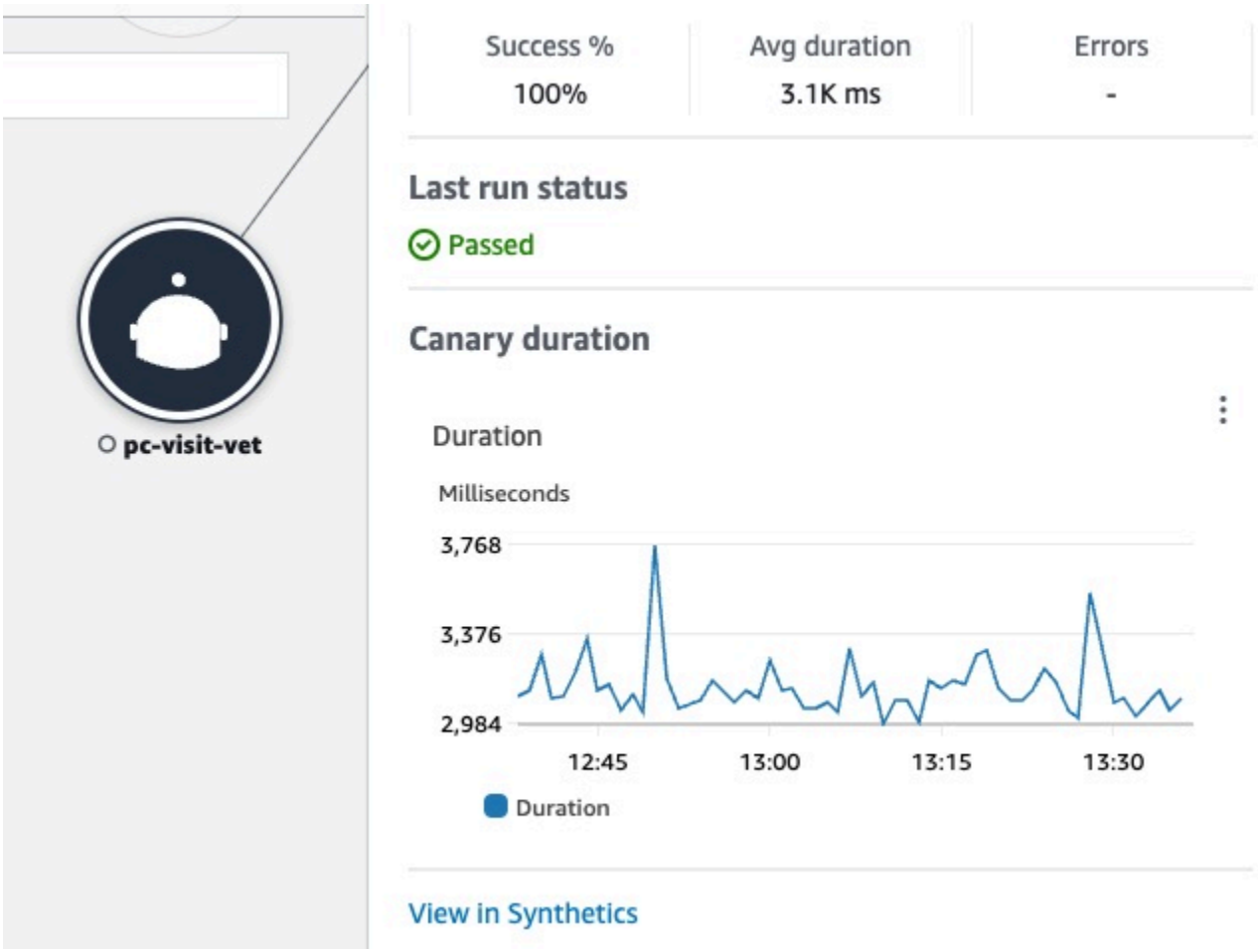


Los canarios se agrupan de forma predeterminada en un único icono expandible. Seleccione el ícono (+), como se muestra en la imagen anterior, para expandir el grupo y ver sus elementos individuales.

El siguiente icono representa un ejemplo de un cliente de RUM en la asignación de servicios:

- Un canario sintético:


pc-create-owners



En el panel de nodos de canarios, puede ver lo siguiente:

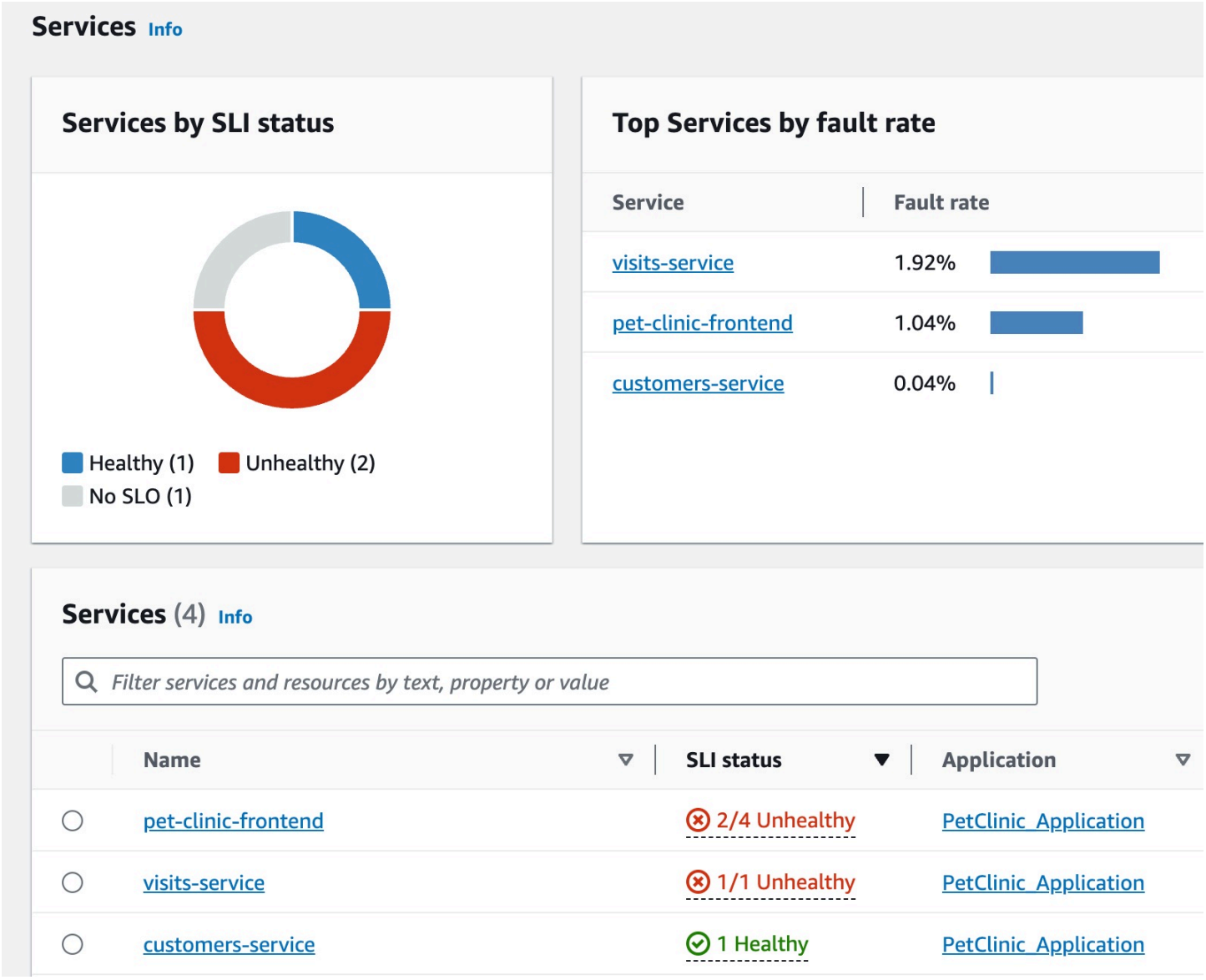
- Métricas del porcentaje correcto, la duración promedio y los errores.
- El estado de la última ejecución de valor controlado.
- Un gráfico que muestra la duración de la ejecución de valor controlado. Coloque el cursor sobre un conjunto de gráficos para ver una ventana emergente con más información.
- Un enlace para mostrar los detalles de canario en CloudWatch Synthetics.

Ejemplo: utilice Application Signals para resolver un problema de estado operativo

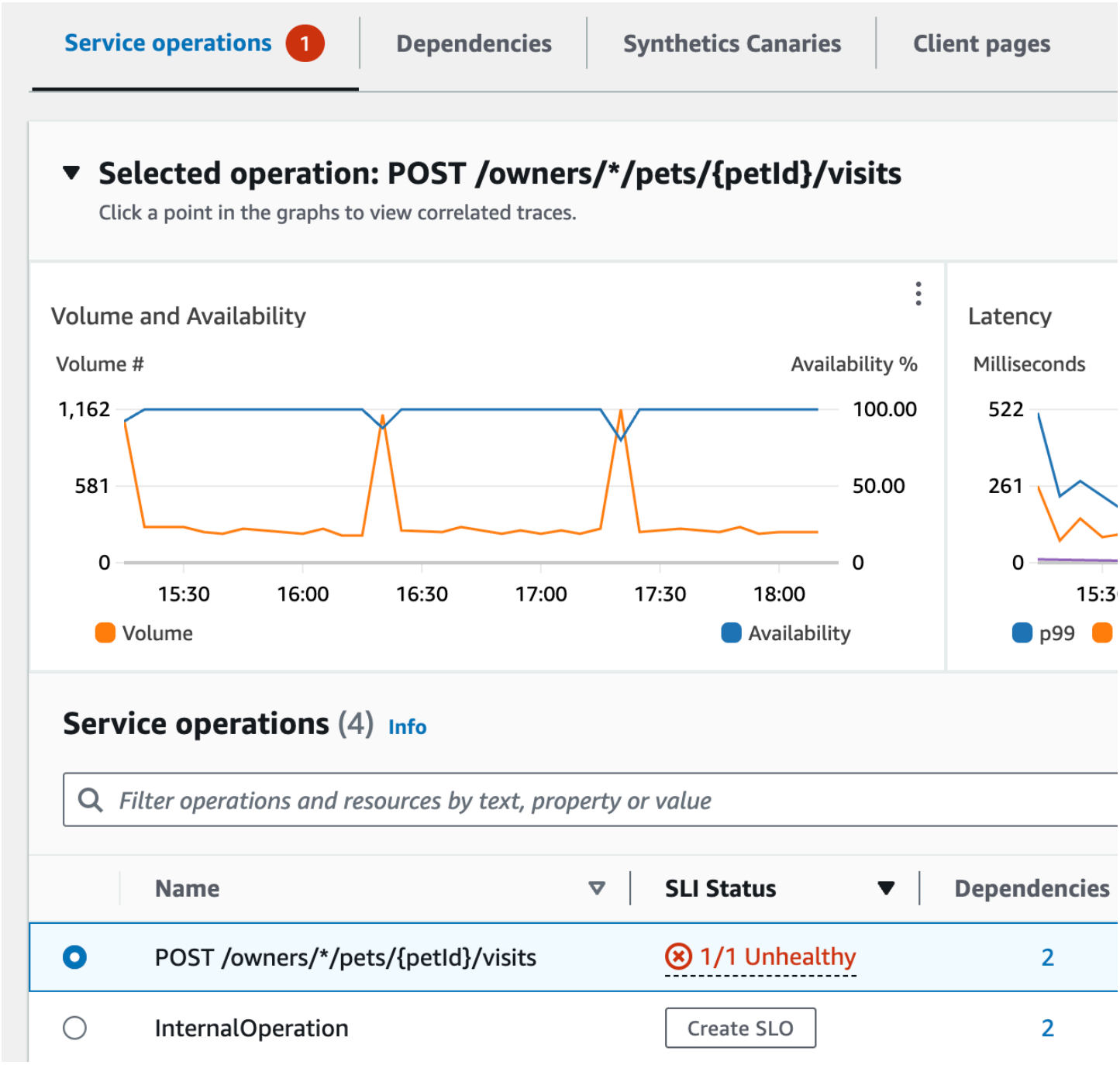
⚠ Application Signals se encuentra en versión preliminar para Amazon CloudWatch y está sujeto a cambios.

El siguiente escenario brinda un ejemplo de cómo se puede utilizar Application Signals para monitorear sus servicios e identificar problemas de calidad del servicio. Desplácese para identificar las posibles causas subyacentes y tome medidas para resolver el problema. Este ejemplo se centra en una aplicación de clínica de mascotas compuesta por varios microservicios que llaman a Servicios de AWS, como DynamoDB.

Jane forma parte de un equipo de DevOps que supervisa el estado operativo de una aplicación de clínica de mascotas. El equipo de Jane se encarga de asegurar que la aplicación tenga una alta disponibilidad y capacidad de respuesta. Utilizan objetivos de nivel de servicio (SLO) para medir el rendimiento de las aplicaciones en relación con estos compromisos empresariales. Jane recibe una alerta sobre varios indicadores de nivel de servicio (SLI) incorrectos. Abre la consola de CloudWatch y se dirige a la página Servicios, donde observa varios servicios que no funcionan de forma correcta.

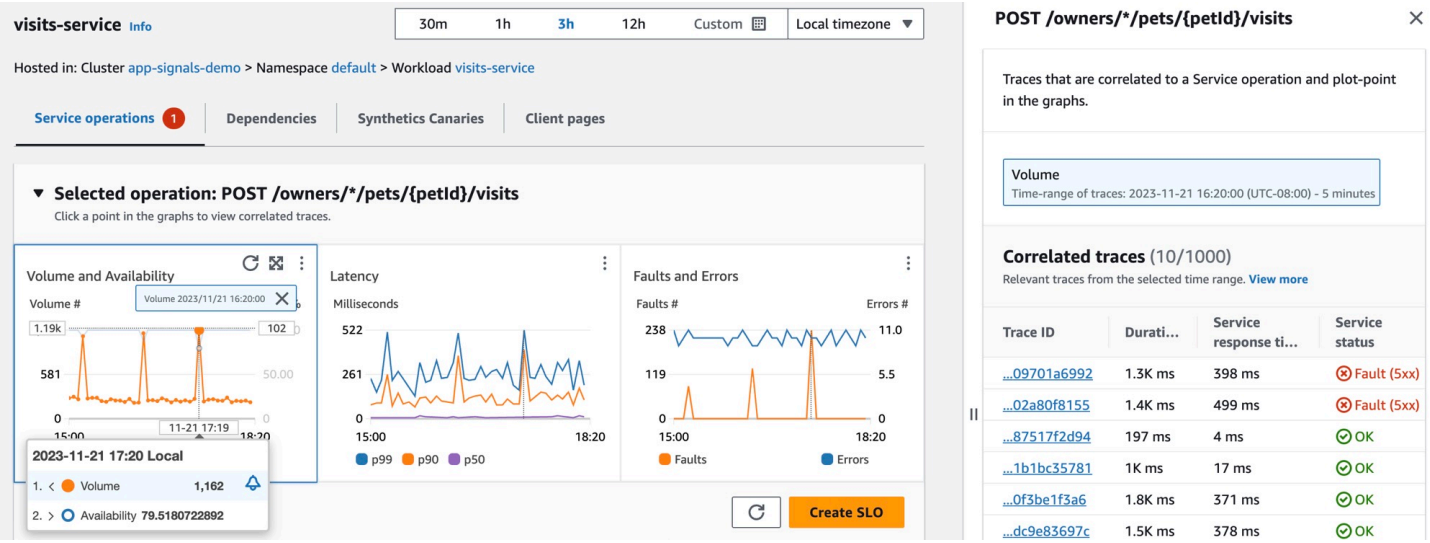


En la parte superior de la página, Jane ve que el `visits-service` es el servicio principal por tasa de fallos. Selecciona el enlace en el gráfico, que abre la página Detalles del servicio para ese servicio. Observa que hay una operación incorrecta en la tabla Operaciones del servicio. Selecciona esta operación y observa en el gráfico de Volumen y disponibilidad que hay picos periódicos en el volumen de llamadas que parecen relacionarse con caídas en la disponibilidad.

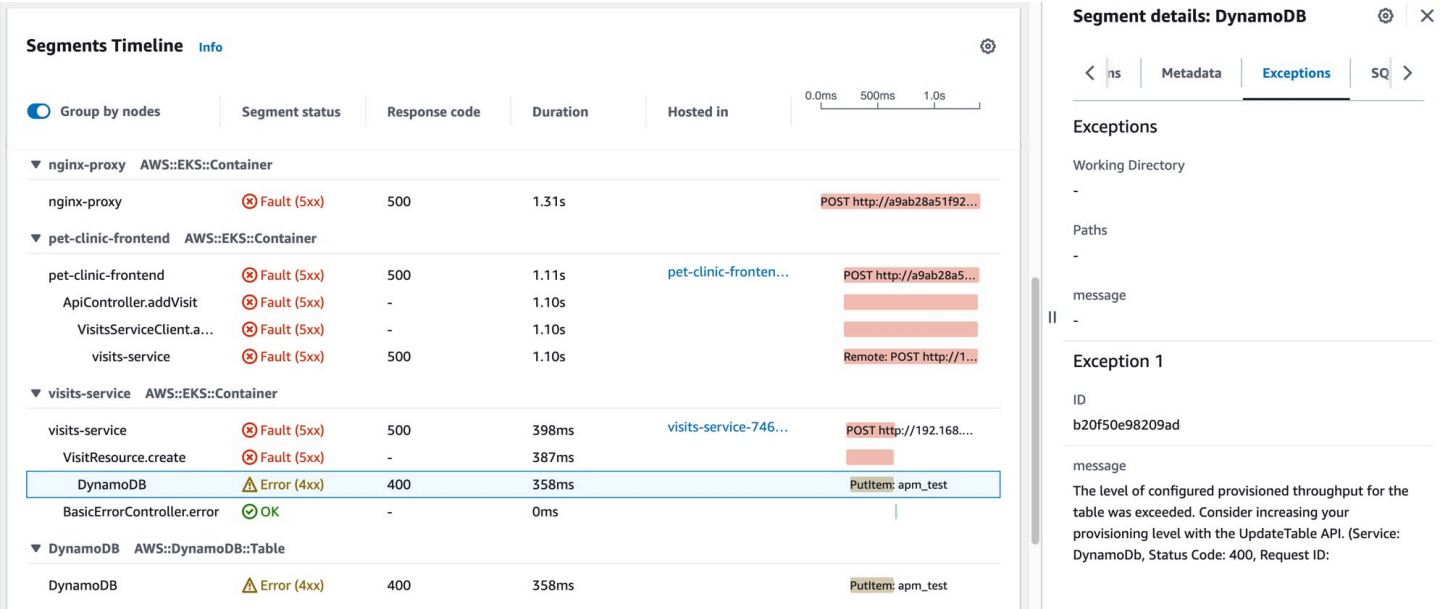


Para analizar más de cerca las caídas en la disponibilidad del servicio, Jane selecciona uno de los puntos de datos de disponibilidad del gráfico. Se abre un panel que muestra los seguimientos de X-

Ray que se correlacionan con el punto de datos seleccionado. Observa que hay varios seguimientos que contienen errores.



Jane selecciona uno de los seguimientos correlacionados con un estado de error, lo que abre la página de detalles de Seguimiento de X-Ray del seguimiento seleccionado. Jane se desplaza hacia la sección Escala de tiempo de los segmentos y sigue la ruta de llamada hasta que observa que las llamadas a una tabla de DynamoDB presentan errores. Selecciona el segmento de DynamoDB y navega hasta la pestaña Excepciones del panel de la derecha.



Jane observa que un recurso de DynamoDB está mal configurado, lo que provoca errores durante los picos de solicitudes de los clientes. El nivel de rendimiento aprovisionado de la tabla de DynamoDB se supera de forma periódica, lo que provoca problemas de disponibilidad del servicio y

SLI incorrectos. Según esta información, su equipo puede configurar un nivel superior de rendimiento aprovisionado y asegurar una alta disponibilidad de la aplicación.

Recopilación de métricas de aplicaciones estándar

⚠ Application Signals está en versión de prueba. Si tiene algún comentario sobre esta característica, puede ponerse en contacto con nosotros en app-signals-feedback@amazon.com.

Application Signals recopila métricas de aplicaciones estándar a partir de los servicios que detecta. Estas métricas se relacionan con los aspectos más críticos del rendimiento de un servicio: latencia, fallos y errores. Pueden ayudarlo a identificar problemas, monitorear las tendencias de rendimiento y optimizar los recursos para mejorar la experiencia general del usuario.

La siguiente tabla enumera las métricas recopiladas por Application Signals. Estas métricas se envían a CloudWatch en el espacio de nombres `AppSignals`.

Métrica	Descripción
`Latency`	El retraso antes de la transferencia de datos comienza una vez realizada la solicitud. Unidades: milisegundos
`Faults`	Un recuento de los errores del servidor HTTP 5XX y de los errores de estado del intervalo de OpenTelemetry. Unidades: ninguna
`Errors`	Un recuento de los errores HTTP 4XX del cliente. Se los considera errores de solicitud que no se deben a problemas de servicio. Por lo tanto, la métrica `Availability` que se muestra en los paneles de Application Signals no considera estos errores como fallos del servicio. Unidades: ninguna

La métrica `Availability` que se muestra en los paneles de Application Signals se calcula como (1 - **`Faults`**/Total)*100. Las respuestas correctas son todas las respuestas sin el error `5XX`. Las respuestas `4XX` se consideran correctas cuando Application Signals calcula la `Availability`.

Dimensiones recopiladas y combinaciones de dimensiones

Se definen las siguientes dimensiones para cada una de las métricas de aplicaciones estándar. Para obtener más información acerca de las dimensiones, consulte Dimensiones.

Se recopilan diferentes dimensiones para las métricas de servicio y las métricas de dependencia. Dentro de los servicios detectados por Application Signals, cuando el microservicio A llama al microservicio B, el microservicio B atiende la solicitud. En este caso, el microservicio A emite métricas de dependencia y el microservicio B emite métricas de servicio. Cuando un cliente llama al microservicio A, el microservicio A atiende la solicitud y emite las métricas del servicio.

Dimensiones de métricas de servicio

Se recopilan las siguientes dimensiones para las métricas de servicio.

Dimensión	Descripción
`Service`	Nombre del servicio.
`Operation`	Nombre de la operación de la API u otra actividad.
`HostedIn.EKS.Cluster`	Nombre del clúster de Amazon EKS donde se ejecutan los servicios. Esta dimensión se recopila solo si los servicios se ejecutan en Amazon EKS.
`HostedIn.K8s.Namespace`	Nombre del espacio de nombres de Kubernetes en el que se ejecutan los servicios. Esta dimensión se recopila solo si los servicios se ejecutan en Amazon EKS.
`HostedIn.Environment`	Nombre definido por el usuario del entorno en el que se ejecutan los servicios.

Dimensión	Descripción
	Esta dimensión se recopila solo si los servicios se ejecutan en un entorno que no es Amazon EKS.

Al observar estas métricas en la consola de CloudWatch, puede elegir verlas con las siguientes combinaciones de dimensiones.

- `Service`, `Operation`, `HostedIn.EKS.Cluster`, `HostedIn.K8s.Namespace`
- `Service`, `HostedIn.EKS.Cluster`, `HostedIn.K8s.Namespace`

En el caso de las plataformas que no son Amazon EKS, también puede ver las métricas de servicio con las siguientes combinaciones de dimensiones.

- `Service`, `Operation`, `HostedIn.Environment`
- `Service`, `HostedIn.Environment`

Dimensiones de las métricas de dependencia

Se recopilan las siguientes dimensiones para las métricas de dependencia.

Dimensión	Descripción
`Service`	Nombre del servicio.
`Operation`	Nombre de la operación de la API u otra actividad.
`RemoteService`	Nombre del servicio remoto que se invoca.
`RemoteOperation`	Nombre de la operación de API que se invoca.
`HostedIn.EKS.Cluster`	Nombre del clúster de Amazon EKS donde se ejecutan los servicios. Esta dimensión se recopila solo si los servicios se ejecutan en Amazon EKS.
`HostedIn.K8s.Namespace`	Nombre del espacio de nombres de Kubernetes en el que se ejecutan los servicios.

Dimensión	Descripción
	Esta dimensión se recopila solo si los servicios se ejecutan en Amazon EKS.
`K8s.RemoteNamespace`	El nombre del espacio de nombres de Kubernetes en el que se ejecutan los servicios de dependencia. Esta dimensión se recopila solo si los servicios se ejecutan en Amazon EKS.
`RemoteTarget`	Nombre del recurso invocado por las llamadas remotas. Esta dimensión no tiene valor si las llamadas remotas no se dirigen a recursos específicos. Esta dimensión se recopila solo si los servicios se ejecutan en Amazon EKS.
`HostedIn.Environment`	Nombre definido por el usuario del entorno en el que se ejecutan los servicios. Esta dimensión se recopila solo si los servicios se ejecutan en un entorno que no es Amazon EKS.

Al observar estas métricas en la consola de CloudWatch, puede elegir verlas con las siguientes combinaciones de dimensiones.

Se ejecuta en cualquier plataforma

- `RemoteService`

Se ejecuta en clústeres de Amazon EKS

- `Service`, `Operation`, `HostedIn.EKS.Cluster`, `HostedIn.K8s.Namespace`, `RemoteService`, `RemoteOperation`, `K8s.RemoteNamespace`, `RemoteTarget`
- `Service`, `Operation`, `HostedIn.EKS.Cluster`, `HostedIn.K8s.Namespace`, `RemoteService`, `RemoteOperation`, `K8s.RemoteNamespace`

- `Service`, `Operation`, `HostedIn.EKS.Cluster`, `HostedIn.K8s.Namespace`, `RemoteService`, `RemoteOperation`, `RemoteTarget`
- `Service`, `Operation`, `HostedIn.EKS.Cluster`, `RemoteService`, `RemoteOperation`,
- `Service`, `HostedIn.EKS.Cluster`, `HostedIn.K8s.Namespace`, `RemoteService`, `K8s.RemoteNamespace`
- `Service`, `HostedIn.EKS.Cluster`, `RemoteService`, `K8s.RemoteNamespace`
- `Service`, `HostedIn.EKS.Cluster`, `HostedIn.K8s.Namespace`, `RemoteService`, `RemoteOperation`, `K8s.RemoteNamespace`, `RemoteTarget`
- `Service`, `HostedIn.EKS.Cluster`, `HostedIn.K8s.Namespace`, `RemoteService`, `RemoteOperation`, `K8s.RemoteNamespace`
- `Service`, `HostedIn.EKS.Cluster`, `HostedIn.K8s.Namespace`, `RemoteService`, `RemoteOperation`, `RemoteTarget`
- `Service`, `HostedIn.EKS.Cluster`, `HostedIn.K8s.Namespace`, `RemoteService`, `RemoteOperation`

Se ejecuta en plataformas distintas de los clústeres de Amazon EKS

- `Service`, `Operation`, `HostedIn.Environment`
- `Service`, `HostedIn.Environment`
- `Service`, `Operation`, `HostedIn.Environment`, `RemoteService`, `RemoteOperation`, `RemoteTarget`
- `Service`, `Operation`, `HostedIn.Environment`, `RemoteService`, `RemoteOperation`,
- `Service`, `HostedIn.Environment`, `RemoteService`
- `Service`, `HostedIn.Environment`, `RemoteService`, `RemoteOperation`, `RemoteTarget`
- `Service`, `HostedIn.Environment`, `RemoteService`, `RemoteOperation`,

Uso de la supervisión sintética

Puede utilizar Amazon CloudWatch Synthetics para crear canarios, scripts configurables que se ejecutan según una programación, para supervisar los puntos de enlace y las API. Los Canaries siguen las mismas rutas y realizan las mismas acciones que un cliente, lo que le permite verificar continuamente su experiencia de cliente incluso cuando no tiene tráfico de clientes en sus

aplicaciones. Mediante el uso de canaries, puede descubrir problemas antes de que sus clientes lo hagan.

Los canaries son scripts escritos en Node.js o en Python. Crean funciones de Lambda en la cuenta que usan Node.js o Python como marco. Los canaries funcionan a través de protocolos HTTP y HTTPS. Los valores controlados utilizan capas de Lambda que contienen la biblioteca de CloudWatch Synthetics. La biblioteca contiene la versión NodeJS de CloudWatch Synthetics para valores controlados de NodeJS y la versión Python de CloudWatch Synthetics para valores controlados de Python. Las capas pertenecen a la cuenta de servicio de CloudWatch Synthetics. Las bibliotecas nunca transmiten ni almacenan información de los clientes. Todos los datos de los clientes se almacenan únicamente en la cuenta del cliente.

Los canaries ofrecen acceso programático a un navegador Google Chrome headless a través de Puppeteer o Selenium Webdriver. Para obtener más información acerca de Puppeteer, consulte Puppeteer. Para obtener más información acerca de Selenium, consulte www.selenium.dev/.

Los Canaries comprueban la disponibilidad y latencia de sus puntos de enlace, y pueden almacenar datos de tiempo de carga y capturas de pantalla de la interfaz de usuario. Supervisan las API REST, las URL y el contenido del sitio web, y pueden comprobar si hay cambios no autorizados de suplantación de identidad, inyección de código y scripting entre sitios.

CloudWatch Synthetics se integra con Application Signals, que puede detectar y supervisar los servicios de aplicaciones, los clientes, los valores controlados de Synthetics y las dependencias de los servicios. Use Application Signals para ver una lista o un mapa visual de sus servicios, ver las métricas del estado en función de los objetivos de nivel de servicio (SLO) y profundizar para ver los seguimientos de X-Ray correlacionados para una solución de problemas más detallada. Para ver sus valores controlados en Application Signals, active el seguimiento activo de X-Ray. Los valores controlados se muestran en el Mapa de servicio conectado a sus servicios y en la página de Detalles del servicio de los servicios a los que llaman.

Para ver una demostración en video de los valores controlados, consulte lo siguiente:

- Introduction to Amazon CloudWatch Synthetics (Introducción a Amazon CloudWatch Synthetics)
- Amazon CloudWatch Synthetics Demo (Demostración de Amazon CloudWatch Synthetics)
- Create Canaries Using Amazon CloudWatch Synthetics (Crear valores controlados con Amazon CloudWatch Synthetics)
- Visual Monitoring with Amazon CloudWatch Synthetics (Supervisión visual con Amazon CloudWatch Synthetics)

Puede ejecutar un valor controlado una vez o de forma periódica. Los canaries pueden ejecutarse con una frecuencia de una vez por minuto. Puede usar expresiones cron y de frecuencia para programar valores controlados.

Para obtener información acerca de los problemas de seguridad que deben tenerse en cuenta antes de crear y ejecutar valores programados, consulte Consideraciones de seguridad para los canaries de Synthetics.

De forma predeterminada, los canaries crean algunas métricas de CloudWatch en el espacio de nombres de `CloudWatchSynthetics`. Estas métricas tienen `CanaryName` como dimensión. Los canaries que utilizan la función `executeStep()` o `executeHttpStep()` de la biblioteca de funciones también tienen `StepName` como dimensión. Para obtener más información sobre la biblioteca de funciones de valor controlado, consulte Funciones de la biblioteca disponibles para los scripts de valor controlado.

CloudWatch Synthetics se integra bien con el mapa de seguimiento de X-Ray, el cual utiliza CloudWatch con AWS X-Ray para proporcionar una visión integral de los servicios a fin de ayudarlo a detallar de manera más eficiente los cuellos de botella de rendimiento e identificar a los usuarios afectados. Los valores controlados que crea con CloudWatch Synthetics aparecen en el mapa de seguimiento. Para obtener más información, consulte Mapa de seguimiento de X-Ray.

Actualmente, CloudWatch Synthetics está disponible en todas las regiones comerciales de AWS y en las regiones de GovCloud.

> **Note**
>
> En Asia-Pacífico (Osaka), AWS PrivateLink no es compatible. En Asia-Pacífico (Yakarta), AWS PrivateLink y X-Ray no son compatibles.

Temas

- Canaries y rastreo X-Ray
- Ejecución de un valor controlado en una VPC
- Cifrado de artefactos de un valor controlado
- Visualización de las estadísticas y los detalles de los valores controlados
- Métricas de CloudWatch que los canaries publican
- Edición o eliminación de un valor controlado
- Inicio, detención, eliminación o actualización del tiempo de ejecución de varios valores controlados
- Supervisión de eventos del valor controlado con Amazon EventBridge

Roles y permisos necesarios para los canaries de CloudWatch

Tanto los usuarios que crean y administran valores controlados como los propios valores controlados deben tener ciertos permisos.

Roles y permisos necesarios para los usuarios que administran valores controlados de CloudWatch

Para ver los detalles de los valores controlados y los resultados de sus ejecuciones, debe iniciar sesión como usuario con las políticas adjuntas de `CloudWatchSyntheticsFullAccess` o `CloudWatchSyntheticsReadOnlyAccess`. Para leer todos los datos de Synthetics en la consola, también necesita las políticas `AmazonS3ReadOnlyAccess` y `CloudWatchReadOnlyAccess`. Para ver el código fuente utilizado por canaries, también necesita la política `AWSLambda_ReadOnlyAccess`.

Para crear valores controlados, debe haber iniciado sesión como un usuario que tenga la política de `CloudWatchSyntheticsFullAccess` o un conjunto similar de permisos. Para crear Roles de IAM para los canaries, también necesita la siguiente declaración de política insertada:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iam:CreateRole",
                "iam:CreatePolicy",
                "iam:AttachRolePolicy"
```

```
            ],
            "Resource": [
                "arn:aws:iam::*:role/service-role/CloudWatchSyntheticsRole*",
                "arn:aws:iam::*:policy/service-role/CloudWatchSyntheticsPolicy*"
            ]
        }
    ]
}
```

> **Important**
>
> Otorgar a un usuario los permisos `iam:CreateRole`, `iam:CreatePolicy` e `iam:AttachRolePolicy` proporciona a ese usuario acceso administrativo a su cuenta de AWS. Por ejemplo, un usuario con estos permisos puede crear una política que tenga permisos completos para todos los recursos y puede asociarla a cualquier rol. Sea muy cauteloso en lo referente a la persona a la que concede estos permisos.

Para obtener información acerca de cómo asociar políticas y conceder permisos a los usuarios, consulte Cambio de los permisos de un usuario de IAM y Para integrar una política en línea de un usuario o un rol.

Roles y permisos necesarios para los valores controlados

Cada valor controlado debe estar asociado a un rol de IAM que tenga ciertos permisos adjuntos. Cuando crea un valor controlado con la consola de CloudWatch, puede elegir que CloudWatch Synthetics cree un rol de IAM para el valor controlado. Si lo hace, el rol contará con los permisos necesarios.

Si desea crear el rol de IAM o desea crear un rol de IAM que pueda utilizar cuando use la AWS CLI o las API para crear un valor controlado, el rol debe contener los permisos enumerados en esta sección.

Todos los roles de IAM para valores controlados deben incluir la siguiente declaración de política de confianza.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
```

```
            "Effect": "Allow",
            "Principal": {
                "Service": "lambda.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

Además, el rol de IAM del valor controlado necesita una de las siguientes declaraciones.

Valor controlado básico que no usa AWS KMS ni necesita acceso a Amazon VPC

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:PutObject",
                "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::path/to/your/s3/bucket/canary/results/folder"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetBucketLocation"
            ],
            "Resource": [
                "arn:aws:s3:::name/of/the/s3/bucket/that/contains/canary/results"

            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogStream",
                "logs:PutLogEvents",
                "logs:CreateLogGroup"
            ],
            "Resource": [
```

```
                "arn:aws:logs:canary_region_name:canary_account_id:log-group:/aws/
lambda/cwsyn-canary_name-*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:ListAllMyBuckets",
                "xray:PutTraceSegments"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Effect": "Allow",
            "Resource": "*",
            "Action": "cloudwatch:PutMetricData",
            "Condition": {
               "StringEquals": {
                "cloudwatch:namespace": "CloudWatchSynthetics"
               }
            }
        }
    ]
}
```

Valor controlado que usa AWS KMS para cifrar los artefactos de valor controlado, pero que no necesita acceso a Amazon VPC

```
{
    "Version": "2012-10-17",
    "Statement": [{
            "Effect": "Allow",
            "Action": [
                "s3:PutObject",
                "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::path/to/your/S3/bucket/canary/results/folder"
            ]
        },
        {
```

```
            "Effect": "Allow",
            "Action": [
                "s3:GetBucketLocation"
            ],
            "Resource": [
                "arn:aws:s3:::name/of/the/S3/bucket/that/contains/canary/results"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogStream",
                "logs:PutLogEvents",
                "logs:CreateLogGroup"
            ],
            "Resource": [
                "arn:aws:logs:canary_region_name:canary_account_id:log-group:/aws/
lambda/cwsyn-canary_name-*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:ListAllMyBuckets",
                "xray:PutTraceSegments"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Effect": "Allow",
            "Resource": "*",
            "Action": "cloudwatch:PutMetricData",
            "Condition": {
                "StringEquals": {
                    "cloudwatch:namespace": "CloudWatchSynthetics"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt",
```

```
                "kms:GenerateDataKey"
            ],
            "Resource":
 "arn:aws:kms:KMS_key_region_name:KMS_key_account_id:key/KMS_key_id",
            "Condition": {
                "StringEquals": {
                    "kms:ViaService": [
                        "s3.region_name_of_the_canary_results_S3_bucket.amazonaws.com"
                    ]
                }
            }
        }
    ]
}
```

Valor controlado que no usa AWS KMS, pero sí necesita acceso a Amazon VPC

```
{
    "Version": "2012-10-17",
    "Statement": [{
            "Effect": "Allow",
            "Action": [
                "s3:PutObject",
                "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::path/to/your/S3/bucket/canary/results/folder"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetBucketLocation"
            ],
            "Resource": [
                "arn:aws:s3:::name/of/the/S3/bucket/that/contains/canary/results"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogStream",
                "logs:PutLogEvents",
```

```
                "logs:CreateLogGroup"
            ],
            "Resource": [
                "arn:aws:logs:canary_region_name:canary_account_id:log-group:/aws/
lambda/cwsyn-canary_name-*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:ListAllMyBuckets",
                "xray:PutTraceSegments"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Effect": "Allow",
            "Resource": "*",
            "Action": "cloudwatch:PutMetricData",
            "Condition": {
                "StringEquals": {
                    "cloudwatch:namespace": "CloudWatchSynthetics"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateNetworkInterface",
                "ec2:DescribeNetworkInterfaces",
                "ec2:DeleteNetworkInterface"
            ],
            "Resource": [
                "*"
            ]
       }
    ]
}
```

Valor controlado que usa AWS KMS para cifrar los artefactos de valor controlado y también necesita acceso a Amazon VPC

Si actualiza un valor controlado que no es de VPC de manera que comience a utilizar una VPC, tendrá que actualizar el rol del valor controlado para que incluya los permisos de la interfaz de red enumerados en la siguiente política.

```
{
    "Version": "2012-10-17",
    "Statement": [{
            "Effect": "Allow",
            "Action": [
                "s3:PutObject",
                "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::path/to/your/S3/bucket/canary/results/folder"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetBucketLocation"
            ],
            "Resource": [
                "arn:aws:s3:::name/of/the/S3/bucket/that/contains/canary/results"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogStream",
                "logs:PutLogEvents",
                "logs:CreateLogGroup"
            ],
            "Resource": [
                "arn:aws:logs:canary_region_name:canary_account_id:log-group:/aws/
lambda/cwsyn-canary_name-*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:ListAllMyBuckets",
                "xray:PutTraceSegments"
            ],
```

```
            "Resource": [
                "*"
            ]
        },
        {
            "Effect": "Allow",
            "Resource": "*",
            "Action": "cloudwatch:PutMetricData",
            "Condition": {
                "StringEquals": {
                    "cloudwatch:namespace": "CloudWatchSynthetics"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateNetworkInterface",
                "ec2:DescribeNetworkInterfaces",
                "ec2:DeleteNetworkInterface"
            ],
            "Resource": [
                "*"
            ]
       },
       {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt",
                "kms:GenerateDataKey"
            ],
            "Resource":
 "arn:aws:kms:KMS_key_region_name:KMS_key_account_id:key/KMS_key_id",
            "Condition": {
                "StringEquals": {
                    "kms:ViaService": [
                        "s3.region_name_of_the_canary_results_S3_bucket.amazonaws.com"
                    ]
                }
            }
        }
    ]
}
```

Políticas administradas de AWS para CloudWatch Synthetics

Para agregar permisos a usuarios, grupos y roles, es más fácil utilizar las políticas administradas de AWS que escribirlas uno mismo. Se necesita tiempo y experiencia para crear políticas administradas por el cliente de IAM que le brinden a su equipo solo los permisos necesarios. Para comenzar a hacerlo con rapidez, puede utilizar nuestras políticas administradas de AWS. Estas políticas cubren casos de uso comunes y están disponibles en su cuenta de AWS. Para obtener más información acerca de las políticas administradas por AWS, consulte Políticas administradas por AWS Políticas administradas por AWS en la Guía del usuario de IAM.

Los servicios de AWS mantienen y actualizan las políticas administradas por AWS. No puede cambiar los permisos en las políticas administradas de AWS. Los servicios cambian ocasionalmente los permisos en una política administrada de AWS. Este tipo de actualización afecta a todas las identidades (usuarios, grupos y roles) donde se asocia la política.

CloudWatch Synthetics se actualiza según las políticas administradas de AWS

Puede consultar los detalles sobre las actualizaciones de las políticas administradas de AWS para CloudWatch Synthetics desde que el servicio comenzó a rastrear estos cambios. Para obtener alertas automáticas sobre cambios en esta página, suscríbase a la fuente RSS en la página de historial de documentos de CloudWatch

Cambio	Descripción	Fecha
Acciones redundantes eliminadas de CloudWatchSyntheticsFullAccess	CloudWatch Synthetics eliminó las acciones `s3:PutBucketEncryption` y `lambda:GetLayerVersionByArn` de la política CloudWatchSyntheticsFullAccess porque esas acciones eran redundantes con otros permisos de la política. Las acciones eliminadas no proporcionaron ningún permiso y no hay ningún cambio neto en los	12 de marzo de 2021

Cambio	Descripción	Fecha
	permisos que la política ha otorgado.	
CloudWatch Synthetics comenzó a rastrear los cambios	CloudWatch Synthetics comenzó a rastrear los cambios para las políticas administradas de AWS.	10 de marzo de 2021

CloudWatchSyntheticsFullAccess

Aquí está el contenido de la política `CloudWatchSyntheticsFullAccess`:

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "synthetics:*"
         ],
         "Resource":"*"
      },
      {
         "Effect":"Allow",
         "Action":[
            "s3:CreateBucket",
            "s3:PutEncryptionConfiguration"
         ],
         "Resource":[
            "arn:aws:s3:::cw-syn-results-*"
         ]
      },
      {
         "Effect":"Allow",
         "Action":[
            "iam:ListRoles",
            "s3:ListAllMyBuckets",
            "s3:GetBucketLocation",
            "xray:GetTraceSummaries",
            "xray:BatchGetTraces",
```

```
            "apigateway:GET"
         ],
         "Resource":"*"
      },
      {
         "Effect":"Allow",
         "Action":[
            "s3:GetObject",
            "s3:ListBucket"
         ],
         "Resource":"arn:aws:s3:::cw-syn-*"
      },
      {
         "Effect":"Allow",
         "Action":[
            "s3:GetObjectVersion"
         ],
         "Resource":"arn:aws:s3:::aws-synthetics-library-*"
      },
      {
         "Effect":"Allow",
         "Action":[
            "iam:PassRole"
         ],
         "Resource":[
            "arn:aws:iam::*:role/service-role/CloudWatchSyntheticsRole*"
         ],
         "Condition":{
            "StringEquals":{
               "iam:PassedToService":[
                  "lambda.amazonaws.com",
                  "synthetics.amazonaws.com"
               ]
            }
         }
      },
      {
         "Effect":"Allow",
         "Action":[
            "iam:GetRole"
         ],
         "Resource":[
            "arn:aws:iam::*:role/service-role/CloudWatchSyntheticsRole*"
         ]
```

```
        },
        {
            "Effect":"Allow",
            "Action":[
                "cloudwatch:GetMetricData",
                "cloudwatch:GetMetricStatistics"
            ],
            "Resource":"*"
        },
        {
            "Effect":"Allow",
            "Action":[
                "cloudwatch:PutMetricAlarm",
                "cloudwatch:DeleteAlarms"
            ],
            "Resource":[
                "arn:aws:cloudwatch:*:*:alarm:Synthetics-*"
            ]
        },
        {
            "Effect":"Allow",
            "Action":[
                "cloudwatch:DescribeAlarms"
            ],
            "Resource":[
                "arn:aws:cloudwatch:*:*:alarm:*"
            ]
        },
        {
            "Effect":"Allow",
            "Action":[
                "lambda:CreateFunction",
                "lambda:AddPermission",
                "lambda:PublishVersion",
                "lambda:UpdateFunctionConfiguration",
                "lambda:GetFunctionConfiguration"
            ],
            "Resource":[
                "arn:aws:lambda:*:*:function:cwsyn-*"
            ]
        },
        {
            "Effect":"Allow",
            "Action":[
```

```
            "lambda:GetLayerVersion",
            "lambda:PublishLayerVersion"
         ],
         "Resource":[
            "arn:aws:lambda:*:*:layer:cwsyn-*",
            "arn:aws:lambda:*:*:layer:Synthetics:*"
         ]
      },
      {
         "Effect":"Allow",
         "Action":[
            "ec2:DescribeVpcs",
            "ec2:DescribeSubnets",
            "ec2:DescribeSecurityGroups"
         ],
         "Resource":[
            "*"
         ]
      },
      {
         "Effect":"Allow",
         "Action":[
            "sns:ListTopics"
         ],
         "Resource":[
            "*"
         ]
      },
      {
         "Effect":"Allow",
         "Action":[
            "sns:CreateTopic",
            "sns:Subscribe",
            "sns:ListSubscriptionsByTopic"
         ],
         "Resource":[
            "arn:*:sns:*:*:Synthetics-*"
         ]
      }
   ]
}
```

CloudWatchSyntheticsReadOnlyAccess

Aquí está el contenido de la política `CloudWatchSyntheticsReadOnlyAccess`:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "synthetics:Describe*",
                "synthetics:Get*",
                "synthetics:List*"
            ],
            "Resource": "*"
        }
    ]
}
```

Limitar a un usuario a ver canaries específicos

Puede limitar la capacidad de un usuario para ver información sobre canaries, de modo que solo pueda ver información sobre los canaries que usted especifique. Para ello, utilice una política de IAM con una declaración de `Condition` similar a la siguiente y asocie esta política a un usuario o a un rol de IAM.

En el ejemplo siguiente se limita al usuario a ver únicamente información sobre `name-of-allowed-canary-1` y `name-of-allowed-canary-2`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "synthetics:DescribeCanaries",
            "Resource": "*",
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "synthetics:Names": [
                        "name-of-allowed-canary-1",
                        "name-of-allowed-canary-2"
                    ]
                }
            }
```

```
        }
    ]
}
```

CloudWatch Synthetics admite mostrar hasta cinco elementos en la matriz `synthetics:Names`.

También puede crear una política que utilice un * como un comodín en nombres de valor controlado que deben permitirse, como en el ejemplo siguiente:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": "synthetics:DescribeCanaries",
            "Resource": "*",
            "Condition": {
                "ForAnyValue:StringLike": {
                    "synthetics:Names": [
                        "my-team-canary-*"
                    ]
                }
            }
        }
    ]
}
```

Cualquier usuario que haya iniciado sesión con una de estas políticas asociadas no puede utilizar la consola de CloudWatch para ver ninguna información de valor controlado. Solo pueden ver la información de valor controlado para los valores controlados autorizados por la política y exclusivamente mediante el uso de la API DescribeCanaries o el comando describe-canaries en la AWS CLI.

Creación de un valor controlado

> **Important**
>
> Asegúrese de utilizar canaries de Synthetics para supervisar solo aquellos puntos de enlace y API en los que tenga propiedad o permisos. En función de la configuración de frecuencia

de los valores controlados, estos puntos de conexión pueden experimentar un aumento del tráfico.

Cuando utiliza la consola de CloudWatch para crear un valor controlado, puede utilizar un esquema que CloudWatch proporciona para crear un valor controlado o escribir su propio script. Para obtener más información, consulte Uso de esquemas de valores controlados.

También puede crear un valor controlado mediante la AWS CloudFormation si está utilizando su propio script para el valor controlado. Para obtener más información, consulte AWS::Synthetics::Canary en la Guía del usuario de AWS CloudFormation.

Si está escribiendo su propio script, puede utilizar varias funciones que CloudWatch Synthetics ha integrado en una biblioteca. Para obtener más información, consulte Versiones de tiempo de ejecución de Synthetics.

> **Note**
>
> Cuando crea un valor controlado, una de las capas que se crean es una capa de Synthetics a la que se le antepone `Synthetics`. Esta capa es propiedad de la cuenta del servicio Synthetics y contiene el código del tiempo de ejecución.

Cómo crear un valor controlado

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Señales de aplicación, Valores controlados de Synthetics.
3. Elija Crear valor controlado.
4. Seleccione una de las siguientes opciones:
   - Para basar su valor controlado en un script de proyecto, elija Usar un esquema, y, a continuación, elija el tipo de valor controlado que desea crear. Para obtener más información acerca de lo que hace cada tipo de proyecto, consulte Uso de esquemas de valores controlados.
   - Para cargar su propio script de Node.js y crear un valor controlado personalizado, elija Cargar un script.

A continuación, puede arrastrar su script al área Script o elegir Browse files (Examinar archivos) para desplazarse hasta el script en su sistema de archivos.

- Para importar el script desde un bucket de S3, elija Import from S3 (Importar de S3). En Ubicación de origen, escriba la ruta completa al valor controlado o elija Examinar S3.

  Debe tener permisos `s3:GetObject` y `s3:GetObjectVersion` para el bucket de S3 que utilice. El bucket debe estar en la misma Región de AWS en la que está creando el valor controlado.

5. En Nombre, escriba un nombre para su valor controlado. El nombre se utiliza en muchas páginas, por lo que le recomendamos que le asigne un nombre descriptivo que lo diferencie de otros canaries.
6. En URL del punto de conexión o aplicación, escriba la dirección URL que desea que pruebe el valor controlado. Esta URL debe incluir el protocolo (como https://).

   Si desea que el valor controlado pruebe un punto de conexión en una VPC, también debe escribir información sobre la VPC más adelante en este procedimiento.
7. Si utiliza su propio script para el valor controlado, en Controlador de Lambda, escriba el punto de entrada donde desea que comience el valor controlado. Si utiliza un tiempo de ejecución anterior a `syn-nodejs-puppeteer-3.4` o `syn-python-selenium-1.1`, la cadena que ingrese debe terminar en `.handler`. Si utiliza `syn-nodejs-puppeteer-3.4`, `syn-python-selenium-1.1` o un tiempo de ejecución posterior, no se aplica esta restricción.
8. Si está utilizando variables de entorno en el script, elija Environment variables (Variables de entorno) y, a continuación, especifique un valor para cada variable de entorno definida en el script. Para obtener más información, consulte Variables de entorno.
9. Bajo el título Programa, elija si desea ejecutar el valor controlado solo una vez, ejecutarlo continuamente con una expresión de frecuencia o programarlo con una expresión cron.

   - Cuando utilice la consola de CloudWatch para crear un valor controlado que funcione continuamente, puede elegir un índice entre una vez al minuto y una vez por hora.

   - Para obtener más información sobre cómo escribir una expresión cron para la programación de valores controlados, consulte Programación de las ejecuciones de valores controlados con cron.
10. (Opcional) Para establecer un tiempo de espera para el valor controlado, elija Configuración adicional y, a continuación, especifique el valor de tiempo de espera. Haga que no sea inferior

a 15 segundos para permitir arranques en frío de Lambda y el tiempo que tarda en arrancar la instrumentación del valor controlado.

11. En Retención de datos, especifique cuánto tiempo se retiene la información sobre las ejecuciones de valores controlados fallidas y correctas. El intervalo abarca de 1 a 455 días.

    Esta configuración afecta solo a los datos que CloudWatch Synthetics almacena y muestra en la consola. No afecta a los datos almacenados en los buckets de Amazon S3, ni a los registros o métricas publicados por el valor controlado.

12. En Almacenamiento de datos, seleccione el bucket de S3 que desea utilizar para almacenar los datos de las ejecuciones de valores controlados. El nombre del bucket no puede contener un punto (.). Si deja este valor en blanco, se utilizará un bucket de S3 predeterminado o se creará uno nuevo.

    Si utiliza `syn-nodejs-puppeteer-3.0` o un tiempo de ejecución posterior, al introducir la dirección URL del bucket en el cuadro de texto, puede especificar un bucket en la Región actual o en otra región. Si utiliza una versión anterior del tiempo de ejecución, el bucket debe estar en la región actual.

13. (Opcional) De forma predeterminada, los canaries almacenan sus artefactos en Amazon S3 y los artefactos se cifran en reposo mediante una clave de AWS KMS administrada por AWS. Puede utilizar una opción de cifrado diferente si selecciona Additional configuration (Configuración adicional) en la sección Data Storage (Almacenamiento de datos). Luego, puede elegir el tipo de clave que desea utilizar para el cifrado. Para obtener más información, consulte Cifrado de artefactos de un valor controlado.

14. En Permisos de acceso, elija si desea crear un nuevo rol de IAM para ejecutar el valor controlado o utilizar uno existente.

    Si hace que CloudWatch Synthetics cree el rol, este incluirá automáticamente todos los permisos necesarios. Si desea crear el rol, consulte Roles y permisos necesarios para los valores controlados para obtener información acerca de los permisos necesarios.

    Si utiliza la consola de CloudWatch para crear un rol para un valor controlado, al crear el valor controlado no podrá volver a utilizar el rol para otros valores controlados, ya que estos roles son específicos de un solo valor controlado. Si ha creado manualmente un rol que funcione para varios canaries, puede utilizar ese rol existente.

    Para utilizar un rol existente, debe tener el permiso `iam:PassRole` para pasar ese rol a Synthetics y Lambda. También debe tener el permiso `iam:GetRole`.

15. (Opcional) Bajo el título Alarmas, elija si desea que se creen alarmas predeterminadas de CloudWatch para este valor controlado. Si decide crear alarmas, se crean con la siguiente convención de nombres: `Synthetics-Alarm-`*`canaryName`*`-`*`index`*

    `index` es un número que representa cada alarma que se crea para este valor controlado. La primera alarma tiene un índice de 1, la segunda alarma tiene un índice de 2, y así sucesivamente.

16. (Opcional) Para que este valor controlado pruebe un punto de conexión que está en una VPC, elija Configuración de VPC y, a continuación, haga lo siguiente:

    a. Seleccione la VPC que aloja el punto de enlace.

    b. Seleccione una o más subredes en la VPC. Debe seleccionar una subred privada, ya que la instancia de Lambda no se puede configurar para ejecutarse en una subred pública cuando no se puede asignar una dirección IP a la instancia de Lambda durante la ejecución. Para obtener más información, consulte Configuración de una función de Lambda para obtener acceso a los recursos en una VPC.

    c. Seleccione uno o varios grupos de seguridad en la VPC.

    Si el punto de conexión está en una VPC, debe habilitar el valor controlado para enviar información a CloudWatch y a Amazon S3. Para obtener más información, consulte Ejecución de un valor controlado en una VPC.

17. (Opcional) En Etiquetas, agregue uno o más pares de clave-valor como etiquetas para este valor controlado. Las etiquetas pueden ayudarle a identificar y organizar sus recursos de AWS y a realizar un seguimiento de sus costos de AWS. Para obtener más información, consulte Etiquetado de los recursos de Amazon CloudWatch.

18. (Opcional) Bajo el título Rastreo activo, elija si desea habilitar el rastreo activo de X-Ray para este valor controlado. Esta opción solo está disponible si el valor controlado utiliza la versión de tiempo de ejecución syn-nodejs-2.0 o una posterior. Para obtener más información, consulte Canaries y rastreo X-Ray.

Recursos que se crean para canaries

Al crear un valor controlado, se crean los siguientes recursos para el mismo:

- Un rol de IAM con el nombre `CloudWatchSyntheticsRole-`*`canary-name`*`-`*`uuid`* (si utiliza la consola de CloudWatch para crear el valor controlado y especifica que se cree un nuevo rol para el valor controlado)
- Una política de IAM con el nombre `CloudWatchSyntheticsPolicy-`*`canary-name`*`-`*`uuid`*.
- Un bucket de S3 con el nombre `cw-syn-results-`*`accountID`*`-`*`region`*.
- Alarmas con el nombre `Synthetics-Alarm-`*`MyCanaryName`*, si desea que se creen alarmas para el valor controlado.
- Capas y funciones de Lambda, si utiliza un esquema para crear el valor controlado. Estos recursos tienen el prefijo `cwsyn-`*`MyCanaryName`*.
- Grupos de registro de CloudWatch Logs con el nombre `/aws/lambda/cwsyn-`*`MyCanaryName`*`-`*`randomId`*.

Uso de esquemas de valores controlados

Esta sección proporciona detalles sobre cada uno de los esquemas de valor controlado y las tareas para las que cada esquema es más adecuado. Se proporcionan esquemas para los siguientes tipos de valor controlado:

- Monitor de latidos
- Valor controlado de la API
- Verificador de enlaces que no funcionan
- Supervisión visual
- Registrador de valores controlados
- Flujo de trabajo de la GUI

Cuando se utiliza un esquema para crear un valor controlado, mientras rellena los campos en la consola de CloudWatch, el área de la página Editor de scripts muestra el valor controlado que se está creando como script Node.js. También puede editar su valor controlado en esta área para personalizarlo aún más.

Supervisión de latidos

Los scripts de latidos cargan la URL especificada y almacenan una captura de pantalla de la página y un archivo HTTP (archivo HAR). También almacenan registros de direcciones URL a las que se accede.

Puede utilizar los archivos HAR para ver datos de rendimiento detallados sobre las páginas web. Puede analizar la lista de solicitudes web y detectar problemas de rendimiento, como el tiempo de carga de un elemento.

Si el valor controlado utiliza `syn-nodejs-puppeteer-3.1` o una versión posterior en el tiempo de ejecución, puede utilizar el esquema de supervisión de latidos para supervisar varias direcciones URL y ver el estado, la duración, las capturas de pantalla asociadas y el motivo del error para cada URL en el resumen de pasos del informe de ejecución del valor controlado.

Valor controlado de la API

Los canaries de API pueden probar las funciones básicas de lectura y escritura de una API REST. REST significa transferencia de estado representacional y es un conjunto de reglas que siguen los desarrolladores al crear una API. Una de estas reglas establece que un enlace a una URL específica debe devolver un fragmento de datos.

Los canaries pueden trabajar con cualquier API y probar todo tipo de funcionalidades. Cada valor controlado puede hacer varias llamadas a la API.

En los canarios que usan la versión de tiempo de ejecución `syn-nodejs-2.2` o posterior, el esquema de canario de API admite canarios de varios pasos que supervisan las API como pasos HTTP. Puede probar varias API en un único valor controlado. Cada paso es una solicitud independiente que puede acceder a una URL diferente, usar cabeceras diferentes y utilizar diferentes reglas para la captura de cabeceras y de cuerpos de respuesta. Al no capturar cabeceras y cuerpo de respuesta, puede evitar que se registre información confidencial.

Cada solicitud en un valor controlado de API consta de la siguiente información:

- El punto de enlace, que es la URL que solicita.
- El método, que es el tipo de solicitud que se envía al servidor. Las API REST admiten operaciones GET (lectura), POST (escritura), PUT (actualización), PATCH (actualización) y DELETE (eliminación).
- Los encabezados, que proporcionan información tanto al cliente como al servidor. Se utilizan para la autenticación y para proporcionar información sobre el contenido del cuerpo. Para obtener una lista de cabeceras válidos, consulte HTTP Headers (Cabeceras HTTP).
- Los datos (o el cuerpo), contienen información que se enviará al servidor. Esto se utiliza solo para solicitudes POST, PUT, PATCH o DELETE.

El esquema de valor controlado de API admite los métodos GET y POST. Cuando utilice este valor controlado, debe especificar los encabezados. Por ejemplo, puede especificar **`Authorization`** como Key (Clave) y especificar los datos de autorización necesarios como Value (Valor) para esa clave.

Si está probando una solicitud POST, especifique también el contenido que se va a publicar en el campo Data (Datos).

Integración con la API Gateway

El proyecto de la API está integrado con Amazon API Gateway. Esto le permite seleccionar una API de API Gateway y un escenario desde la misma cuenta y región de AWS que el valor controlado, o cargar una plantilla de Swagger desde API Gateway para la supervisión de la API entre cuentas y entre regiones. A continuación, puede elegir el resto de los detalles en la consola para crear el valor controlado, en lugar de introducirlos desde cero. Para obtener más información sobre API Gateway, consulte What is Amazon API Gateway? (¿Qué es Amazon API Gateway?)

Uso de una API privada

Puede crear un valor controlado que utilice una API privada en Amazon API Gateway. Para obtener más información, consulte Creación de una API privada en Amazon API Gateway.

Verificador de enlaces que no funcionan

El verificador de enlaces que no funcionan recopila todos los enlaces dentro de la dirección URL que está probando mediante `document.getElementsByTagName('a')`. Solo prueba el número de enlaces que especifique; la dirección URL en sí se cuenta como primer enlace. Por ejemplo, si desea comprobar todos los enlaces de una página que contiene cinco enlaces, debe especificar que el valor controlado siga seis enlaces.

Los valores controlados de verificadores de enlaces que no funcionan que se han creado con el tiempo de ejecución `syn-nodejs-2.0-beta` o posterior admiten las siguientes características adicionales:

- Proporciona un informe que incluye los enlaces que se han verificado, el código de estado, el motivo del error (si los hay) y las capturas de pantalla de la página fuente y de destino.
- Al ver los resultados del valor controlado, puede utilizar el filtro para ver solo los enlaces que no funcionan y, a continuación, corregir el enlace en función del motivo del error.
- Esta versión toma capturas de pantalla de la página fuente anotadas para cada enlace y resalta el anclaje donde se encontró el enlace. Los componentes ocultos no se anotan.

- Puede configurar esta versión para tomar capturas de pantalla de páginas fuente y de destino, pero solo páginas fuente o páginas de destino.
- Esta versión corrige un problema de la versión anterior en el que el script del valor controlado se detiene después del primer enlace roto, incluso cuando se raspan más enlaces desde la primera página.

Si desea actualizar un valor controlado existente con `syn-1.0` para utilizar el nuevo tiempo de ejecución, debe eliminar y volver a crear el valor controlado. Actualizar un valor controlado existente al nuevo tiempo de ejecución no hace que estas características estén disponibles.

Un valor controlado de verificadores de enlaces que no funcionan detecta los siguientes tipos de errores en un enlace:

- 404 Página no encontrada
- Nombre de host no válido
- URL incorrecta. Por ejemplo, a la URL le falta un corchete, tiene barras diagonales adicionales o utiliza el protocolo incorrecto.
- Código de respuesta HTTP no válido
- El servidor host devuelve respuestas vacías sin contenido ni código de respuesta.
- Las solicitudes HTTP agotan constantemente el tiempo de espera durante la ejecución del valor controlado.
- El host interrumpe constantemente las conexiones porque está mal configurado o está demasiado ocupado.

Proyecto de supervisión visual

El esquema de supervisión visual incluye un código para comparar capturas de pantalla que se toman durante una ejecución de valor controlado con capturas de pantalla que se toman durante una ejecución de valor controlado de línea de base. Si la discrepancia entre las dos capturas de pantalla está más allá de un porcentaje umbral, el valor controlado falla. La supervisión visual es compatible en canaries que ejecutan syn-puppeteer-node-3.2 y versiones posteriores. Actualmente no es compatible con canaries que ejecutan Python y Selenium.

El esquema de supervisión visual incluye la siguiente línea de código en el script de valor controlado de esquema predeterminado, que permite la supervisión visual.

```
syntheticsConfiguration.withVisualCompareWithBaseRun(true);
```

La primera vez que el valor controlado se ejecuta correctamente después de agregar esta línea al script, utiliza las capturas de pantalla que se toman durante esa ejecución como línea de base para la comparación. Después de la primera ejecución del valor controlado, se puede usar la consola de CloudWatch para editar el valor controlado para realizar cualquiera de las siguientes acciones:

- Establecer la siguiente ejecución del valor controlado como la nueva línea de base.
- Establecer límites en la captura de pantalla de línea de base actual para designar áreas de la captura de pantalla que se ignorarán durante las comparaciones visuales.
- Eliminar una captura de pantalla de ser utilizada para la supervisión visual.

Para obtener más información sobre cómo usar la consola de CloudWatch para editar un valor controlado, consulte Edición o eliminación de un valor controlado.

También puede cambiar la ejecución del valor controlado que se utiliza como línea de base mediante los parámetros `nextrun` o `lastrun` o mediante la especificación de un ID de ejecución de valor controlado en la API UpdateCanary.

Cuando utilice el proyecto de supervisión visual, ingrese la dirección URL donde desea que se tome la captura de pantalla y especifique un umbral de diferencia como porcentaje. Después de la ejecución de la línea de base, las futuras ejecuciones del valor controlado que detectan una diferencia visual mayor que ese umbral desencadenan un fallo del valor controlado. Después de la ejecución de la línea de base, también puede editar el valor controlado para “dibujar” límites en la captura de pantalla de línea de base que desea omitir durante la supervisión visual.

La característica de supervisión visual está impulsada por el conjunto de herramientas de software de código abierto ImageMagick. Para obtener más información, consulte ImageMagick.

Registrador de valores controlados

Con el esquema del registrador de valores controlados, puede utilizar CloudWatch Synthetics Recorder para registrar las acciones de cliqueo y tipeo en un sitio web y generar automáticamente un script Node.js que se puede utilizar para crear un valor controlado que siga los mismos pasos. Recorder CloudWatch Synthetics es una extensión de Amazon para Google Chrome.

Créditos: (Créditos): el registrador CloudWatch Synthetics se basa en el Headless recorder (Registrador sin procesador).

Para obtener más información, consulte Uso del registrador de CloudWatch Synthetics para Google Chrome.

Generador de flujo de trabajo de la GUI

El proyecto del generador de flujo de trabajo de la GUI verifica que se pueden realizar acciones en la página web. Por ejemplo, si tiene una página web con un formulario de inicio de sesión, el valor controlado puede rellenar los campos de usuario y contraseña y enviar el formulario para verificar que la página web funciona correctamente.

Cuando utiliza un esquema para crear este tipo de valor controlado, especifique las acciones que desea que el valor controlado realice en la página web. Las acciones que puede utilizar son las siguientes:

- Clic: selecciona el elemento especificado y simula un usuario al hacer clic o al elegir el elemento.

  Para especificar el elemento en un script Node.js, utilice `[id=]` o `a[class=]`.

  Para especificar el elemento en un script Python, utilice `xpath //*[@id=]` o `//*[@class=]`.
- Verificación del selector: verifica que el elemento especificado existe en la página web. Esta prueba es útil para comprobar que una acción anterior hace que los elementos correctos rellenen la página.

  Para especificar el elemento que se va a verificar en un script Node.js, utilice `[id=]` o `a[class=]`.

  Para especificar el elemento que se va a verificar en un script Python, utilice `xpath //*[@id=]` o `//*[class=]`.
- Verificación de texto: verifica que la cadena especificada está contenida en el elemento de destino. Esta prueba es útil para verificar que una acción anterior ha causado que se muestre el texto correcto.

  Para especificar el elemento en un script Node.js, utilice un formato como `div[@id=]//h1`, ya que esta acción utiliza la función `waitForXPath` en Puppeteer.

  Para especificar el elemento en un script Python, utilice el formato xpath como `//*[@id=]` o //*[@class =] porque esta acción utiliza la función `implicitly_wait` en Selenium.
- Ingresar texto: ingresa el texto especificado en el elemento de destino.

Para especificar el elemento que se va a verificar en un script Node.js, utilice `[id=]` o `a[class=]`.

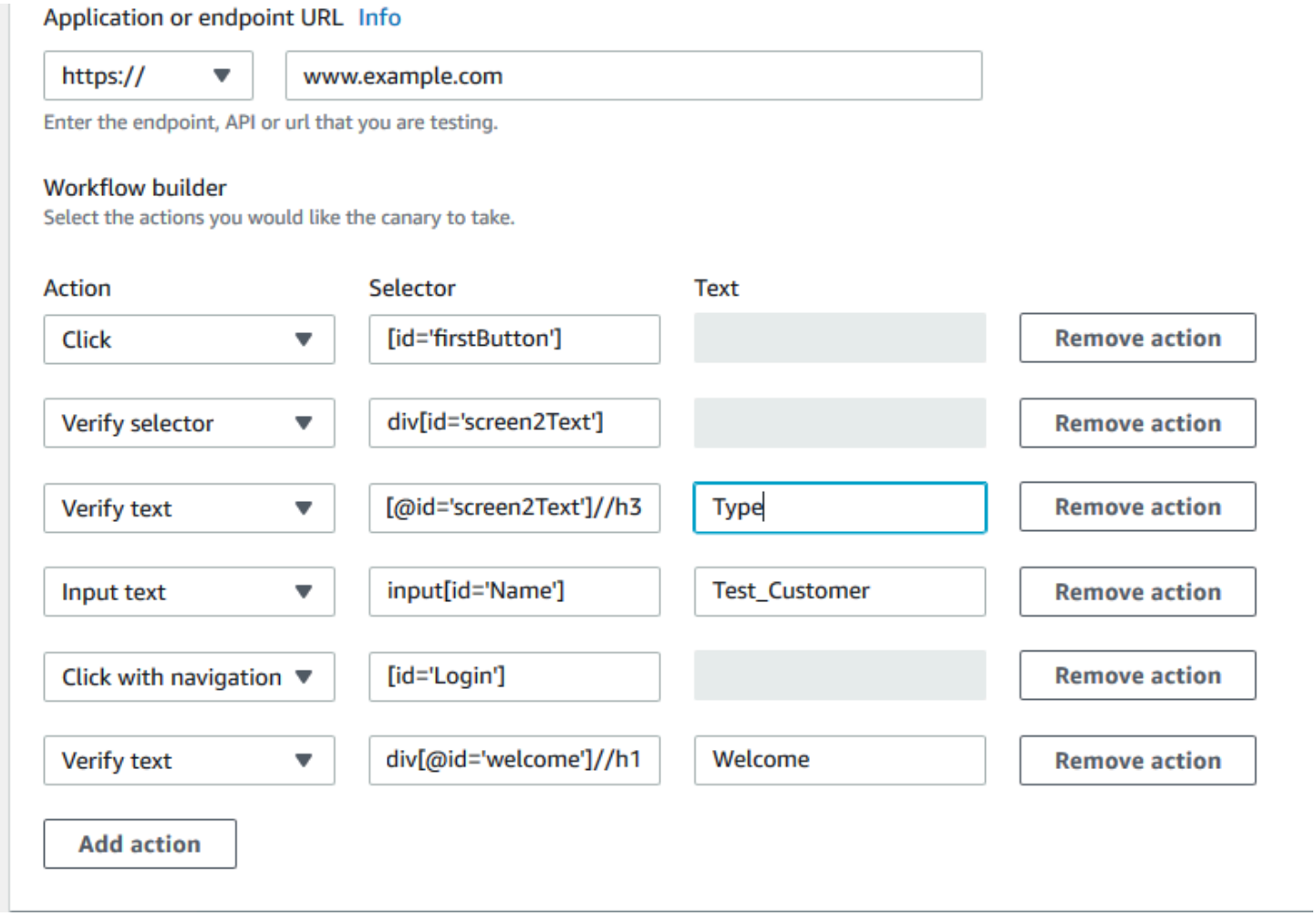
Para especificar el elemento que se va a verificar en un script Python, utilice `xpath //*[@id=]` o `//*[@class=]`.

- Clic con navegación: espera a que toda la página se cargue después de elegir el elemento especificado. Esto es más útil cuando necesita volver a cargar la página.

  Para especificar el elemento en un script Node.js, utilice `[id=]` o `a[class=]`.

  Para especificar el elemento en un script Python, utilice `xpath //*[@id=]` o `//*[@class=]`.

Por ejemplo, el siguiente blueprint utiliza Node.js. Hace clic en firstButton en la URL especificada, comprueba que aparece el selector esperado con el texto esperado, ingresa el nombre `Test_Customer` en el campo Name (Nombre), hace clic en el botón Login (Inicio de sesión) y, luego, comprueba que el inicio de sesión sea complete correctamente mediante la verificación del texto Welcome (Bienvenida) en la página siguiente.



Los valores controlados de flujo de trabajo de la GUI que utilizan los siguientes tiempos de ejecución también proporcionan un resumen de los pasos ejecutados para cada ejecución de un valor controlado. Puede utilizar las capturas de pantalla y el mensaje de error asociados con cada paso para encontrar la causa raíz del error.

- `syn-nodejs-2.0` o posterior.
- `syn-python-selenium-1.0` o posterior.

Uso del registrador de CloudWatch Synthetics para Google Chrome

Amazon proporciona un registrador de CloudWatch Synthetics para ayudarle a crear canaries de manera más fácil. El registrador es una extensión de Google Chrome.

Registra las acciones de cliqueo y de tipeo en un sitio web y genera automáticamente un script Node.js que se puede utilizar para crear un valor controlado que siga los mismos pasos.

Después de iniciar el registro, CloudWatch Synthetics Recorder detecta las acciones en el navegador y las convierte en un script. Puede pausar y reanudar el registro según sea necesario. Cuando deja de registrar, el registrador produce un script Node.js de las acciones, que puede copiar fácilmente con el comando Copy to Clipboard (Copiar al portapapeles). A continuación, puede utilizar el script para crear un valor controlado en CloudWatch Synthetics.

Credits (Créditos): el registrador CloudWatch Synthetics se basa en el Headless recorder (Registrador sin procesador).

Instalación de la extensión CloudWatch Synthetics Recorder para Google Chrome

Si desea utilizar CloudWatch Synthetics Recorder, puede empezar a crear un valor controlado y elegir el esquema Registrador de valores controlados. Si lo hace cuando aún no ha descargado el registrador, la consola de CloudWatch Synthetics proporciona un enlace para descargarlo.

Alternativamente, puede seguir estos pasos para descargar e instalar el registrador directamente.

Para instalar el registrador de CloudWatch Synthetics

1. Con Google Chrome, ingrese a este sitio web: https://chrome.google.com/webstore/detail/cloudwatch-synthetics-rec/bhdnlmmgiplmbcdmkkdfplenecpegfno
2. Seleccione Add to Chrome (Añadir a Chrome) y, después, Add extension (Agregar extensión).

Uso del registrador de CloudWatch Synthetics para Google Chrome

Si desea utilizar CloudWatch Synthetics Recorder para obtener ayuda para crear un valor controlado, puede elegir Crear valor controlado en la consola de CloudWatch y luego, Usar un esquema, Registrador de valores controlados. Para obtener más información, consulte Creación de un valor controlado.

Alternativamente, puede usar el registrador para registrar pasos sin usarlos de manera inmediata para crear un valor controlado.

Para utilizar CloudWatch Synthetics Recorder a fin de registrar las acciones en un sitio web

1. Desplácese hasta la página que desea supervisar.
2. Elija el icono de extensiones de Chrome y, a continuación, elija CloudWatch Synthetics Recorder.
3. Seleccione Start Recording (Comenzar a registrar).

4. Realice los pasos que desea registrar. Para pausar el registro, elija Pause (Pausar).
5. Cuando termine de registrar el flujo de trabajo, elija Stop recording (Detener registro).
6. Seleccione Copy to clipboard (Copiar al portapapeles) para copiar el script generado al portapapeles. O bien, si desea comenzar de nuevo, elija New recording (Registro nuevo).
7. Para crear un valor controlado con el script copiado, puede pegar el script copiado en el editor en línea del esquema del registro o guardarlo en un bucket de Amazon S3 e importarlo desde allí.
8. Si no está creando un valor controlado inmediatamente, puede guardar el script registrado en un archivo.

Limitaciones conocidas CloudWatch Synthetics Recorder

CloudWatch Synthetics Recorder para Google Chrome tiene las siguientes limitaciones por el momento.

- Los elementos HTML que no tienen ID usarán selectores CSS. Esto puede dañar los canaries si la estructura de la página web cambia más adelante. Se planea proporcionar algunas opciones de configuración (como el uso del atributo data-id) en torno a esto en una versión futura del registrador.
- El registrador no admite acciones, como hacer doble clic o copiar o pegar, y no admite combinaciones de teclas como CMD+0.
- Para verificar la presencia de un elemento o texto en la página, los usuarios deben agregar aserciones después de generar el script. El registrador no admite la verificación de un elemento sin realizar ninguna acción sobre ese elemento. Esto es similar a las opciones “Verificar texto” o “Verificar elemento” en el generador de flujo de trabajo de valores controlados. Se planea agregar compatibilidad con aserciones en una versión futura del registrador.
- El registrador graba todas las acciones en la pestaña donde se inicia el registro. No registra ventanas emergentes (por ejemplo, para permitir el seguimiento de ubicación) ni la navegación a páginas diferentes desde ventanas emergentes.

Versiones de tiempo de ejecución de Synthetics

Al crear o actualizar un valor controlado, se elige una versión de tiempo de ejecución de Synthetics para este. Un tiempo de ejecución de Synthetics es una combinación del código de Synthetics que llama al controlador de scripts y de las capas Lambda de las dependencias agrupadas.

CloudWatch Synthetics admite actualmente los tiempos de ejecución que utilizan Node.js para scripts y el marco de Puppeteer, y los tiempos de ejecución que utilizan Python para scripting y Selenium Webdriver para el marco.

Recomendamos utilizar siempre la versión de tiempo de ejecución más reciente para sus valores controlados a fin de poder usar las últimas características y actualizaciones realizadas en la biblioteca de Synthetics.

Cuando crea un valor controlado, una de las capas que se crean es una capa de Synthetics a la que se le antepone `Synthetics`. Esta capa es propiedad de la cuenta del servicio Synthetics y contiene el código del tiempo de ejecución.

> **Note**
>
> Siempre que actualice un valor controlado para utilizar una nueva versión del tiempo de ejecución de Synthetics, todas las funciones de la biblioteca Synthetics que utilice su valor controlado también se actualizarán automáticamente a la misma versión de NodeJS que admita el tiempo de ejecución de Synthetics.

Temas

Política de soporte de tiempo de ejecución de CloudWatch Synthetics

Las versiones de tiempo de ejecución de Synthetics están sujetas a operaciones de mantenimiento y actualizaciones de seguridad. Cuando cualquier componente de una versión de tiempo de ejecución deja de ser compatible, dicha versión de tiempo de ejecución de Synthetics queda obsoleta.

No puede crear canaries nuevos con versiones de tiempo de ejecución obsoletas. Los Canaries que utilizan tiempos de ejecución obsoletos siguen ejecutándose. Puede detener, iniciar y eliminar estos canaries. Para actualizar un valor controlado existente que utiliza versiones de tiempo de ejecución obsoletas, actualícelo de forma que utilice una versión de tiempo de ejecución compatible.

CloudWatch Synthetics le notifica por email si tiene canaries que utilizan tiempos de ejecución programados para quedar obsoletos en los próximos 60 días. Se recomienda que migre los canaries

a una versión de tiempo de ejecución compatible para beneficiarse de las nuevas mejoras de funcionalidad, seguridad y de rendimiento que se incluyen en las versiones más recientes.

¿Cómo se actualiza un valor controlado a una nueva versión en tiempo de ejecución?

Se puede actualizar la versión en tiempo de ejecución de un valor controlado mediante la consola de CloudWatch, AWS CloudFormation, AWS CLI o el SDK de AWS. Cuando utilice la consola de CloudWatch, puede actualizar hasta cinco valores controlados a la vez si los selecciona en la página de la lista de valores controlados y, a continuación, elije Acciones, Actualizar tiempo de ejecución.

Se puede verificar la actualización al clonar primero el valor controlado mediante la consola de CloudWatch y si se actualiza la versión de tiempo de ejecución. Esto crea otro valor controlado que es un clon del valor controlado original. Una vez que haya verificado el valor controlado con la nueva versión de tiempo de ejecución, puede actualizar la versión de tiempo de ejecución del valor controlado original y eliminar el valor controlado clon.

También puede actualizar varios canaries si se utiliza un script de actualización. Para obtener más información, consulte Script de actualización en tiempo de ejecución de valores controlados.

Si se actualiza un valor controlado y falla, consulte Solución de problemas de un valor controlado.

Fechas de caducidad de los tiempos de ejecución

Versión de tiempo de ejecución	Fecha de caducidad
`syn-nodejs-puppeteer-6.1`	8 de marzo de 2024
`syn-nodejs-puppeteer-6.0`	8 de marzo de 2024
`syn-nodejs-puppeteer-5.1`	8 de marzo de 2024
`syn-nodejs-puppeteer-5.0`	8 de marzo de 2024
`syn-nodejs-puppeteer-4.0`	8 de marzo de 2024

Versión de tiempo de ejecución	Fecha de caducidad
`syn-nodejs-puppeteer-3.9`	8 de enero de 2024
`syn-nodejs-puppeteer-3.8`	8 de enero de 2024
`syn-python-selenium-2.0`	8 de marzo de 2024
`syn-python-selenium-1.3`	8 de marzo de 2024
`syn-python-selenium-1.2`	8 de marzo de 2024
`syn-python-selenium-1.1`	8 de marzo de 2024
`syn-python-selenium-1.0`	8 de marzo de 2024
`syn-nodejs-puppeteer-3.7`	8 de enero de 2024
`syn-nodejs-puppeteer-3.6`	8 de enero de 2024
`syn-nodejs-puppeteer-3.5`	8 de enero de 2024
`syn-nodejs-puppeteer-3.4`	13 de noviembre de 2022
`syn-nodejs-puppeteer-3.3`	13 de noviembre de 2022

Versión de tiempo de ejecución	Fecha de caducidad
`syn-nodejs-puppeteer-3.2`	13 de noviembre de 2022
`syn-nodejs-puppeteer-3.1`	13 de noviembre de 2022
`syn-nodejs-puppeteer-3.0`	13 de noviembre de 2022
`syn-nodejs-2.2`	28 de mayo de 2021
`syn-nodejs-2.1`	28 de mayo de 2021
`syn-nodejs-2.0`	28 de mayo de 2021
`syn-nodejs-2.0-beta`	8 de febrero de 2021
`syn-1.0`	28 de mayo de 2021

Script de actualización en tiempo de ejecución de valores controlados

Para actualizar un script de valor controlado a una versión compatible de tiempo de ejecución, utilice el siguiente script.

```
const AWS = require('aws-sdk');

// You need to configure your AWS credentials and Region.
//   https://docs.aws.amazon.com/sdk-for-javascript/v3/developer-guide/setting-
credentials-node.html
//   https://docs.aws.amazon.com/sdk-for-javascript/v3/developer-guide/setting-
region.html

const synthetics = new AWS.Synthetics();

const DEFAULT_OPTIONS = {
  /**
   * The number of canaries to upgrade during a single run of this script.
```

```
   */
  count: 10,
  /**
   * No canaries are upgraded unless force is specified.
   */
  force: false
};

/**
 * The number of milliseconds to sleep between GetCanary calls when
 * verifying that an update succeeded.
 */
const SLEEP_TIME = 5000;

(async () => {
  try {
    const options = getOptions();

    const versions = await getRuntimeVersions();
    const canaries = await getAllCanaries();
    const upgrades = canaries
      .filter(canary => !versions.isLatestVersion(canary.RuntimeVersion))
      .map(canary => {
        return {
          Name: canary.Name,
          FromVersion: canary.RuntimeVersion,
          ToVersion: versions.getLatestVersion(canary.RuntimeVersion)
        };
      });

    if (options.force) {
      const promises = [];

      for (const upgrade of upgrades.slice(0, options.count)) {
        const promise = upgradeCanary(upgrade);
        promises.push(promise);
        // Sleep for 100 milliseconds to avoid throttling.
        await usleep(100);
      }

      const succeeded = [];
      const failed = [];
      for (let i = 0; i < upgrades.slice(0, options.count).length; i++) {
        const upgrade = upgrades[i];
```

```
        const promise = promises[i];
        try {
          await promise;
          console.log(`The update of ${upgrade.Name} succeeded.`);
          succeeded.push(upgrade.Name);
        } catch (e) {
          console.log(`The update of ${upgrade.Name} failed with error: ${e}`);
          failed.push({
            Name: upgrade.Name,
            Reason: e
          });
        }
      }

      if (succeeded.length) {
        console.group('The following canaries were upgraded successfully.');
        for (const name of succeeded) {
          console.log(name);
        }
        console.groupEnd()
      } else {
        console.log('No canaries were upgraded successfully.');
      }

      if (failed.length) {
        console.group('The following canaries were not upgraded successfully.');
        for (const failure of failed) {
          console.log('\x1b[31m', `${failure.Name}: ${failure.Reason}`, '\x1b[0m');
        }
        console.groupEnd();
      }
    } else {
      console.log('Run with --force [--count <count>] to perform the first <count>
 upgrades shown. The default value of <count> is 10.')
      console.table(upgrades);
    }
  } catch (e) {
    console.error(e);
  }
})();

function getOptions() {
  const force = getFlag('--force', DEFAULT_OPTIONS.force);
  const count = getOption('--count', DEFAULT_OPTIONS.count);
```

```
  return { force, count };

  function getFlag(key, defaultValue) {
    return process.argv.includes(key) || defaultValue;
  }
  function getOption(key, defaultValue) {
    const index = process.argv.indexOf(key);
    if (index < 0) {
      return defaultValue;
    }
    const value = process.argv[index + 1];
    if (typeof value === 'undefined' || value.startsWith('-')) {
      throw `The ${key} option requires a value.`;
    }
    return value;
  }
}

function getAllCanaries() {
  return new Promise((resolve, reject) => {
    const canaries = [];

    synthetics.describeCanaries().eachPage((err, data) => {
      if (err) {
        reject(err);
      } else {
        if (data === null) {
          resolve(canaries);
        } else {
          canaries.push(...data.Canaries);
        }
      }
    });
  });
}

function getRuntimeVersions() {
  return new Promise((resolve, reject) => {
    const jsVersions = [];
    const pythonVersions = [];
    synthetics.describeRuntimeVersions().eachPage((err, data) => {
      if (err) {
        reject(err);
      } else {
```

```
        if (data === null) {
          jsVersions.sort((a, b) => a.ReleaseDate - b.ReleaseDate);
          pythonVersions.sort((a, b) => a.ReleaseDate - b.ReleaseDate);
          resolve({
            isLatestVersion(version) {
              const latest = this.getLatestVersion(version);
              return latest === version;
            },
            getLatestVersion(version) {
              if (jsVersions.some(v => v.VersionName === version)) {
                return jsVersions[jsVersions.length - 1].VersionName;
              } else if (pythonVersions.some(v => v.VersionName === version)) {
                return pythonVersions[pythonVersions.length - 1].VersionName;
              } else {
                throw Error(`Unknown version ${version}`);
              }
            }
          });
        } else {
          for (const version of data.RuntimeVersions) {
            if (version.VersionName === 'syn-1.0') {
              jsVersions.push(version);
            } else if (version.VersionName.startsWith('syn-nodejs-2.')) {
              jsVersions.push(version);
            } else if (version.VersionName.startsWith('syn-nodejs-puppeteer-')) {
              jsVersions.push(version);
            } else if (version.VersionName.startsWith('syn-python-selenium-')) {
              pythonVersions.push(version);
            } else {
              throw Error(`Unknown version ${version.VersionName}`);
            }
          }
        }
      }
    });
  });
}

async function upgradeCanary(upgrade) {
  console.log(`Upgrading canary ${upgrade.Name} from ${upgrade.FromVersion} to
 ${upgrade.ToVersion}`);
  await synthetics.updateCanary({ Name: upgrade.Name, RuntimeVersion:
 upgrade.ToVersion }).promise();
  while (true) {
```

```
    await usleep(SLEEP_TIME);
    console.log(`Getting the state of canary ${upgrade.Name}`);
    const response = await synthetics.getCanary({ Name: upgrade.Name }).promise();
    const state = response.Canary.Status.State;
    console.log(`The state of canary ${upgrade.Name} is ${state}`);
    if (state === 'ERROR' || response.Canary.Status.StateReason) {
      throw response.Canary.Status.StateReason;
    }
    if (state !== 'UPDATING') {
      return;
    }
  }
}

function usleep(ms) {
  return new Promise(resolve => setTimeout(resolve, ms));
}
```

Versiones en tiempo de ejecución con Node.js y Puppeteer

La primera versión en tiempo de ejecución para Node.js y Puppeteer se ha denominado `syn-1.0`. Las versiones posteriores en tiempo de ejecución tienen la convención de nomenclatura `syn-`*`language`*`-`*`majorversion`*`.`*`minorversion`*. A partir de `syn-nodejs-puppeteer-3.0`, la convención de nomenclatura es `syn-`*`language`*`-`*`framework`*`-`*`majorversion`*`.`*`minorversion`*

Un sufijo adicional `-beta` muestra que la versión en tiempo de ejecución se encuentra actualmente en una versión preliminar beta.

Las versiones de tiempo de ejecución con el mismo número de versión principal son compatibles con las versiones anteriores.

> **Important**
>
> Está previsto que las siguientes versiones de tiempo de ejecución de CloudWatch Synthetics queden obsoletas el 8 de marzo de 2024.
>
> - `syn-nodejs-puppeteer-6.1`
> - `syn-nodejs-puppeteer-6.0`
> - `syn-nodejs-puppeteer-5.1`
> - `syn-nodejs-puppeteer-5.0`
> - `syn-nodejs-puppeteer-4.0`

Para obtener más información, consulte Política de soporte de tiempo de ejecución de CloudWatch Synthetics.

Important

IMPORTANTE: La dependencia incluida de AWS SDK para JavaScript v2 se eliminará y se actualizará para utilizar AWS SDK para JavaScript v3 en una futura versión del tiempo de ejecución. Cuando eso ocurra, podrá actualizar las referencias de su código de valor controlado. Como alternativa, puede seguir haciendo referencia a la dependencia incluida de AWS SDK para JavaScript v2 y utilizarla agregándola como una dependencia a su archivo zip de código fuente.

Notas para todas las versiones de tiempo de ejecución

Cuando se utiliza la versión de tiempo de ejecución `syn-nodejs-puppeteer-3.0`, asegúrese de que el script de valor controlado sea compatible con Node.js 12.x. Si se utiliza una versión anterior de una versión de tiempo de ejecución `syn-nodejs`, asegúrese de que el script sea compatible con Node.js 10.x.

El código de Lambda de un valor controlado está configurado para tener una memoria máxima de 1 GB. El tiempo de espera de cada ejecución de un valor controlado se agota transcurrido el valor correspondiente configurado. Si no se especifica ningún valor de tiempo de espera para un valor controlado, CloudWatch elige uno en función de la frecuencia de dicho valor controlado. Si configura un valor de tiempo de espera, haga que no sea inferior a 15 segundos para permitir arranques en frío de Lambda y el tiempo que tarda en arrancar la instrumentación de valor controlado.

Note

Las siguientes versiones de tiempo de ejecución de CloudWatch Synthetics quedaron obsoletas el 8 de enero de 2024. Esto se debe a que AWS Lambda hará que el tiempo de ejecución de Lambda Node.js 14 deje de funcionar el 4 de diciembre de 2023.

- `syn-nodejs-puppeteer-3.9`
- `syn-nodejs-puppeteer-3.8`
- `syn-nodejs-puppeteer-3.7`

- `syn-nodejs-puppeteer-3.6`
- `syn-nodejs-puppeteer-3.5`

Está previsto que las siguientes versiones de ejecución de CloudWatch Synthetics queden obsoletas el 13 de noviembre de 2022. Esto se debe a que AWS Lambda hará que el entorno de ejecución de Lambda Node.js 12 deje de funcionar el 14 de noviembre de 2022.

- `syn-nodejs-puppeteer-3.4`
- `syn-nodejs-puppeteer-3.3`
- `syn-nodejs-puppeteer-3.2`
- `syn-nodejs-puppeteer-3.1`
- `syn-nodejs-puppeteer-3.0`

Para obtener más información, consulte Política de soporte de tiempo de ejecución de CloudWatch Synthetics.

syn-nodejs-puppeteer-7.0

El tiempo de ejecución `syn-nodejs-puppeteer-7.0` es la versión de tiempo de ejecución más reciente para el tiempo de ejecución Node.js 18.x de Lambda. Utiliza Node.js y Puppeteer.

Dependencias principales:

- Tiempo de ejecución de Lambda Node.js 18.x
- Puppeteer-core de versión 21.9.0
- Chromium versión 121.0.6167.139

Tamaño del código:

El tamaño del código y las dependencias que puede empaquetar en este tiempo de ejecución es de 80 MB.

Nuevas características en syn-nodejs-puppeteer-7.0:

- Versiones actualizadas de las bibliotecas agrupadas incluidas en Chromium: las dependencias de Chromium y Puppeteer se actualizan a versiones nuevas.

> **Important**
>
> La migración de Puppeteer 19.7.0 a Puppeteer 21.9.0 introduce cambios importantes en relación con las pruebas y los filtros. Para obtener más información, consulte las secciones de CAMBIOS IMPORTANTES en puppeteer: v20.0.0 y puppeteer-core: v21.0.0.

Actualización recomendada al SDK v3 de AWS

El tiempo de ejecución de Lambda nodejs18.x no es compatible con el SDK v2 de AWS. Se recomienda que migre al SDK v3 de AWS.

syn-nodejs-puppeteer-6.2

Dependencias principales:

- Tiempo de ejecución de Lambda Node.js 18.x
- Puppeteer-core de versión 19.7.0
- Chromium versión 111.0.5563.146

Nuevas características en syn-nodejs-puppeteer-6.2:

- Versiones actualizadas de las bibliotecas agrupadas incluidas en Chromium
- Supervisión del almacenamiento efímero: este tiempo de ejecución agrega la supervisión del almacenamiento efímero en las cuentas de los clientes.
- Correcciones de errores

syn-nodejs-puppeteer-5.2

El tiempo de ejecución `syn-nodejs-puppeteer-5.2` es la versión de tiempo de ejecución más reciente para el tiempo de ejecución Node.js 16.x de Lambda. Utiliza Node.js y Puppeteer.

Dependencias principales:

- Tiempo de ejecución de Lambda Node.js 16.x
- Puppeteer-core de versión 19.7.0
- Chromium versión 111.0.5563.146

Nuevas características en syn-nodejs-puppeteer-5.2:

- Versiones actualizadas de las bibliotecas agrupadas incluidas en Chromium
- Correcciones de errores

syn-nodejs-puppeteer-6.1

> **Important**
>
> Esta versión de tiempo de ejecución está programada para quedar obsoleta el 8 de marzo de 2024. Para obtener más información, consulte Política de soporte de tiempo de ejecución de CloudWatch Synthetics.

Relaciones principales:

- Tiempo de ejecución de Lambda Node.js 18.x
- Puppeteer-core de versión 19.7.0
- Chromium versión 111.0.5563.146

Nuevas características en syn-nodejs-puppeteer-6.1:

- Mejoras de estabilidad: se añadió una lógica de reintento automático para administrar los errores de lanzamiento intermitentes de Puppeteer.
- Actualizaciones de dependencias: actualizaciones para algunos paquetes de dependencias de terceros.
- Valores controlados sin permisos de Amazon S3: se han corregido errores para que puedan seguir funcionando los valores controlados que no tengan ningún permiso de Amazon S3. Estos valores controlados que no tengan permisos de Amazon S3 no podrán subir capturas de pantalla u otros artefactos a Amazon S3. Para obtener más información sobre permisos de valores controlados, consulte Roles y permisos necesarios para los valores controlados.

> **Important**
>
> IMPORTANTE: La dependencia incluida de AWS SDK para JavaScript v2 se eliminará y se actualizará para utilizar AWS SDK para JavaScript v3 en una futura versión del tiempo

de ejecución. Cuando eso ocurra, podrá actualizar las referencias de su código de valor controlado. Como alternativa, puede seguir haciendo referencia a la dependencia incluida de AWS SDK para JavaScript v2 y utilizarla agregándola como una dependencia a su archivo zip de código fuente.

syn-nodejs-puppeteer-6.0

Important

Esta versión de tiempo de ejecución está programada para quedar obsoleta el 8 de marzo de 2024. Para obtener más información, consulte Política de soporte de tiempo de ejecución de CloudWatch Synthetics.

Relaciones principales:

- Tiempo de ejecución de Lambda Node.js 18.x
- Puppeteer-core de versión 19.7.0
- Chromium versión 111.0.5563.146

Nuevas características en syn-nodejs-puppeteer-6.0:

- Actualización de la dependencia: la dependencia de Node.js se ha actualizado a la versión 18.x.
- Compatibilidad con el modo de intercepción: se agregó la compatibilidad con el modo de intercepción cooperativa de Puppeteer a la biblioteca de tiempos de ejecución con valores controlados de Synthetics.
- Cambio en el comportamiento de rastreo: se modificó el comportamiento de rastreo predeterminado para rastrear solo las solicitudes de recuperación y xhr, y no las solicitudes de recursos. Puede habilitar el seguimiento de las solicitudes de recursos configurando la opción `traceResourceRequests`.
- Métrica de duración mejorada: la métrica `Duration` ahora excluye el tiempo de operación que utiliza el valor controlado para cargar artefactos, hacer capturas de pantalla y generar métricas de CloudWatch. Los valores de las métricas de `Duration` se notifican a CloudWatch y también se pueden ver en la consola Synthetics.

- Corrección de errores: limpia el núcleo volcado que se genera cuando Chromium se bloquea durante una ejecución de valor controlado.

> **Important**
>
> IMPORTANTE: La dependencia incluida de AWS SDK para JavaScript v2 se eliminará y se actualizará para utilizar AWS SDK para JavaScript v3 en una futura versión del tiempo de ejecución. Cuando eso ocurra, podrá actualizar las referencias de su código de valor controlado. Como alternativa, puede seguir haciendo referencia a la dependencia incluida de AWS SDK para JavaScript v2 y utilizarla agregándola como una dependencia a su archivo zip de código fuente.

syn-nodejs-puppeteer-5.1

> **Important**
>
> Esta versión de tiempo de ejecución está programada para quedar obsoleta el 8 de marzo de 2024. Para obtener más información, consulte Política de soporte de tiempo de ejecución de CloudWatch Synthetics.

Relaciones principales:

- Tiempo de ejecución de Lambda Node.js 16.x
- Puppeteer-core de versión 19.7.0
- Chromium versión 111.0.5563.146

Correcciones de errores en syn-nodejs-puppeteer-5.1:

- Corrección de errores: este tiempo de ejecución corrige un error en `syn-nodejs-puppeteer-5.0` por el que a los archivos HAR creados por los valores controlados les faltaban encabezados de solicitud.

syn-nodejs-puppeteer-5.0

> **Important**
>
> Esta versión de tiempo de ejecución está programada para quedar obsoleta el 8 de marzo de 2024. Para obtener más información, consulte Política de soporte de tiempo de ejecución de CloudWatch Synthetics.

Relaciones principales:

- Tiempo de ejecución de Lambda Node.js 16.x
- Puppeteer-core de versión 19.7.0
- Chromium versión 111.0.5563.146

Nuevas características en syn-nodejs-puppeteer-5.0:

- Actualización de dependencias: la versión básica de Puppeteer-core se ha actualizado a la 19.7.0. La versión de Chromium se ha actualizado a la 111.0.5563.146.

> **Important**
>
> La nueva versión básica de Puppeteer no es totalmente compatible con las versiones anteriores de Puppeteer. Algunos de los cambios de esta versión pueden provocar que los valores controlados existentes que utilizan funciones obsoletas de Puppeteer fallen. Para obtener más información, consulte los cambios importantes en los registros de cambios de las versiones 19.7.0 a 6.0 de Puppeteer-core, en los registros de cambios de Puppeteer.

syn-nodejs-puppeteer-4.0

> **Important**
>
> Esta versión de tiempo de ejecución está programada para quedar obsoleta el 8 de marzo de 2024. Para obtener más información, consulte Política de soporte de tiempo de ejecución de CloudWatch Synthetics.

Relaciones principales:

- Tiempo de ejecución de Lambda Node.js 16.x
- Puppeteer-core de versión 5.5.0
- Chromium versión 92.0.4512

Nuevas características en syn-nodejs-puppeteer-4.0:

- Actualización de la dependencia: la dependencia de Node.js se ha actualizado a la versión 16.x.

Tiempos de ejecución obsoletos para Node.js y Puppeteer

Los siguientes tiempos de ejecución para Node.js y Puppeteer han quedado obsoletos.

syn-nodejs-puppeteer-3.9

> **Important**
>
> Esta versión de tiempo de ejecución quedó obsoleta el 8 de enero de 2024. Para obtener más información, consulte Política de soporte de tiempo de ejecución de CloudWatch Synthetics.

Relaciones principales:

- Tiempo de ejecución Node.js 14.x de Lambda
- Puppeteer-core de versión 5.5.0
- Chromium versión 92.0.4512

Nuevas características de syn-nodejs-puppeteer-3.9:

- Actualizaciones de dependencias: actualiza algunos paquetes de dependencias de terceros.

syn-nodejs-puppeteer-3.8

> **Important**
>
> Esta versión de tiempo de ejecución quedó obsoleta el 8 de enero de 2024. Para obtener más información, consulte Política de soporte de tiempo de ejecución de CloudWatch Synthetics.

Relaciones principales:

- Tiempo de ejecución Node.js 14.x de Lambda
- Puppeteer-core de versión 5.5.0
- Chromium versión 92.0.4512

Nuevas características de syn-nodejs-puppeteer-3.8:

- Limpieza de perfiles: los perfiles de Chromium ahora se limpian después de cada ejecución de un valor controlado.

Correcciones de errores en syn-nodejs-puppeteer-3.8:

- Correcciones de errores: anteriormente, los valores controlados de supervisión visual a veces dejaban de funcionar correctamente después de una ejecución sin capturas de pantalla. Esto ya está resuelto.

syn-nodejs-puppeteer-3.7

> **Important**
>
> Esta versión de tiempo de ejecución quedó obsoleta el 8 de enero de 2024. Para obtener más información, consulte Política de soporte de tiempo de ejecución de CloudWatch Synthetics.

Relaciones principales:

- Tiempo de ejecución Node.js 14.x de Lambda

- Puppeteer-core de versión 5.5.0
- Chromium versión 92.0.4512

Nuevas características de syn-nodejs-puppeteer-3.7:

- Mejoras del registro: el valor controlado cargará los registros en Amazon S3 incluso si se agota el tiempo de espera o se bloquea.
- Reducción del tamaño de la capa de Lambda: el tamaño de la capa de Lambda utilizada para los valores controlados se reduce en un 34 %.

Correcciones de errores en syn-nodejs-puppeteer-3.7:

- Correcciones de errores: las fuentes en japonés, chino simplificado y chino tradicional se representarán correctamente.

syn-nodejs-puppeteer-3.6

> **Important**
>
> Esta versión de tiempo de ejecución quedó obsoleta el 8 de enero de 2024. Para obtener más información, consulte Política de soporte de tiempo de ejecución de CloudWatch Synthetics.

Relaciones principales:

- Tiempo de ejecución Node.js 14.x de Lambda
- Puppeteer-core de versión 5.5.0
- Chromium versión 92.0.4512

Nuevas características de syn-nodejs-puppeteer-3.6:

- Marcas de tiempo más precisas: la hora de inicio y la hora de parada de las ejecuciones de valores controlados ahora tienen una precisión de milisegundos.

syn-nodejs-puppeteer-3.5

> **Important**
>
> Esta versión de tiempo de ejecución quedó obsoleta el 8 de enero de 2024. Para obtener más información, consulte Política de soporte de tiempo de ejecución de CloudWatch Synthetics.

Relaciones principales:

- Tiempo de ejecución Node.js 14.x de Lambda
- Puppeteer-core de versión 5.5.0
- Chromium versión 92.0.4512

Nuevas características de syn-nodejs-puppeteer-3.5:

- Dependencias actualizadas: las únicas características nuevas de este tiempo de ejecución son las dependencias actualizadas.

syn-nodejs-puppeteer-3.4

> **Important**
>
> Esta versión de tiempo de ejecución quedó obsoleta el 13 de noviembre de 2022. Para obtener más información, consulte Política de soporte de tiempo de ejecución de CloudWatch Synthetics.

Dependencias principales:

- Tiempo de ejecución Node.js 12.x de Lambda
- Puppeteer-core de versión 5.5.0
- Chromium versión 88.0.4298.0

Nuevas características de syn-nodejs-puppeteer-3.4:

- Función de controlador personalizada: ahora puede utilizar una función de controlador personalizada para los scripts de los valores controlados. Los tiempos de ejecución anteriores requerían que el punto de entrada del script incluyera `.handler`.

  También puede colocar scripts de valores controlados en cualquier carpeta y pasar el nombre de la carpeta como parte del controlador. Por ejemplo, `MyFolder/MyScriptFile.functionname` se puede utilizar como punto de entrada.
- Información ampliada de archivos HAR: ahora puede ver las solicitudes con fallas, pendientes e incompletas en los archivos HAR producidos por los valores controlados.

syn-nodejs-puppeteer-3.3

> **Important**
>
> Esta versión de tiempo de ejecución quedó obsoleta el 13 de noviembre de 2022. Para obtener más información, consulte Política de soporte de tiempo de ejecución de CloudWatch Synthetics.

Dependencias principales:

- Tiempo de ejecución Node.js 12.x de Lambda
- Puppeteer-core de versión 5.5.0
- Chromium versión 88.0.4298.0

Nuevas características en syn-nodejs-puppeteer-3.3:

- Más opciones de cifrado de artefactos: para los valores controlados que utilicen este tiempo de ejecución o posterior, en lugar de utilizar una clave administrada de AWS para cifrar artefactos que el valor controlado almacena en Amazon S3, puede optar por utilizar una clave administrada por el cliente de AWS KMS o una clave administrada por Amazon S3. Para obtener más información, consulte Cifrado de artefactos de un valor controlado.

syn-nodejs-puppeteer-3.2

> **Important**
>
> Esta versión de tiempo de ejecución quedó obsoleta el 13 de noviembre de 2022. Para obtener más información, consulte Política de soporte de tiempo de ejecución de CloudWatch Synthetics.

Dependencias principales:

- Tiempo de ejecución Node.js 12.x de Lambda
- Puppeteer-core de versión 5.5.0
- Chromium versión 88.0.4298.0

Nuevas características en syn-nodejs-puppeteer-3.2:

- supervisión visual con capturas de pantalla: los valores controlados que utilizan este tiempo de ejecución o uno posterior pueden comparar una captura de pantalla que se ha tomado durante una ejecución con una versión de línea de base de la misma captura de pantalla. Si las capturas de pantalla son más diferentes que un umbral de porcentaje especificado, el valor controlado falla. Para obtener más información, consulte Supervisión visual o Proyecto de supervisión visual.
- Nuevas funciones relacionadas con información confidencial Se puede evitar que la información confidencial aparezca en los registros de valores controlados e informes. Para obtener más información, consulte Clase de SyntheticSloghelper.
- Función obsoleta La clase `RequestResponseLogHelper` ha quedado obsoleta en favor de otras opciones de configuración nuevas. Para obtener más información, consulte RequestResponseLogHelper class.

syn-nodejs-puppeteer-3.1

> **Important**
>
> Esta versión de tiempo de ejecución quedó obsoleta el 13 de noviembre de 2022. Para obtener más información, consulte Política de soporte de tiempo de ejecución de CloudWatch Synthetics.

Dependencias principales:

- Tiempo de ejecución Node.js 12.x de Lambda
- Puppeteer-core de versión 5.5.0
- Chromium versión 88.0.4298.0

Nuevas características en syn-nodejs-puppeteer-3.1:

- Capacidad para configurar métricas de CloudWatch: con este tiempo de ejecución, puede desactivar las métricas que no se necesitan. De lo contrario, los valores controlados publican varias métricas de CloudWatch para cada ejecución de valores controlados.
- Vinculación de captura de pantalla: se puede vincular una captura de pantalla a un paso de valor controlado una vez finalizado el paso. Para ello, tome la captura de pantalla mediante el comando takeScreenshot, con el nombre del paso al que desea asociar la captura de pantalla. Por ejemplo, puede que desee realizar un paso, agregar un tiempo de espera y, a continuación, tomar la captura de pantalla.
- El esquema de supervisión de latidos puede supervisar varias URL: se puede utilizar el esquema de supervisión de latidos en la consola de CloudWatch para supervisar varias URL y ver el estado, la duración, las capturas de pantalla asociadas y el motivo del error de cada URL en el resumen de pasos del informe de ejecución del canario.

syn-nodejs-puppeteer-3.0

> **Important**
>
> Esta versión de tiempo de ejecución quedó obsoleta el 13 de noviembre de 2022. Para obtener más información, consulte Política de soporte de tiempo de ejecución de CloudWatch Synthetics.

Dependencias principales:

- Tiempo de ejecución Node.js 12.x de Lambda
- Puppeteer-core de versión 5.5.0
- Chromium versión 88.0.4298.0

Nuevas características en syn-nodejs-puppeteer-3.0:

- Relaciones actualizadas: esta versión utiliza Puppeteer versión 5.5.0, Node.js 12.x y Chromium 88.0.4298.0.
- Acceso a buckets entre regiones: ahora puede especificar un bucket de S3 en otra región como el bucket donde el valor controlado almacena los archivos de registro, capturas de pantalla y archivos HAR.
- Nuevas funciones disponibles: esta versión añade funciones de biblioteca para recuperar el nombre del valor controlado y la versión de tiempo de ejecución de Synthetics.

  Para obtener más información, consulte Clase de Synthetics.

syn-nodejs-2.2

Esta sección contiene información sobre la versión de tiempo de ejecución `syn-nodejs-2.2`.

> **Important**
>
> Esta versión de tiempo de ejecución quedó obsoleta el 28 de mayo de 2021. Para obtener más información, consulte Política de soporte de tiempo de ejecución de CloudWatch Synthetics.

Relaciones principales:

- Tiempo de ejecución Node.JS 10.x de Lambda
- Puppeteer-core de versión 3.3.0
- Chromium versión 83.0.4103.0

Nuevas características en syn-nodejs-2.2:

- Supervise los valores controlados como pasos HTTP: ahora puede probar varias API en un único valor controlado. Cada API se prueba como un paso HTTP independiente, y CloudWatch Synthetics supervisa el estado de cada paso mediante métricas de pasos y el informe de pasos de CloudWatch Synthetics. CloudWatch Synthetics crea métricas de `SuccessPercent` y `Duration` para cada paso HTTP.

Esta funcionalidad la implementa la función executeHttpStep(stepName, requestOptions, callback, stepConfig). Para obtener más información, consulte executeHttpStep(stepName, requestOptions, [callback], [stepConfig]).

El esquema del valor controlado de API se actualiza para utilizar esta nueva característica.

- Informes de solicitudes HTTP: ahora se pueden ver informes detallados de solicitudes HTTP que capturan detalles como las cabeceras de solicitud y de respuesta, cuerpos de respuesta, códigos de estado, tiempos de error y rendimiento, tiempos de conexión TCP, tiempos de enlace TLS, la hora del primer byte y el tiempo de transferencia de contenido. Todas las solicitudes HTTP que utilizan el módulo HTTP o HTTPS que no se ven a simple vista se capturan aquí. Las cabeceras y el cuerpo de respuesta no se capturan de forma predeterminada, pero se pueden habilitar si se establecen opciones de configuración.
- Configuración global y a nivel de paso: se pueden establecer configuraciones de CloudWatch Synthetics a nivel global, que se aplican a todos los pasos de los canaries. También se pueden anular estas configuraciones en el nivel de paso al pasar los pares clave-valor de configuración para habilitar o desactivar determinadas opciones.

  Para obtener más información, consulte Clase de SyntheticsConfiguration.
- Continúe con la configuración del error del paso: puede optar por continuar la ejecución del valor controlado cuando un paso falla . Para la función `executeHttpStep`, esta opción está activada de forma predeterminada. Puede establecer esta opción una vez a nivel global o definirla de manera diferente por paso.

syn-nodejs-2.1

> **Important**
>
> Esta versión de tiempo de ejecución quedó obsoleta el 28 de mayo de 2021. Para obtener más información, consulte Política de soporte de tiempo de ejecución de CloudWatch Synthetics.

Relaciones principales:

- Tiempo de ejecución Node.JS 10.x de Lambda
- Puppeteer-core de versión 3.3.0

- Chromium versión 83.0.4103.0

Nuevas características en syn-nodejs-2.1:

- Capturas de pantalla de comportamiento configurable: proporciona la opción de desactivar las capturas de pantalla a través de canaries de UI. En canaries que utilizan versiones anteriores de los tiempos de ejecución, los canaries de UI siempre toman capturas de pantalla antes y después de cada paso. Con `syn-nodejs-2.1`, esto es configurable. La desactivación de las capturas de pantalla puede reducir los costos de almacenamiento de Amazon S3 y puede ayudarle a cumplir con las normas HIPAA. Para obtener más información, consulte Clase de SyntheticsConfiguration.
- Personalice los parámetros de inicio de Google Chrome Ahora puede configurar los argumentos utilizados cuando un valor controlado inicia una ventana del navegador Google Chrome. Para obtener más información, consulte Lanzamiento (opciones).

Puede haber un pequeño aumento en la duración del valor controlado al usar syn-nodejs-2.0 o uno posterior, en comparación con versiones anteriores de los tiempos de ejecución de valores controlados.

syn-nodejs-2.0

> **Important**
>
> Esta versión de tiempo de ejecución quedó obsoleta el 28 de mayo de 2021. Para obtener más información, consulte Política de soporte de tiempo de ejecución de CloudWatch Synthetics.

Relaciones principales:

- Tiempo de ejecución Node.JS 10.x de Lambda
- Puppeteer-core de versión 3.3.0
- Chromium versión 83.0.4103.0

Nuevas características en syn-nodejs-2.0:

- Relaciones actualizadas: esta versión de tiempo de ejecución utiliza Puppeteer-core versión 3.3.0 y Chromium versión 83.0.4103.0

- Soporte para el rastreo activo de X-Ray. Cuando un valor controlado tiene habilitado el rastreo, se envían los seguimientos de X-Ray para todas las llamadas que el valor controlado ha realizado que utilizan el navegador, el SDK de AWS, o módulos HTTP o HTTPS. Los valores controlados con seguimiento activado aparecen en el mapa de seguimiento de X-Ray, incluso cuando no envían solicitudes a otros servicios o aplicaciones que tienen habilitado el rastreo. Para obtener más información, consulte Canaries y rastreo X-Ray.
- Informes de Synthetics: para cada ejecución del valor controlado, CloudWatch Synthetics crea un informe llamado `SyntheticsReport-PASSED.json` o `SyntheticsReport-FAILED.json` que registra datos, como la hora de inicio, la hora de finalización, el estado y los errores. También registra los estados SUPERADO o NO SUPERADO de cada paso del script de valor controlado, así como los fallos y las capturas de pantalla tomadas para cada paso.
- Informe del verificador de enlaces que no funcionan: la nueva versión del verificador de enlaces que no funcionan que está incluido en este tiempo de ejecución crea un informe que incluye los enlaces que se han verificado, el código de estado, el motivo del error (si existe) y las capturas de pantalla de la página fuente y la de destino.
- Nuevas métricas de CloudWatch: Synthetics publica métricas denominadas `2xx`, `4xx`, `5xx`, y `RequestFailed` en el espacio de nombres de `CloudWatchSynthetics`. Estas métricas muestran el número de 200, 400, 500 y los errores de solicitud en las ejecuciones de valores controlados. Con esta versión de tiempo de ejecución, estas métricas se notifican solo para canaries de la UI y no para canaries de la API. También se reportan para los canaries de la API que comienzan con la versión de tiempo de ejecución `syn-nodejs-puppeteeer-2.2`.
- Archivos HAR ordenables: ahora puede ordenar los archivos HAR por código de estado, tamaño de solicitud y duración.
- Marca de tiempo de las métricas: las métricas de CloudWatch ahora se informan según el tiempo de invocación de Lambda en lugar de la hora de finalización de ejecución del valor controlado.

Corrección de errores en syn-nodejs-2.0:

- Se ha corregido el problema de errores de carga de artefactos de valores controlados que no se notificaban. Ahora aparecen como errores de ejecución.
- Se ha corregido el problema con respecto a las solicitudes redirigidas (3xx) que se registraban incorrectamente como errores.
- Se ha corregido el problema con respecto a las capturas de pantalla que se enumeraban a partir de 0. Ahora deben comenzar con 1.

- Se ha corregido el problema con respecto a las capturas de pantalla que eran ilegibles para caracteres chinos y japoneses.

Puede haber un pequeño aumento en la duración del valor controlado al usar syn-nodejs-2.0 o uno posterior, en comparación con versiones anteriores de los tiempos de ejecución de valores controlados.

syn-nodejs-2.0-beta

> **Important**
>
> Esta versión de tiempo de ejecución quedó obsoleta el 8 de febrero de 2021. Para obtener más información, consulte Política de soporte de tiempo de ejecución de CloudWatch Synthetics.

Relaciones principales:

- Tiempo de ejecución Node.JS 10.x de Lambda
- Puppeteer-core de versión 3.3.0
- Chromium versión 83.0.4103.0

Nuevas características en syn-nodejs-2.0-beta:

- Relaciones actualizadas: esta versión de tiempo de ejecución utiliza Puppeteer-core versión 3.3.0 y Chromium versión 83.0.4103.0
- Informe de Synthetics: para cada ejecución del valor controlado, CloudWatch Synthetics crea un informe llamado `SyntheticsReport-PASSED.json` o `SyntheticsReport-FAILED.json` que registra datos, como la hora de inicio, la hora de finalización, el estado y los fallos. También registra los estados SUPERADO o NO SUPERADO de cada paso del script de valor controlado, así como los fallos y las capturas de pantalla tomadas para cada paso.
- Informe del verificador de enlaces que no funcionan: la nueva versión del verificador de enlaces que no funcionan que está incluido en este tiempo de ejecución crea un informe que incluye los enlaces que se han verificado, el código de estado, el motivo del error (si existe) y las capturas de pantalla de la página fuente y la de destino.
- Nuevas métricas de CloudWatch: Synthetics publica métricas denominadas `2xx`, `4xx`, `5xx`, y `RequestFailed` en el espacio de nombres de `CloudWatchSynthetics`. Estas métricas

muestran el número de 200, 400, 500 y errores de solicitud en las ejecuciones de los valores controlados. Estas métricas se notifican solo para los canaries de la UI y no para los canarios de la API.

- Archivos HAR ordenables: ahora puede ordenar los archivos HAR por código de estado, tamaño de solicitud y duración.
- Marca de tiempo de las métricas: las métricas de CloudWatch ahora se informan según el tiempo de invocación de Lambda en lugar de la hora de finalización de la ejecución del valor controlado.

Corrección de errores en syn-nodejs-2.0-beta:

- Se ha corregido el problema de errores de carga de artefactos de valores controlados que no se notificaban. Ahora aparecen como errores de ejecución.
- Se ha corregido el problema con respecto a las solicitudes redirigidas (3xx) que se registraban incorrectamente como errores.
- Se ha corregido el problema con respecto a las capturas de pantalla que se enumeraban a partir de 0. Ahora deben comenzar con 1.
- Se ha corregido el problema con respecto a las capturas de pantalla que eran ilegibles para caracteres chinos y japoneses.

syn-1.0

> **Important**
>
> Esta versión de tiempo de ejecución está programada para quedar obsoleta el 28 de mayo de 2021. Para obtener más información, consulte Política de soporte de tiempo de ejecución de CloudWatch Synthetics.

La primera versión de tiempo de ejecución de Synthetics es `syn-1.0`.

Relaciones principales:

- Tiempo de ejecución Node.JS 10.x de Lambda
- Puppeteer-core de versión 1.14.0
- La versión de Chromium que coincide con Puppeteer-core 1.14.0

Versiones en tiempo de ejecución con Python y Selenium Webdriver

Las siguientes secciones contienen información acerca de las versiones de tiempo de ejecución de CloudWatch Synthetics para Python y Selenium Webdriver. Selenium es una herramienta de automatización de navegadores de código abierto. Para obtener más información acerca de Selenium, consulte www.selenium.dev/

La convención de nomenclatura de estas versiones de tiempo de ejecución es `syn-`*`language`*`-`*`framework`*`-`*`majorversion`*`.`*`minorversion`*.

> **Important**
>
> Está previsto que las siguientes versiones de tiempo de ejecución de CloudWatch Synthetics queden obsoletas el 8 de marzo de 2024.
>
> - `syn-python-selenium-2.0`
> - `syn-python-selenium-1.3`
> - `syn-python-selenium-1.2`
> - `syn-python-selenium-1.1`
> - `syn-python-selenium-1.0`
>
> Para obtener más información, consulte Política de soporte de tiempo de ejecución de CloudWatch Synthetics.

syn-python-selenium-3.0

La versión 3.0 es el tiempo de ejecución más reciente de CloudWatch Synthetics para Python y Selenium.

Relaciones principales:

- Python 3.8
- Selenium 4.15.1
- Chromium versión 121.0.6167.139

Nuevas características en syn-python-selenium-3.0:

- Versiones actualizadas de las bibliotecas agrupadas incluidas en Chromium: la dependencia de Chromium se actualiza a una versión nueva.

syn-python-selenium-2.1

Relaciones principales:

- Python 3.8
- Selenium 4.15.1
- Chromium versión 111.0.5563.146

Nuevas características en syn-python-selenium-2.1:

- Versiones actualizadas de las bibliotecas agrupadas incluidas en Chromium: las dependencias de Chromium y Selenium se actualizan a versiones nuevas.

syn-python-selenium-2.0

> **Important**
>
> Esta versión de tiempo de ejecución está programada para quedar obsoleta el 8 de marzo de 2024. Para obtener más información, consulte Política de soporte de tiempo de ejecución de CloudWatch Synthetics.

Relaciones principales:

- Python 3.8
- Selenium 4.10.0
- Chromium versión 111.0.5563.146

Nuevas características en syn-python-selenium-2.0:

- Dependencias actualizadas: las dependencias de Chromium y Selenium se actualizan a las nuevas versiones.

Correcciones de errores en syn-python-selenium-2.0:

- Marca de tiempo agregada: se ha agregado una marca de tiempo los registros de valores controlados.
- Reutilización de la sesión: se ha corregido un error que impedía a los valores controlados reutilizar la sesión de su anterior ejecución.

syn-python-selenium-1.3

> **Important**
>
> Esta versión de tiempo de ejecución está programada para quedar obsoleta el 8 de marzo de 2024. Para obtener más información, consulte Política de soporte de tiempo de ejecución de CloudWatch Synthetics.

Relaciones principales:

- Python 3.8
- Selenium 3.141.0
- Chromium versión 92.0.4512.0

Nuevas características en syn-python-selenium-1.3:

- Marcas de tiempo más precisas: la hora de inicio y la hora de parada de las ejecuciones del valores controlados ahora tienen una precisión de milisegundos.

syn-python-selenium-1.2

> **Important**
>
> Esta versión de tiempo de ejecución está programada para quedar obsoleta el 8 de marzo de 2024. Para obtener más información, consulte Política de soporte de tiempo de ejecución de CloudWatch Synthetics.

Relaciones principales:

- Python 3.8
- Selenium 3.141.0
- Chromium versión 92.0.4512.0

- Dependencias actualizadas: las únicas características nuevas de este tiempo de ejecución son las dependencias actualizadas.

syn-python-selenium-1.1

> **Important**
>
> Esta versión de tiempo de ejecución está programada para quedar obsoleta el 8 de marzo de 2024. Para obtener más información, consulte Política de soporte de tiempo de ejecución de CloudWatch Synthetics.

Relaciones principales:

- Python 3.8
- Selenium 3.141.0
- Chromium versión 83.0.4103.0

Características:

- Función de controlador personalizada: ahora puede utilizar una función de controlador personalizada para los scripts de los valores controlados. Los tiempos de ejecución anteriores requerían que el punto de entrada del script incluyera `.handler`.

  También puede colocar scripts de valores controlados en cualquier carpeta y pasar el nombre de la carpeta como parte del controlador. Por ejemplo, `MyFolder/MyScriptFile.functionname` se puede utilizar como punto de entrada.
- Opciones de configuración para agregar métricas y configuraciones de error de pasos: estas opciones ya estaban disponibles en los tiempos de ejecución para los valores controlados de Node.js. Para obtener más información, consulte Clase SyntheticsConfiguration.
- Argumentos personalizados en Chrome: ahora puede abrir un navegador en el modo de incógnito o pasar con la configuración del servidor proxy. Para obtener más información, consulte Chrome().

- Buckets de artefactos de varias regiones: un valor controlado puede almacenar sus artefactos en un bucket de Amazon S3 de una región diferente.
- Correcciones de errores, incluida una corrección para el problema **`index.py`**: con los tiempos de ejecución anteriores, un archivo de valor controlado denominado `index.py` causaba excepciones porque entraba en conflicto con el nombre del archivo de la biblioteca. Este problema ya está resuelto.

Syn-python-selenium-1.0

> **Important**
>
> Esta versión de tiempo de ejecución está programada para quedar obsoleta el 8 de marzo de 2024. Para obtener más información, consulte Política de soporte de tiempo de ejecución de CloudWatch Synthetics.

Relaciones principales:

- Python 3.8
- Selenium 3.141.0
- Chromium versión 83.0.4103.0

Características:

- Compatibilidad con Selenium: puede escribir scripts de valores controlados mediante el marco de prueba de Selenium. Puede llevar los scripts de Selenium desde otro lugar a CloudWatch Synthetics con cambios mínimos, y funcionarán con servicios de AWS.

Escritura de un script de valor controlado

En las siguientes secciones, se explica cómo escribir un script de canario y cómo integrar un canario con otros servicios de AWS y con bibliotecas y dependencias externas.

Temas

- Escritura de un script de valor controlado Node.js
- Escritura de un script de valor controlado Python

- Cambio de un script de Puppeteer existente para usarlo como un valor controlado de Synthetics
- Cambio de un script existente de Puppeteer Synthetics para autenticar certificados no estándar

Escritura de un script de valor controlado Node.js

Temas

- Creación de un valor controlado de CloudWatch Synthetics desde cero
- Empaquetado de los archivos de valores controlados de Node.js
- Cambio de un script de Puppeteer existente para usarlo como valor controlado de Synthetics
- Variables de entorno
- Integración del valor controlado con otros servicios de AWS
- Forzar al valor controlado para que utilice una dirección IP estática

Creación de un valor controlado de CloudWatch Synthetics desde cero

Aquí hay un ejemplo de script mínimo de valor controlado de Synthetics. Este script pasa como una ejecución correcta y devuelve una cadena. Para ver el aspecto de un valor controlado erróneo, cambie `let fail = false;` a `let fail = true;`.

Debe definir una función de punto de entrada para el script de valor controlado. Para ver cómo se cargan los archivos en la ubicación especificada de Amazon S3 como la `ArtifactS3Location` del valor controlado, cree estos archivos en la carpeta /tmp. Después de que el script se ejecuta, el estado superado o no superado y las métricas de duración se ejecutan en CloudWatch y los archivos bajo /tmp se cargan en S3.

```
const basicCustomEntryPoint = async function () {

    // Insert your code here

    // Perform multi-step pass/fail check

    // Log decisions made and results to /tmp

    // Be sure to wait for all your code paths to complete
    // before returning control back to Synthetics.
    // In that way, your canary will not finish and report success
    // before your code has finished executing
```

```
    // Throw to fail, return to succeed
    let fail = false;
    if (fail) {
        throw "Failed basicCanary check.";
    }

    return "Successfully completed basicCanary checks.";
};

exports.handler = async () => {
    return await basicCustomEntryPoint();
};
```

A continuación, expandiremos el script para usar el registro de Synthetics y realizar una llamada usando el SDK de AWS. A modo de demostración, este script creará un cliente de Amazon DynamoDB y realizará una llamada a las listTables DynamoDB de la API. Registra la respuesta a la solicitud y los registros se superan o no en función de si la solicitud se realizó correctamente.

```
const log = require('SyntheticsLogger');
const AWS = require('aws-sdk');
// Require any dependencies that your script needs
// Bundle additional files and dependencies into a .zip file with folder structure
// nodejs/node_modules/additional files and folders

const basicCustomEntryPoint = async function () {

    log.info("Starting DynamoDB:listTables canary.");

    let dynamodb = new AWS.DynamoDB();
    var params = {};
    let request = await dynamodb.listTables(params);
    try {
        let response = await request.promise();
        log.info("listTables response: " + JSON.stringify(response));
    } catch (err) {
        log.error("listTables error: " + JSON.stringify(err), err.stack);
        throw err;
    }

    return "Successfully completed DynamoDB:listTables canary.";
};

exports.handler = async () => {
```

```
    return await basicCustomEntryPoint();
};
```

Empaquetado de los archivos de valores controlados de Node.js

Si carga los scripts de valores controlados mediante una ubicación de Amazon S3, el archivo .zip debe incluir el script en esta estructura de carpetas: `nodejs/node_modules/`*`myCanaryFilename.js file`*.

Si tiene más de un solo archivo `.js` o tiene una dependencia de la que depende su script, puede agruparlos todos en un único archivo ZIP que contenga la estructura de carpetas `nodejs/node_modules/`*`myCanaryFilename.js file and other folders and files`*. Si utiliza `syn-nodejs-puppeteer-3.4` o uno posterior, puede optar por colocar los archivos de valores controlados en otra carpeta y crear su estructura de carpetas de la siguiente manera: `nodejs/node_modules/`*`myFolder`*`/`*`myCanaryFilename.js file and other folders and files`*.

Nombre del controlador

Asegúrese de establecer el punto de entrada del script (controlador) del valor controlado como `myCanaryFilename.functionName` para que coincida con el nombre de archivo del punto de entrada del script. Si utiliza un tiempo de ejecución anterior a `syn-nodejs-puppeteer-3.4`, el `functionName` debe ser `handler`. Si utiliza `syn-nodejs-puppeteer-3.4` o uno posterior, puede elegir cualquier nombre de función como el controlador. Si utiliza `syn-nodejs-puppeteer-3.4` o uno posterior, también puede almacenar el valor controlado en una carpeta independiente, como `nodejs/node_modules/myFolder/my_canary_filename`. Si lo almacena en una carpeta independiente, especifique esa ruta en el punto de entrada del script, como `myFolder/my_canary_filename.functionName`.

Cambio de un script de Puppeteer existente para usarlo como valor controlado de Synthetics

En esta sección se explica cómo tomar scripts de Puppeteer y modificarlos para que se ejecuten como scripts de valor controlado de Synthetics. Para obtener más información acerca de Puppeteer, consulte Puppeteer API v1.14.0.

Comenzaremos con este ejemplo de script de Puppeteer:

```
const puppeteer = require('puppeteer');

(async () => {
  const browser = await puppeteer.launch();
  const page = await browser.newPage();
```

```
  await page.goto('https://example.com');
  await page.screenshot({path: 'example.png'});

  await browser.close();
})();
```

Los pasos de conversión son los siguientes:

- Crear y exportar una función de `handler`. El controlador es la función de punto de entrada para el script. Si utiliza un tiempo de ejecución anterior a `syn-nodejs-puppeteer-3.4`, la función del controlador debe denominarse `handler`. Si utiliza `syn-nodejs-puppeteer-3.4` o uno posterior, la función puede tener cualquier nombre, pero debe ser el mismo nombre que se usa en el script. Además, si utiliza `syn-nodejs-puppeteer-3.4` o uno posterior, puede almacenar los scripts en cualquier carpeta y especificar dicha carpeta como parte del nombre del controlador.

  ```
  const basicPuppeteerExample = async function () {};

  exports.handler = async () => {
      return await basicPuppeteerExample();
  };
  ```

- Use la dependencia de `Synthetics`.

  ```
  var synthetics = require('Synthetics');
  ```

- Utilice la función de `Synthetics.getPage` para obtener un objeto `Page` de Puppeteer.

  ```
  const page = await synthetics.getPage();
  ```

  El objeto page devuelto por la función Synthetics.getPage tiene los eventos page.on `request`, `response` y `requestfailed` instrumentados para el registro. Synthetics también configura la generación de archivos HAR para las solicitudes y respuestas en la página y agrega el ARN del valor controlado a los encabezados del agente de usuario de las solicitudes salientes en la página.

El script ya está listo para ser ejecutado como un valor controlado de Synthetics. Aquí está el script actualizado:

```
var synthetics = require('Synthetics');  // Synthetics dependency

const basicPuppeteerExample = async function () {
```

```
    const page = await synthetics.getPage(); // Get instrumented page from Synthetics
    await page.goto('https://example.com');
    await page.screenshot({path: '/tmp/example.png'}); // Write screenshot to /tmp
 folder
};

exports.handler = async () => {  // Exported handler function
    return await basicPuppeteerExample();
};
```

Variables de entorno

Puede utilizar variables de entorno al crear canaries. Esto le permite escribir un único script de valor controlado y luego usar ese script con diferentes valores para crear rápidamente varios valores controlados que tengan una tarea similar.

Suponga, por ejemplo, que su organización tiene puntos de enlaces como `prod`, `dev`, y `pre-release` para las diferentes etapas del desarrollo del software, y que necesita crear canaries para probar cada uno de estos puntos de enlace. Puede escribir un único script de valor controlado que pruebe el software y, a continuación, especificar los valores diferentes para la variable de entorno de punto de conexión cuando cree cada uno de los tres valores controlados. A continuación, cuando se crea un valor controlado, se especifica el script y los valores que se van a utilizar para las variables de entorno.

Los nombres de las variables de entorno pueden contener letras, números y guiones bajos. Deben comenzar con una letra y tener al menos dos caracteres. El tamaño total de las variables de entorno no puede superar los 4 KB. No es posible especificar variables de entorno reservadas de Lambda como claves para sus variables de entorno. Para obtener más información acerca de las variables de entorno reservadas, consulte Runtime environment variables (Variables de entorno en tiempo de ejecución).

> **Important**
>
> Las claves y los valores de las variables de entorno no están cifrados. No guarde información confidencial en ellos.

En el siguiente ejemplo el script utiliza dos variables de entorno. Este script es para un valor controlado que verifica si hay una página web disponible. Utiliza variables de entorno para parametrizar tanto la URL que verifica como el nivel de registro de CloudWatch Synthetics que utiliza.

La siguiente función establece `LogLevel` al valor de la variable de entorno `LOG_LEVEL`.

```
 synthetics.setLogLevel(process.env.LOG_LEVEL);
```

La función establece `URL` al valor de la variable de entorno `URL`.

```
const URL = process.env.URL;
```

Este es el script completo. Cuando se crea un valor controlado con este script, se especifican los valores para las variables de entorno `LOG_LEVEL` y `URL`.

```
var synthetics = require('Synthetics');
const log = require('SyntheticsLogger');

const pageLoadEnvironmentVariable = async function () {

    // Setting the log level (0-3)
    synthetics.setLogLevel(process.env.LOG_LEVEL);
    // INSERT URL here
    const URL = process.env.URL;

    let page = await synthetics.getPage();
    //You can customize the wait condition here. For instance,
    //using 'networkidle2' may be less restrictive.
    const response = await page.goto(URL, {waitUntil: 'domcontentloaded', timeout:
 30000});
    if (!response) {
        throw "Failed to load page!";
    }
    //Wait for page to render.
    //Increase or decrease wait time based on endpoint being monitored.
    await page.waitFor(15000);
    await synthetics.takeScreenshot('loaded', 'loaded');
    let pageTitle = await page.title();
    log.info('Page title: ' + pageTitle);
    log.debug('Environment variable:' + process.env.URL);

    //If the response status code is not a 2xx success code
    if (response.status() < 200 || response.status() > 299) {
        throw "Failed to load page!";
    }
};
```

```
exports.handler = async () => {
    return await pageLoadEnvironmentVariable();
};
```

Traspaso de las variables de entorno al script

Para pasar variables de entorno al script cuando cree un valor controlado en la consola, especifique las claves y los valores de las variables de entorno en la sección Variables de entorno en la consola. Para obtener más información, consulte Creación de un valor controlado.

Para pasar variables de entorno a través de la API o AWS CLI, utilice el parámetro `EnvironmentVariables` en la sección `RunConfig`. A continuación, se observa un ejemplo del comando de AWS CLI que crea un valor controlado que utiliza dos variables de entorno con claves de `Environment` y `Region`.

```
aws synthetics create-canary --cli-input-json '{
   "Name":"nameofCanary",
   "ExecutionRoleArn":"roleArn",
   "ArtifactS3Location":"s3://cw-syn-results-123456789012-us-west-2",
   "Schedule":{
      "Expression":"rate(0 minute)",
      "DurationInSeconds":604800
   },
   "Code":{
      "S3Bucket": "canarycreation",
      "S3Key": "cwsyn-mycanaryheartbeat-12345678-d1bd-1234-
abcd-123456789012-12345678-6a1f-47c3-b291-123456789012.zip",
      "Handler":"pageLoadBlueprint.handler"
   },
   "RunConfig": {
      "TimeoutInSeconds":60,
      "EnvironmentVariables": {
         "Environment":"Production",
         "Region": "us-west-1"
      }
   },
   "SuccessRetentionPeriodInDays":13,
   "FailureRetentionPeriodInDays":13,
   "RuntimeVersion":"syn-nodejs-2.0"
}'
```

Integración del valor controlado con otros servicios de AWS

Todos los canaries pueden utilizar la biblioteca de AWS SDK. Puede utilizar esta biblioteca cuando escriba su valor controlado para integrarlo con otros servicios de AWS.

Para ello, debe agregar el siguiente código al valor controlado. Para estos ejemplos, AWS Secrets Manager se utiliza como servicio para la integración del valor controlado.

- Importar el SDK de AWS.

```
const AWS = require('aws-sdk');
```

- Cree un cliente para el servicio de AWS con el que se está integrando.

```
const secretsManager = new AWS.SecretsManager();
```

- Use el cliente para realizar llamadas a la API a ese servicio.

```
var params = {
  SecretId: secretName
};
return await secretsManager.getSecretValue(params).promise();
```

El siguiente fragmento de código de script de valor controlado muestra un ejemplo de integración con Secrets Manager con más detalle.

```
var synthetics = require('Synthetics');
const log = require('SyntheticsLogger');

const AWS = require('aws-sdk');
const secretsManager = new AWS.SecretsManager();

const getSecrets = async (secretName) => {
    var params = {
        SecretId: secretName
    };
    return await secretsManager.getSecretValue(params).promise();
}

const secretsExample = async function () {
    let URL = "<URL>";
```

```
    let page = await synthetics.getPage();

    log.info(`Navigating to URL: ${URL}`);
    const response = await page.goto(URL, {waitUntil: 'domcontentloaded', timeout:
 30000});

    // Fetch secrets
    let secrets = await getSecrets("secretname")

    /**
    * Use secrets to login.
    *
    * Assuming secrets are stored in a JSON format like:
    * {
    *   "username": "<USERNAME>",
    *   "password": "<PASSWORD>"
    * }
    **/
    let secretsObj = JSON.parse(secrets.SecretString);
    await synthetics.executeStep('login', async function () {
        await page.type(">USERNAME-INPUT-SELECTOR<", secretsObj.username);
        await page.type(">PASSWORD-INPUT-SELECTOR<", secretsObj.password);

        await Promise.all([
          page.waitForNavigation({ timeout: 30000 }),
          await page.click(">SUBMIT-BUTTON-SELECTOR<")
        ]);
    });

    // Verify login was successful
    await synthetics.executeStep('verify', async function () {
        await page.waitForXPath(">SELECTOR<", { timeout: 30000 });
    });
};

exports.handler = async () => {
    return await secretsExample();
};
```

Forzar al valor controlado para que utilice una dirección IP estática

Se puede configurar un valor controlado para que utilice una dirección IP estática.

Para forzar a un valor controlado a utilizar una dirección IP estática

1. Cree una nueva VPC Para obtener más información, consulte Utilización de DNS con su VPC.
2. Cree una gateway de Internet. Para obtener más información, consulte Adding an internet gateway to your VPC (Cómo añadir una gateway de Internet a la VPC).
3. Cree una subred pública en la nueva VPC.
4. Agregue una nueva tabla de enrutamiento a la VPC.
5. Agregue una ruta en la nueva tabla de enrutamiento, que va desde `0.0.0.0/0` a la gateway de Internet.
6. Asocie la nueva tabla de enrutamiento con la subred pública.
7. Cree una dirección IP elástica Para obtener más información, consulte Elastic IP addresses (Direcciones IP elásticas).
8. Cree una nueva gateway NAT y asígnela a la subred pública y a la dirección IP elástica.
9. Cree las subredes privadas en la VPC
10. Agregue una ruta a la tabla de enrutamiento predeterminada de la VPC, que va desde `0.0.0.0/0` a la gateway NAT
11. Cree el valor controlado.

Escritura de un script de valor controlado Python

Este script pasa como una ejecución correcta y devuelve una cadena. Cómo ver el aspecto de un valor controlado erróneo, cambie error = Falso a error = verdadero

```
def basic_custom_script():
    # Insert your code here
    # Perform multi-step pass/fail check
    # Log decisions made and results to /tmp
    # Be sure to wait for all your code paths to complete
    # before returning control back to Synthetics.
    # In that way, your canary will not finish and report success
    # before your code has finished executing
    fail = False
    if fail:
        raise Exception("Failed basicCanary check.")
    return "Successfully completed basicCanary checks."
def handler(event, context):
    return basic_custom_script()
```

Empaquetado de los archivos de valores controlados de Python

Si tiene más de un archivo .py o el script tiene una dependencia, puede agruparlos todos en un único archivo ZIP. Si utiliza el tiempo de ejecución `syn-python-selenium-1.1`, el archivo ZIP debe contener el archivo .py principal del valor controlado dentro de una carpeta `python`, como `python/my_canary_filename.py`. Si utiliza `syn-python-selenium-1.1` o uno posterior, puede utilizar una carpeta diferente, como `python/myFolder/my_canary_filename.py`.

Este archivo ZIP debe contener todas las carpetas y archivos necesarios, pero los demás archivos no necesitan estar en la carpeta `python`.

Asegúrese de establecer el punto de entrada del script del valor controlado como `my_canary_filename.functionName` para que coincida con el nombre del archivo y el nombre de la función del punto de entrada de su script. Si utiliza el tiempo de ejecución `syn-python-selenium-1.0`, el `functionName` debe ser `handler`. Si utiliza `syn-python-selenium-1.1` o uno posterior, no se aplica esta restricción para el nombre del controlador y, además, usted puede optar por almacenar el valor controlado en una carpeta independiente, como `python/myFolder/my_canary_filename.py`. Si lo almacena en una carpeta independiente, especifique esa ruta en el punto de entrada del script, como `myFolder/my_canary_filename.functionName`.

Cambio de un script de Puppeteer existente para usarlo como un valor controlado de Synthetics

Se puede modificar rápidamente un script existente para Python y Selenium para ser utilizado como un valor controlado. Para obtener más información acerca de Selenium, consulte www.selenium.dev/.

Para este ejemplo comenzaremos con el siguiente script de Selenium:

```
from selenium import webdriver

def basic_selenium_script():
    browser = webdriver.Chrome()
    browser.get('https://example.com')
    browser.save_screenshot('loaded.png')

basic_selenium_script()
```

Los pasos de conversión son los siguientes:

Para convertir un script de Selenio para ser utilizado como un valor controlado

1. Cambie la instrucción `import` para usar Selenium del módulo `aws_synthetics`:

```
from aws_synthetics.selenium import synthetics_webdriver as webdriver
```

El módulo Selenium de `aws_synthetics` garantiza que el valor controlado pueda emitir métricas y registros, generar un archivo HAR y trabajar con otras características de CloudWatch Synthetics.

2. Cree una función de controlador y llame al método de Selenium. El controlador es la función de punto de entrada para el script.

   Si utiliza `syn-python-selenium-1.0`, la función del controlador debe denominarse `handler`. Si utiliza `syn-python-selenium-1.1` o uno posterior, la función puede tener cualquier nombre, pero debe ser el mismo nombre que se usa en el script. Además, si utiliza `syn-python-selenium-1.1` o uno posterior, puede almacenar los scripts en cualquier carpeta y especificar dicha carpeta como parte del nombre del controlador.

```
def handler(event, context):
    basic_selenium_script()
```

El script ahora se ha actualizado para ser un valor controlado de CloudWatch Synthetics. Aquí está el script actualizado:

```
from aws_synthetics.selenium import synthetics_webdriver as webdriver

def basic_selenium_script():
    browser = webdriver.Chrome()
    browser.get('https://example.com')
    browser.save_screenshot('loaded.png')

def handler(event, context):
    basic_selenium_script()
```

Cambio de un script existente de Puppeteer Synthetics para autenticar certificados no estándar

Un caso de uso importante de canarios de Synthetics es poder supervisar sus propios puntos de conexión. Si quiere supervisar un punto de conexión que no está preparado para el tráfico externo, esta supervisión puede significar a veces que no dispone de un certificado adecuado firmado por una autoridad de certificación externa de confianza.

Dos posibles soluciones para este escenario son las siguientes:

- Para autenticar un certificado de cliente, consulte How to validate authentication using Amazon CloudWatch Synthetics – Part 2.
- Para autenticar un certificado autofirmado, consulte How to validate authentication with self-signed certificates in Amazon CloudWatch Synthetics.

No solo tiene estas dos opciones cuando utiliza canarios de CloudWatch Synthetics. Puede ampliar estas características y agregar su lógica empresarial mediante la ampliación del código del canario.

> **Note**
>
> Los canarios de Synthetics que se ejecutan en tiempos de ejecución de Python tienen la marca `--ignore-certificate-errors` habilitada de forma innata, por lo que esos canarios no deberían tener problemas para llegar a sitios con configuraciones de certificados no estándar.

Funciones de la biblioteca disponibles para los scripts de valor controlado

CloudWatch Synthetics incluye varias funciones y clases integradas a las que puede llamar cuando escriba scripts de Node.js para su uso como canaries.

Algunas aplican a los canaries de la UI y a la API. Otras se aplican únicamente a los canaries de la interfaz de usuario. Un valor controlado de interfaz de usuario es aquel que utiliza la función `getPage()` y que usa Puppeteer como controlador web para navegar e interactuar con páginas web.

> **Note**
>
> Siempre que actualice un valor controlado para utilizar una nueva versión del tiempo de ejecución de Synthetics, todas las funciones de la biblioteca Synthetics que utilice su valor controlado también se actualizarán automáticamente a la misma versión de NodeJS que admita el tiempo de ejecución de Synthetics.

Temas

- Funciones de la biblioteca disponibles para los scripts de valor controlado de Node.js

- Funciones de la biblioteca disponibles para los scripts de valores controlados de Python que usan Selenium

Funciones de la biblioteca disponibles para los scripts de valor controlado de Node.js

En esta sección se enumeran las funciones de biblioteca disponibles para los scripts de valor controlado de Node.js.

Temas

- Funciones y clases de biblioteca aplicables a todos los canaries
- Funciones y clases de biblioteca Node.js que solo se aplican a los canaries de la UI
- Clases y funciones de biblioteca Node.js que se aplican sólo a los canaries de la API

Funciones y clases de biblioteca aplicables a todos los canaries

Las siguientes funciones de biblioteca de CloudWatch Synthetics para Node.js son útiles para todos los canaries.

Temas

- Clase de Synthetics
- Clase de SyntheticsConfiguration
- Registrador de Synthetics
- Clase de SyntheticSloghelper

Clase de Synthetics

Las siguientes funciones para todos los canaries están en la clase de Synthetics.

addExecutionError(errorMessage, ex);

`errorMessage` describe el error y `ex` es la excepción que se ha encontrado

`addExecutionError` puede usarse para establecer errores de ejecución para el valor controlado. Se produce un error en el valor controlado sin interrumpir la ejecución del script. Tampoco afecta a las métricas de `successPercent`.

Debe realizar un seguimiento de los errores como errores de ejecución sólo si no son importantes para indicar el éxito o el error del script valor controlado.

A continuación, se muestra un ejemplo del uso de un `addExecutionError`. Está supervisando la disponibilidad de su punto de conexión y tomando capturas de pantalla después de que la página se haya cargado. Debido a que el hecho de no tomar una captura de pantalla no determina la disponibilidad del punto de enlace, puede detectar cualquier error que encuentre al tomar capturas de pantalla y agregarlos como errores de ejecución. Las métricas de disponibilidad seguirán indicando que el punto de conexión está activo y en ejecución, pero el estado del valor controlado se marcará como fallido. El siguiente bloque de código de muestra detecta tal error y lo agrega como un error de ejecución.

```
try {
        await synthetics.takeScreenshot(stepName, "loaded");
    } catch(ex) {
        synthetics.addExecutionError('Unable to take screenshot ', ex);
    }
```

getCanaryName();

Devuelve el nombre del valor controlado.

getCanaryArn();

Devuelve el nombre del valor controlado.

getCanaryUserAgentString();

Devuelve el agente de usuario personalizado del valor controlado.

getRuntimeVersion();

Esta función está disponible en la versión de tiempo de ejecución `syn-nodejs-puppeteer-3.0` y en posteriores. Devuelve la versión de tiempo de ejecución de Synthetics del valor controlado. Por ejemplo, el valor de devuelto podría ser `syn-nodejs-puppeteer-3.0`.

getLogLevel();

Recupera el nivel de registro actual para la biblioteca de Synthetics. Los valores posibles son los siguientes:

- `0`: depuración
- `1`: información

- 2: advertencia
- 3: error

Ejemplo:

```
let logLevel = synthetics.getLogLevel();
```

setLogLevel();

Establece el nivel de registro de la biblioteca de Synthetics. Los valores posibles son los siguientes:

- 0: depuración
- 1: información
- 2: advertencia
- 3: error

Ejemplo:

```
synthetics.setLogLevel(0);
```

Clase de SyntheticsConfiguration

Esta clase solo está disponible en la versión de tiempo de ejecución de `syn-nodejs-2.1` o en posteriores.

La clase SyntheticsConfiguration puede utilizarse para configurar el comportamiento de las funciones de biblioteca de Synthetics. Por ejemplo, puede utilizar esta clase para configurar la función `executeStep()` para no tomar capturas de pantalla.

Pueden establecerse configuraciones de CloudWatch Synthetics a nivel global, que se aplican a todos los pasos de canaries. También se pueden anular estas configuraciones en el nivel de paso al pasar los pares clave-valor de configuración.

Se pueden pasar opciones en el nivel de paso. Para ver ejemplos, consulte async executeStep(stepName, functionToExecute, [stepConfig]); y executeHttpStep(stepName, requestOptions, [callback], [stepConfig]).

Definiciones de las funciones

setConfig(options)

`options` es un objeto, que es un conjunto de opciones configurables para el valor controlado. En las siguientes secciones se explican los posibles campos en *`options`*.

setConfig(options) para todos los canaries

Para los canaries que utilizan `syn-nodejs-puppeteer-3.2` o posteriores, las (options) (opciones) para setConfig pueden incluir los siguientes parámetros:

- `includeRequestHeaders` (booleano): si se deben incluir cabeceras de solicitud en el informe. El valor predeterminado es `false`.
- `includeResponseHeaders` (booleano): si se deben incluir cabeceras de respuesta en el informe. El valor predeterminado es `false`.
- `restrictedHeaders` (matriz): lista de valores de cabecera que se deben ignorar si se incluyen cabeceras. Esto aplica a las cabeceras de solicitud y respuesta. Por ejemplo, puede ocultar las credenciales al pasar includeRequestHeaders como `true` y restrictedHeaders como `['Authorization']`.
- `includeRequestBody` (booleano): si se debe incluir el cuerpo de la solicitud en el informe. El valor predeterminado es `false`.
- `includeResponseBody` (booleano): si se debe incluir el cuerpo de respuesta en el informe. El valor predeterminado es `false`.

setConfig(options) con respecto a las métricas de CloudWatch

Para los valores controlados que utilizan `syn-nodejs-puppeteer-3.1` o posteriores, las (opciones) para setConfig pueden incluir los siguientes parámetros booleanos que determinan qué métricas publica el valor controlado. El valor predeterminado para cada una de estas opciones es `true`. Las opciones que comienzan con `aggregated` determinan si la métrica se emite sin la dimensión `CanaryName`. Se pueden utilizar estas métricas para ver los resultados agregados de todos los canaries. Las otras opciones determinan si la métrica se emite con la dimensión `CanaryName`. Se pueden usar estas métricas para ver los resultados de cada valor controlado individualmente.

Para obtener una lista de las métricas de CloudWatch que los canaries emiten, consulte Métricas de CloudWatch que los canaries publican.

- `failedCanaryMetric` (booleano): si se debe emitir la métrica `Failed` (con la dimensión `CanaryName`) para este valor controlado. El valor predeterminado es `true`.
- `failedRequestsMetric` (booleano): si se debe emitir la métrica `Failed requests` (con la dimensión `CanaryName`) para este valor controlado. El valor predeterminado es `true`.
- `_2xxMetric` (booleano): si se debe emitir la métrica `2xx` (con la dimensión `CanaryName`) para este valor controlado. El valor predeterminado es `true`.
- `_4xxMetric` (booleano): si se debe emitir la métrica `4xx` (con la dimensión `CanaryName`) para este valor controlado. El valor predeterminado es `true`.
- `_5xxMetric` (booleano): si se debe emitir la métrica `5xx` (con la dimensión `CanaryName`) para este valor controlado. El valor predeterminado es `true`.
- `stepDurationMetric` (booleano): si se debe emitir la métrica `Step duration` (con las dimensiones `CanaryName` y `StepName`) para este valor controlado. El valor predeterminado es `true`.
- `stepSuccessMetric` (booleano): si se debe emitir la métrica `Step success` (con las dimensiones `CanaryName` y `StepName`) para este valor controlado. El valor predeterminado es `true`.
- `aggregatedFailedCanaryMetric` (booleano): si se debe emitir el métrica `Failed` (sin la dimensión `CanaryName`) para este valor controlado. El valor predeterminado es `true`.
- `aggregatedFailedRequestsMetric` (booleano): si se debe emitir el métrica `Failed Requests` (sin la dimensión `CanaryName`) para este valor controlado. El valor predeterminado es `true`.
- `aggregated2xxMetric` (booleano): si se debe emitir el métrica `2xx` (sin la dimensión `CanaryName`) para este valor controlado. El valor predeterminado es `true`.
- `aggregated4xxMetric` (booleano): si se debe emitir el métrica `4xx` (sin la dimensión `CanaryName`) para este valor controlado. El valor predeterminado es `true`.
- `aggregated5xxMetric` (booleano): si se debe emitir el métrica `5xx` (sin la dimensión `CanaryName`) para este valor controlado. El valor predeterminado es `true`.
- `visualMonitoringSuccessPercentMetric` (booleano): si se debe emitir la métrica `visualMonitoringSuccessPercent` para este valor controlado. El valor predeterminado es `true`.
- `visualMonitoringTotalComparisonsMetric` (booleano): si se debe emitir la métrica `visualMonitoringTotalComparisons` para este valor controlado. El valor predeterminado es `false`.

- `stepsReport` (booleano): si se debe informar de un resumen de ejecución de pasos. El valor predeterminado es `true`.
- `includeUrlPassword` (booleano): si se debe incluir una contraseña que aparezca en la dirección URL. De forma predeterminada, las contraseñas que aparecen en las direcciones URL se eliminan de los registros e informes para evitar que se divulgue información confidencial. El valor predeterminado es `false`.
- `restrictedUrlParameters` (matriz): lista de la ruta URL o los parámetros de consulta que se van a editar. Esto aplica a las URL que aparecen en registros, informes y en errores. El parámetro no distingue entre mayúsculas y minúsculas. Puede pasar un asterisco (*) como un valor para editar todos los valores de ruta de URL y los parámetros de consulta. El valor predeterminado es una matriz vacía.
- `logRequest` (booleano): si se debe registrar cada solicitud en los registros de valores controlados. Para canaries de UI, esto registra cada solicitud que le navegador envía. El valor predeterminado es `true`.
- `logResponse` (booleano): si se debe registrar cada respuesta en los registros de valores controlados. Para canaries de UI, esto registra todas las respuestas que el navegador recibe. El valor predeterminado es `true`.
- `logRequestBody` (booleano): si se deben registrar los cuerpos de la solicitud junto con las solicitudes en los registros de valores controlados. Esta configuración sólo aplica si `logRequest` es `true`. El valor predeterminado es `false`.
- `logResponseBody` (booleano): si se deben registrar los cuerpos de respuesta junto con las respuestas en los registros de valores controlados. Esta configuración sólo aplica si `logResponse` es `true`. El valor predeterminado es `false`.
- `logRequestHeaders` (booleano): si se deben registrar cabeceras de solicitud junto con las solicitudes en registros de valores controlados. Esta configuración sólo aplica si `logRequest` es `true`. El valor predeterminado es `false`.

  Debe tener en cuenta que `includeRequestHeaders` habilita cabeceras en artefactos.
- `logResponseHeaders` (booleano): si se deben registrar cabeceras de respuesta junto con las respuestas en los registros de valores controlados. Esta configuración sólo aplica si `logResponse` es `true`. El valor predeterminado es `false`.

  Debe tener en cuenta que `includeResponseHeaders` habilita encabezados en artefactos.

> **Note**
>
> Las métricas de `Duration` y de `SuccessPercent` se emiten siempre para cada valor controlado con la métrica `CanaryName` y sin ella.

Métodos para habilitar o desactivar métricas

disableAggregatedRequestMetrics()

Desactiva que el valor controlado emita todas las métricas de solicitud que se emiten sin dimensión `CanaryName`.

disableRequestMetrics()

Deshabilita todas las métricas de solicitud, incluidas las métricas por valor controlado y las métricas agregadas en todos los valores controlados.

disableStepMetrics()

Desactiva todas las métricas de pasos, incluidas las métricas de éxito y de duración de los pasos.

enableAggregatedRequestMetrics()

Permite que el valor controlado emita todas las métricas de solicitud que se emiten sin dimensión `CanaryName`.

enableRequestMetrics()

Habilita todas las métricas de solicitud, incluidas las métricas por valor controlado y las métricas agregadas en todos los valores controlados.

enableStepMetrics()

Habilita todas las métricas de pasos, incluidas las métricas de éxito y de duración de los pasos.

get2xxMetric()

Muestra si el valor controlado emite una métrica `2xx` con la dimensión `CanaryName`.

get4xxMetric()

Muestra si el valor controlado emite una métrica `4xx` con la dimensión `CanaryName`.

get5xxMetric()

Muestra si el valor controlado emite una métrica `5xx` con la dimensión `CanaryName`.

getAggregated2xxMetric()

Muestra si el valor controlado emite una métrica `2xx` sin dimensión.

getAggregated4xxMetric()

Muestra si el valor controlado emite una métrica `4xx` sin dimensión.

getAggregatedFailedCanaryMetric()

Muestra si el valor controlado emite una métrica `Failed` sin dimensión.

getAggregatedFailedRequestsMetric()

Muestra si el valor controlado emite una métrica `Failed requests` sin dimensión.

getAggregated5xxMetric()

Muestra si el valor controlado emite una métrica `5xx` sin dimensión.

getFailedCanaryMetric()

Muestra si el valor controlado emite una métrica `Failed` con la dimensión `CanaryName`.

getFailedRequestsMetric()

Muestra si el valor controlado emite una métrica `Failed requests` con la dimensión `CanaryName`.

getStepDurationMetric()

Muestra si el valor controlado emite una métrica `Duration` con la dimensión `CanaryName` para este valor controlado.

getStepSuccessMetric()

Muestra si el valor controlado emite una métrica `StepSuccess` con la dimensión `CanaryName` para este valor controlado.

with2xxMetric(_2xxMetric)

Acepta un argumento booleano, que especifica si se emitirá una métrica `2xx` con la dimensión `CanaryName` para este valor controlado.

with4xxMetric(_4xxMetric)

Acepta un argumento booleano, que especifica si se emitirá una métrica `4xx` con la dimensión `CanaryName` para este valor controlado.

with5xxMetric(_5xxMetric)

Acepta un argumento booleano, que especifica si se emitirá una métrica `5xx`con la dimensión `CanaryName` para este valor controlado.

withAggregated2xxMetric(aggregated2xxMetric)

Acepta un argumento booleano, que especifica si se emitirá una métrica `2xx`sin dimensión para este valor controlado.

withAggregated4xxMetric(aggregated4xxMetric)

Acepta un argumento booleano, que especifica si se emitirá una métrica `4xx` sin dimensión para este valor controlado.

withAggregated5xxMetric(aggregated5xxMetric)

Acepta un argumento booleano, que especifica si se emitirá una métrica `5xx` sin dimensión para este valor controlado.

withAggregatedFailedCanaryMetric(aggregatedFailedCanaryMetric)

Acepta un argumento booleano, que especifica si se emitirá una métrica `Failed` sin dimensión para este valor controlado.

withAggregatedFailedRequestsMetric(aggregatedFailedRequestsMetric)

Acepta un argumento booleano, que especifica si se emitirá una métrica `Failed requests` sin dimensión para este valor controlado.

withFailedCanaryMetric(failedCanaryMetric)

Acepta un argumento booleano, que especifica si se emitirá una métrica `Failed` con la dimensión `CanaryName` para este valor controlado.

withFailedRequestsMetric(failedRequestsMetric)

Acepta un argumento booleano, que especifica si se emitirá una métrica `Failed requests` con la dimensión `CanaryName` para este valor controlado.

withStepDurationMetric(stepDurationMetric)

Acepta un argumento booleano, que especifica si se emitirá una métrica `Duration` con la dimensión `CanaryName` para este canary.

withStepSuccessMetric(stepSuccessMetric)

Acepta un argumento booleano, que especifica si se emitirá una métrica `StepSuccess` con la dimensión `CanaryName` para este valor controlado.

Métodos para habilitar o desactivar otras características

withHarFile()

Acepta un argumento booleano, que especifica si se debe crear un archivo HAR para este valor controlado.

withStepsReport()

Acepta un argumento booleano, que especifica si se debe informar de un resumen de ejecución de pasos para este valor controlado.

withIncludeUrlPassword()

Acepta un argumento booleano, que especifica si se deben incluir las contraseñas que aparecen en las URL de los registros e informes.

withRestrictedUrlParameters()

Acepta una matriz de ruta de URL o parámetros de consulta para editar. Aplica a las URL que aparecen en registros, informes y en errores. Se puede pasar un asterisco (*) como un valor para redactar todos los valores de ruta de URL y los parámetros de consulta

withLogRequest()

Acepta un argumento booleano, que especifica si se debe registrar cada solicitud en los registros del valor controlado.

withLogResponse()

Acepta un argumento booleano, que especifica si se debe registrar cada respuesta en los registros del valor controlado.

withLogRequestBody()

Acepta un argumento booleano, que especifica si se debe registrar cada cuerpo de la solicitud en los registros del valor controlado.

withLogResponseBody()

Acepta un argumento booleano, que especifica si se debe registrar cada cuerpo de respuesta en los registros del valor controlado.

withLogRequestHeaders()

Acepta un argumento booleano, que especifica si se debe registrar cada cabecera de solicitud en los registros del valor controlado.

withLogResponseHeaders()

Acepta un argumento booleano, que especifica si se debe registrar cada cabecera de respuesta en los registros del valor controlado.

getHarFile()

Muestra si el valor controlado crea un archivo HAR.

getStepsReport()

Muestra si el valor controlado informa un resumen de ejecución de pasos.

getIncludeUrlPassword()

Muestra si el valor controlado incluye contraseñas que aparecen en las URL en los registros e informes.

getRestrictedUrlParameters()

Muestra si el valor controlado redacta la ruta de URL o los parámetros de consulta.

getLogRequest()

Muestra si el valor controlado registra cada solicitud en los registros del valor controlado.

getLogResponse()

Muestra si el valor controlado registra cada respuesta en los registros del valor controlado.

getLogRequestBody()

Muestra si el valor controlado registra cada cuerpo de la solicitud en los registros del valor controlado.

getLogResponseBody()

Muestra si el valor controlado registra cada cuerpo de respuesta en los registros del valor controlado.

getLogRequestHeaders()

Muestra si el valor controlado registra cada cabecera de solicitud en los registros del valor controlado.

getLogResponseHeaders()

Muestra si el valor controlado registra cada cabecera de respuesta en los registros del valor controlado.

Funciones para todos los valores controlados

- `withIncludeRequestHeaders`(includeRequestHeaders)
- `withIncludeResponseHeaders`(includeResponseHeaders)
- `withRestrictedHeaders`(restrictedHeaders)
- `withIncludeRequestBody`(includeRequestBody)
- `withIncludeResponseBody`(includeResponseBody)
- `enableReportingOptions`(): habilita todas las opciones de informes-- includeRequestHeaders, includeResponseHeaders, includeRequestBody, y includeResponseBody.
- `disableReportingOptions`(): desactiva todas las opciones de informes-- includeRequestHeaders, includeResponseHeaders, includeRequestBody, y includeResponseBody.

setConfig(options) para canaries de la UI

Para canaries de la UI, setConfig puede incluir los siguientes parámetros booleanos:

- `continueOnStepFailure` (booleano): si se debe continuar con la ejecución del script valor controlado después de que un paso falle (esto se refiere a la función executeStep). Si algún paso falla, la ejecución del valor controlado seguirá marcándose como fallida. El valor predeterminado es `false`.
- `harFile` (booleano): si se crea un archivo HAR. El valor predeterminado es `True`.
- `screenshotOnStepStart` (booleano): si se toma una captura de pantalla antes de comenzar un paso.
- `screenshotOnStepSuccess` (booleano): si se debe tomar una captura de pantalla después de completar un paso correctamente.
- `screenshotOnStepFailure` (booleano): si se toma una captura de pantalla después de que un paso falla.

Métodos para habilitar o desactivar las capturas de pantalla

disableStepScreenshots()

Deshabilita todas las opciones de captura de pantalla (screenshotOnStepStart, screenshotOnStepSuccess, y screenshotOnStepFailure).

enableStepScreenshots()

Habilita todas las opciones de captura de pantalla (screenshotOnStepStart, screenshotOnStepSuccess, y screenshotOnStepFailure). Estos métodos no están habilitados de forma predeterminada.

getScreenshotOnStepFailure()

Muestra si el valor controlado toma una captura de pantalla después de que un paso falla.

getScreenshotOnStepStart()

Muestra si el valor controlado toma una captura de pantalla antes de iniciar un paso.

getScreenshotOnStepSuccess()

Muestra si el valor controlado toma una captura de pantalla después de completar un paso correctamente.

withScreenshotOnStepStart(screenshotOnStepStart)

Acepta un argumento booleano, que indica si se debe tomar una captura de pantalla antes de iniciar un paso.

withScreenshotOnStepSuccess(screenshotOnStepSuccess)

Acepta un argumento booleano, que indica si se debe tomar una captura de pantalla después de completar un paso correctamente.

withScreenshotOnStepFailure(screenshotOnStepFailure)

Acepta un argumento booleano, que indica si se debe tomar una captura de pantalla después de que un paso falla.

Uso en valores controlados de la IU

Primero, importe la relación de Synthetics y obtenga la configuración.

```
// Import Synthetics dependency
const synthetics = require('Synthetics');

// Get Synthetics configuration
const synConfig = synthetics.getConfiguration();
```

A continuación, establezca la configuración para cada opción mediante llamadas al método SetConfig con una de las siguientes opciones.

```
// Set configuration values
    synConfig.setConfig({
        screenshotOnStepStart: true,
        screenshotOnStepSuccess: false,
        screenshotOnStepFailure: false
    });
```

Or (Disyunción)

```
synConfig.withScreenshotOnStepStart(false).withScreenshotOnStepSuccess(true).withScreenshotOnSt
```

Para desactivar todas las capturas de pantalla, utilice la función disableStepScreenshots() como en este ejemplo.

```
synConfig.disableStepScreenshots();
```

Puede habilitar y desactivar las capturas de pantalla en cualquier punto del código. Por ejemplo, para desactivar las capturas de pantalla solo para un paso, se deben desactivar antes de ejecutar ese paso y habilitarlas después del paso.

setConfig(options) para canaries de la API

Para los canaries de la API, setConfig puede incluir los siguientes parámetros booleanos:

- `continueOnHttpStepFailure` (booleano): si se continúa con la ejecución del script valor controlado después de que se produce un error en un paso HTTP (esto se refiere a la función executeHttpStep). Si algún paso falla, la ejecución del valor controlado seguirá marcándose como fallida. El valor predeterminado es `true`.

Supervisión visual

La supervisión visual compara las capturas de pantalla que se toman durante una ejecución de un valor controlado con las capturas de pantalla que se toman durante una ejecución de un valor controlado de línea de base. Si la discrepancia entre las dos capturas de pantalla está más allá de un porcentaje umbral, el valor controlado falla y se podrán ver las áreas con diferencias resaltadas en color en el informe de ejecución del valor controlado. La supervisión visual es compatible con canaries que ejecutan syn-puppeteer-node-3.2 y posteriores. Por el momento no es compatible con canaries que ejecutan Python y Selenium.

Para habilitar la supervisión visual, agregue la siguiente línea de código al script valor controlado. Para obtener más información, consulte Clase de SyntheticsConfiguration.

```
syntheticsConfiguration.withVisualCompareWithBaseRun(true);
```

La primera vez que el valor controlado se ejecuta correctamente después de agregar esta línea al script, utiliza las capturas de pantalla que se tomaron durante esa ejecución como línea de base para la comparación. Después de la primera ejecución del valor controlado, se puede usar la consola de CloudWatch para editar el valor controlado para realizar cualquiera de las siguientes acciones:

- Establecer la siguiente ejecución del valor controlado como la nueva línea de base.

- Establecer límites en la captura de pantalla de línea de base actual para designar áreas de la captura de pantalla que se ignorarán durante las comparaciones visuales.
- Eliminar una captura de pantalla de ser utilizada para la supervisión visual.

Para obtener más información sobre cómo usar la consola de CloudWatch para editar un valor controlado, consulte Edición o eliminación de un valor controlado.

Otras opciones para la supervisión visual

syntheticsConfiguration.withVisualVarianceThresholdPercentage(desiredPercentage)

Establezca el porcentaje aceptable para la desviación de captura de pantalla en las comparaciones visuales.

syntheticsConfiguration.withVisualVarianceHighlightHexColor("#fafa00")

Establezca el color de resaltado que designa las áreas de desviación cuando vea los informes de ejecución del valor controlado que utilizan supervisión visual.

syntheticsConfiguration.withFailCanaryRunOnVisualVariance(failCanary)

Establezca si el valor controlado falla o no cuando hay una diferencia visual superior al umbral. El valor predeterminado es que el valor controlado falle.

Registrador de Synthetics

SyntheticsLogger escribe registros tanto en la consola como en un archivo de registro local, en el mismo nivel de registro. Este archivo de registro se escribe en ambas ubicaciones solo si el nivel de registro coincide con el deseado para la función de registro a la que se llamó o está por debajo de este.

Los valores "DEBUG: ", "INFO: ", etc. se anteponen a las instrucciones de registro del archivo de registro local para que coincidan con el nivel de registro de la función a la que se llamó.

Puede utilizar SyntheticsLogger si desea ejecutar la biblioteca de Synthetics en el mismo nivel de registro que el registro de valores controlados de Synthetics.

No es necesario que se utilice SyntheticsLogger para crear un archivo de registros que se carga en la ubicación de resultados de S3. En su lugar, puede crear un archivo de registro distinto en la

carpeta `/tmp`. Los archivos creados en la carpeta `/tmp` se cargan en la ubicación de resultados de S3 como artefactos.

Para utilizar el registrador de la biblioteca de Synthetics:

```
const log = require('SyntheticsLogger');
```

Definiciones de funciones útiles:

log.debug(***mensaje***, ***ex***);

Parámetros: *mensaje* es el mensaje que se va a registrar. *ex* es la excepción que se registra, si la hay

Ejemplo:

```
log.debug("Starting step - login.");
```

log.error(***mensaje***, ***ex***);

Parámetros: *mensaje* es el mensaje que se va a registrar. *ex* es la excepción que se registra, si la hay

Ejemplo:

```
try {
  await login();
catch (ex) {
  log.error("Error encountered in step - login.", ex);
}
```

log.info(***mensaje***, ***ex***);

Parámetros: *mensaje* es el mensaje que se va a registrar. *ex* es la excepción que se registra, si la hay

Ejemplo:

```
log.info("Successfully completed step - login.");
```

log.log(***mensaje***, ***ex***);

Este es un alias para `log.info`.

Parámetros: *`mensaje`* es el mensaje que se va a registrar. *`ex`* es la excepción que se registra, si la hay

Ejemplo:

```
log.log("Successfully completed step - login.");
```

log.warn(***mensaje***, ***ex***);

Parámetros: *`mensaje`* es el mensaje que se va a registrar. *`ex`* es la excepción que se registra, si la hay

Ejemplo:

```
log.warn("Exception encountered trying to publish CloudWatch Metric.", ex);
```

Clase de SyntheticSloghelper

La clase `SyntheticsLogHelper` está disponible en el tiempo de ejecución `syn-nodejs-puppeteer-3.2` y en tiempos de ejecución posteriores. Ya está inicializado en la biblioteca CloudWatch Synthetics y está configurado con la configuración de Synthetics. Puede agregar esto como una relación en el script. Esta clase le permite borrar las URL, encabezados y mensajes de error para redactar información confidencial.

> **Note**
>
> Synthetics sanitiza todas las URL y los mensajes de error que registra antes de incluirlos en los registros, informes, archivos HAR y errores de ejecución de los valores controlados basados en la configuración `restrictedUrlParameters` de Synthetics. Tiene que usar `getSanitizedUrl` o `getSanitizedErrorMessage` solo si está registrando direcciones URL o errores en el script. Synthetics no almacena ningún artefacto de valores controlados excepto los errores de valores controlados que el script lanza. Los artefactos de ejecución de valores controlados se almacenan en la cuenta del cliente. Para obtener más información, consulte Consideraciones de seguridad para los canaries de Synthetics.

getSanitizedUrl(url, stepConfig = null)

Esta función está disponible en `syn-nodejs-puppeteer-3.2` y en posteriores. Devuelve cadenas de url sanitizadas basadas en la configuración. Puede optar por eliminar los parámetros de las URL confidenciales como la contraseña y el access_token al establecer la propiedad `restrictedUrlParameters`. De forma predeterminada, las contraseñas de las URL se eliminan. Si es necesario, puede habilitar las contraseñas de las URL si configura `includeUrlPassword` a verdadero.

Esta función arroja un error si la URL pasada no es una URL válida.

Parámetros

- Una *`url`* es una cadena y es la URL para sanitizar.
- *`stepConfig`* (Opcional) anula la configuración global de Synthetics para esta función. Si `stepConfig` no se especifica, la configuración global se utiliza para sanitizar la URL.

Ejemplo

En este ejemplo se usa la siguiente URL de ejemplo: `https://example.com/learn/home?access_token=12345&token_type=Bearer&expires_in=1200`. En este ejemplo, `access_token` contiene su información confidencial que no debe registrarse. Debe tener en cuenta que los servicios de Synthetics no almacenan ningún artefacto de ejecución de valores controlados. Los artefactos como registros, capturas de pantalla e informes se almacenan en un bucket de Amazon S3 de la cuenta de cliente.

El primer paso es configurar la configuración de Synthetics.

```
// Import Synthetics dependency
const synthetics = require('Synthetics');

// Import Synthetics logger for logging url
const log = require('SyntheticsLogger');

// Get Synthetics configuration
const synConfig = synthetics.getConfiguration();

// Set restricted parameters
synConfig.setConfig({
   restrictedUrlParameters: ['access_token'];
```

```
});
```

A continuación, sanitice y registre la URL

```
// Import SyntheticsLogHelper dependency
const syntheticsLogHelper = require('SyntheticsLogHelper');

const sanitizedUrl = synthetics.getSanitizedUrl('https://example.com/learn/home?
access_token=12345&token_type=Bearer&expires_in=1200');
```

Esto registra lo siguiente en el registro del valor controlado.

```
My example url is: https://example.com/learn/home?
access_token=REDACTED&token_type=Bearer&expires_in=1200
```

Puede anular la configuración de Synthetics para una URL si especifica un parámetro opcional que contenga las opciones de configuración de Synthetics, como en el siguiente ejemplo .

```
const urlConfig = {
   restrictedUrlParameters = ['*']
};
const sanitizedUrl = synthetics.getSanitizedUrl('https://example.com/learn/home?
access_token=12345&token_type=Bearer&expires_in=1200', urlConfig);
logger.info('My example url is: ' + sanitizedUrl);
```

El ejemplo anterior elimina todos los parámetros de consulta y se registra de la siguiente manera:

```
My example url is: https://example.com/learn/home?
access_token=REDACTED&token_type=REDACTED&expires_in=REDACTED
```

getSanitizedErrorMessage

Esta función está disponible en `syn-nodejs-puppeteer-3.2` y en posteriores. Devuelve cadenas de error sanitizadas al sanitizar cualquier URL presente en función de la configuración de Synthetics. Puede optar por anular la configuración global de Synthetics cuando llame a esta función mediante la especificación de un parámetro `stepConfig`.

Parámetros

- *`error`* es el error para sanitizar. Puede ser un objeto Error o una cadena.

- *`stepConfig`* (Opcional) anula la configuración global de Synthetics para esta función. Si `stepConfig` no se especifica, la configuración global se utiliza para sanitizar la URL.

Ejemplo

En este ejemplo se utiliza el siguiente error: `Failed to load url: https://example.com/learn/home?access_token=12345&token_type=Bearer&expires_in=1200`

El primer paso es configurar la configuración de Synthetics.

```
// Import Synthetics dependency
const synthetics = require('Synthetics');

// Import Synthetics logger for logging url
const log = require('SyntheticsLogger');

// Get Synthetics configuration
const synConfig = synthetics.getConfiguration();

// Set restricted parameters
synConfig.setConfig({
   restrictedUrlParameters: ['access_token'];
});
```

A continuación, sanitice y registre el mensaje de error

```
// Import SyntheticsLogHelper dependency
const syntheticsLogHelper = require('SyntheticsLogHelper');

try {
   // Your code which can throw an error containing url which your script logs
} catch (error) {
    const sanitizedErrorMessage = synthetics.getSanitizedErrorMessage(errorMessage);
    logger.info(sanitizedErrorMessage);
}
```

Esto registra lo siguiente en el registro del valor controlado.

```
Failed to load url: https://example.com/learn/home?
access_token=REDACTED&token_type=Bearer&expires_in=1200
```

getSanitizedHeaders(headers, stepConfig=null)

Esta función está disponible en `syn-nodejs-puppeteer-3.2` y en posteriores. Devuelve encabezados sanitizados basados en la propiedad `restrictedHeaders` de `syntheticsConfiguration`. Los encabezados especificados en la propiedad `restrictedHeaders` se editan a partir de registros, archivos HAR e informes.

Parámetros

- *`headers`* (cabeceras) es un objeto que contiene las cabeceras para desinfectar.
- *`stepConfig`* (Opcional) anula la configuración global de Synthetics para esta función. Si `stepConfig` no se especifica, la configuración global se utiliza para desinfectar las cabeceras.

Funciones y clases de biblioteca Node.js que solo se aplican a los canaries de la UI

Las siguientes funciones de la biblioteca de CloudWatch Synthetics solo son útiles para los canaries de la UI.

Temas

- Clase de Synthetics
- Clase BrokenLinkCheckerReport
- Clase SyntheticsLink

Clase de Synthetics

Las siguientes funciones están en la clase de Synthetics.

async addUserAgent(page, userAgentString);

Esta función añade *`userAgentString`* al encabezado de agente de usuario de la página especificada.

Ejemplo:

```
await synthetics.addUserAgent(page, "MyApp-1.0");
```

Los resultados del encabezado del agente de usuario de la página se establecen en *`browsers-user-agent-header-value`*`MyApp-1.0`

async executeStep(stepName, functionToExecute, [stepConfig]);

Ejecuta el paso proporcionado y lo integra con iniciar/superar/fallar el registro, iniciar/superar/fallar capturas de pantalla, superar/fallar y métricas de duración.

> **Note**
>
> Si utiliza el `syn-nodejs-2.1` o una versión posterior de tiempo de ejecución, puede configurar si se toman capturas de pantalla y cuándo. Para obtener más información, consulte Clase de SyntheticsConfiguration.

La función `executeStep` también hace lo siguiente:

- Registra que el paso se ha iniciado.
- Toma una captura de pantalla denominada `<stepName>-starting`.
- Inicia un temporizador.
- Ejecuta la función proporcionada.
- Si la función devuelve resultados normalmente, cuenta como superada. Si la función falla, cuenta como error.
- Finaliza el temporizador.
- Registra si el paso se ha superado o no.
- Toma una captura de pantalla denominada `<stepName>-succeeded` o `<stepName>-failed`.
- Emite la métrica `stepName SuccessPercent`, 100 para superado o 0 para no superado.
- Emite la métrica `stepName Duration`, con un valor basado en las horas de inicio y de finalización del paso.
- Por último, devuelve el mismo resultado que `functionToExecute` o vuelve a arrojar el mismo error que `functionToExecute`.

Si el valor controlado utiliza el tiempo de ejecución `syn-nodejs-2.0` o uno posterior, esta función también agrega un resumen de ejecución de pasos al informe del valor controlado. El resumen incluye detalles acerca de cada paso, como la hora de inicio, la hora de finalización, el estado (SUPERADO o NO SUPERADO), el motivo del error (si hubo) y las capturas de pantalla que se tomaron durante la ejecución de cada paso.

Ejemplo:

```
await synthetics.executeStep('navigateToUrl', async function (timeoutInMillis = 30000)
 {
           await page.goto(url, {waitUntil: ['load', 'networkidle0'], timeout:
 timeoutInMillis});});
```

Respuesta:

Devuelve el mismo resultado que `functionToExecute`.

Actualizaciones con syn-nodejs-2.2

A partir de `syn-nodejs-2.2`, se pueden pasar opcionalmente configuraciones de pasos para anular las configuraciones de CloudWatch Synthetics en el nivel de pasos. Para obtener una lista de opciones que puede pasar a `executeStep`, consulte Clase de SyntheticsConfiguration.

En el siguiente ejemplo se anula la configuración predeterminada `false` para `continueOnStepFailure` a `true` y se especifica cuándo tomar capturas de pantalla.

```
var stepConfig = {
    'continueOnStepFailure': true,
    'screenshotOnStepStart': false,
    'screenshotOnStepSuccess': true,
    'screenshotOnStepFailure': false
}

await executeStep('Navigate to amazon', async function (timeoutInMillis = 30000) {
      await page.goto(url, {waitUntil: ['load', 'networkidle0'], timeout:
 timeoutInMillis});
 }, stepConfig);
```

getDefaultLaunchOptions();

La función `getDefaultLaunchOptions()` muestra los resultados de las opciones de lanzamiento del navegador que CloudWatch Synthetics utiliza. Para más información, consulte Tipo de opciones de lanzamiento

```
// This function returns default launch options used by Synthetics.
const defaultOptions = await synthetics.getDefaultLaunchOptions();
```

getPage();

Devuelve la página abierta actual como objeto de Puppeteer. Para obtener más información, consulte Puppeteer API v1.14.0.

Ejemplo:

```
let page = synthetics.getPage();
```

Respuesta:

La página (objeto de Puppeteer) que está abierta en la sesión del explorador actual.

getRequestResponseLogHelper();

> **Important**
>
> En los canaries que utilizan el tiempo de ejecución `syn-nodejs-puppeteer-3.2` o uno posterior, esta función está obsoleta junto con la clase `RequestResponseLogHelper`. El uso de esta función hace que aparezca una advertencia en los registros de los valores controlados. Esta función se eliminará en versiones futuras de tiempo de ejecución. Si está utilizando esta función, utilice en su lugar RequestResponseLogHelper class.

Utilice esta función como patrón generador para modificar las marcas de registro de solicitudes y respuestas.

Ejemplo:

```
synthetics.setRequestResponseLogHelper(getRequestResponseLogHelper().withLogRequestHeaders(fals
```

Respuesta:

```
{RequestResponseLogHelper}
```

Lanzamiento (opciones)

Las opciones para esta función sólo están disponibles en la versión de tiempo de ejecución `syn-nodejs-2.1` o en posteriores.

Esta función solo se usa para los canaries de la UI. Cierra el navegador existente y lanza uno nuevo.

> **Note**
>
> CloudWatch Synthetics siempre lanza un navegador antes de comenzar a ejecutar el script. No es necesario que llame un lanzamiento() a menos que desee lanzar un navegador nuevo con opciones personalizadas.

(opciones) es un conjunto configurable de opciones para configurar en el navegador. Para más información, consulte Tipo de opciones de lanzamiento.

Si llama a esta función sin opciones, Synthetics lanza un navegador con argumentos predeterminados, `executablePath` y `defaultViewport`. La ventana gráfica predeterminada en CloudWatch Synthetics es 1920 x 1080.

Se pueden anular los parámetros de lanzamiento que CloudWatch Synthetics utiliza y pasar parámetros adicionales al lanzar el navegador. Por ejemplo, el siguiente fragmento de código inicia un navegador con argumentos predeterminados y una ruta ejecutable predeterminada, pero con una ventana gráfica de 800 x 600.

```
await synthetics.launch({
        defaultViewport: {
            "deviceScaleFactor": 1,
            "width": 800,
            "height": 600
    }});
```

En el siguiente código de muestra se agrega un nuevo parámetro `ignoreHTTPSErrors` a los parámetros de lanzamiento de CloudWatch Synthetics:

```
await synthetics.launch({
        ignoreHTTPSErrors: true
 });
```

Puede desactivar la seguridad web si agrega un indicador `--disable-web-security` a los argumentos en los parámetros de lanzamiento de CloudWatch Synthetics:

```
// This function adds the --disable-web-security flag to the launch parameters
const defaultOptions = await synthetics.getDefaultLaunchOptions();
```

```
const launchArgs = [...defaultOptions.args, '--disable-web-security'];
await synthetics.launch({
     args: launchArgs
  });
```

RequestResponseLogHelper class

> **Important**
>
> En los canaries que utilizan el tiempo de ejecución `syn-nodejs-puppeteer-3.2` o posteriores, esta clase está obsoleta. El uso de esta clase hace que aparezca una advertencia en los registros de los valores controlados. Esta función se eliminará en versiones futuras de tiempo de ejecución. Si está utilizando esta función, utilice en su lugar RequestResponseLogHelper class.

Controla la configuración y la creación en detalle de las representaciones de cadena de cargas de solicitud y respuesta.

```
class RequestResponseLogHelper {

    constructor () {
        this.request = {url: true, resourceType: false, method: false, headers: false,
 postData: false};
        this.response = {status: true, statusText: true, url: true, remoteAddress:
 false, headers: false};
    }

    withLogRequestUrl(logRequestUrl);

    withLogRequestResourceType(logRequestResourceType);

    withLogRequestMethod(logRequestMethod);

    withLogRequestHeaders(logRequestHeaders);

    withLogRequestPostData(logRequestPostData);


    withLogResponseStatus(logResponseStatus);

    withLogResponseStatusText(logResponseStatusText);
```

```
    withLogResponseUrl(logResponseUrl);

    withLogResponseRemoteAddress(logResponseRemoteAddress);

    withLogResponseHeaders(logResponseHeaders);
```

Ejemplo:

```
synthetics.setRequestResponseLogHelper(getRequestResponseLogHelper()
.withLogRequestPostData(true)
.withLogRequestHeaders(true)
.withLogResponseHeaders(true));
```

Respuesta:

```
{RequestResponseLogHelper}
```

setRequestResponseLogHelper();

> **Important**
>
> En los canaries que utilizan el tiempo de ejecución `syn-nodejs-puppeteer-3.2` o posteriores, esta función está obsoleta junto con la clase `RequestResponseLogHelper`. El uso de esta función hace que aparezca una advertencia en los registros de los valores controlados. Esta función se eliminará en versiones futuras de tiempo de ejecución. Si está utilizando esta función, utilice en su lugar RequestResponseLogHelper class.

Utilice esta función como patrón generador para establecer las marcas de registro de solicitudes y respuestas.

Ejemplo:

```
synthetics.setRequestResponseLogHelper().withLogRequestHeaders(true).withLogResponseHeaders(tru
```

Respuesta:

```
{RequestResponseLogHelper}
```

async takeScreenshot(name, suffix);

Toma una captura de pantalla (.PNG) de la página actual con nombre y un sufijo (opcional).

Ejemplo:

```
await synthetics.takeScreenshot("navigateToUrl", "loaded")
```

Este ejemplo captura y carga una captura de pantalla denominada `01-navigateToUrl-loaded.png` al bucket de S3 del valor controlado.

Puede tomar una captura de pantalla para un paso del valor controlado en particular al pasar `stepName` como primer parámetro. Las capturas de pantalla están vinculadas al paso del valor controlado en los informes, para ayudarle a realizar un rastreo de cada paso durante la depuración.

Los valores controlados de CloudWatch Synthetics toman capturas de pantalla automáticamente antes de comenzar un paso (la función `executeStep`) y después de la finalización del paso (a menos que configure el valor controlado para desactivar las capturas de pantalla). Puede tomar más capturas de pantalla si pasa el nombre del paso en la función `takeScreenshot`.

El siguiente ejemplo toma una captura de pantalla con `signupForm` como el valor de `stepName`. La captura de pantalla se denominará `02-signupForm-address` y se vinculará al paso denominado `signupForm` en el informe del valor controlado.

```
await synthetics.takeScreenshot('signupForm', 'address')
```

Clase BrokenLinkCheckerReport

Esta clase proporciona métodos para agregar un enlace Synthetics. Solo se admite en canaries que utilizan la versión `syn-nodejs-2.0-beta` de tiempo de ejecución o posteriores.

Para utilizar `BrokenLinkCheckerReport`, incluya las siguientes líneas en el script:

```
const BrokenLinkCheckerReport = require('BrokenLinkCheckerReport');

const brokenLinkCheckerReport = new BrokenLinkCheckerReport();
```

Definiciones de funciones útiles:

addLink(***syntheticsLink***, isBroken)

`syntheticsLink` es un objeto `SyntheticsLink` que representa un enlace. Esta función agrega el enlace de acuerdo con el código de estado. De forma predeterminada, considera que un enlace se rompe si el código de estado no está disponible o si el código de estado es 400 o superior. Puede anular este comportamiento predeterminado si pasa el parámetro opcional `isBrokenLink` con un valor de `true` o `false`.

Esta función no tiene un valor de retorno.

getLinks()

Esta función muestra los resultados de una matriz de los objetos `SyntheticsLink` que se incluyen en el informe del verificador de enlaces que no funcionan.

getTotalBrokenLinks()

Esta función muestra los resultados de un número que representa el número total de enlaces que no funcionan.

getTotalLinksChecked()

Esta función muestra los resultados de un número que representa el número total de enlaces incluidos en el informe.

Cómo utilizar BrokenLinkCheckerReport

El siguiente fragmento de código de script valor controlado muestra un ejemplo de navegación a un enlace que se agrega al informe del verificador de enlaces que no funcionan.

1. Importe `SyntheticsLink`, `BrokenLinkCheckerReport`, y `Synthetics`.

```
const BrokenLinkCheckerReport = require('BrokenLinkCheckerReport');
const SyntheticsLink = require('SyntheticsLink');

// Synthetics dependency
const synthetics = require('Synthetics');
```

2. Para agregar un enlace al informe, cree una instancia de `BrokenLinkCheckerReport`.

```
let brokenLinkCheckerReport = new BrokenLinkCheckerReport();
```

3. Desplácese hasta la URL y agréguela al informe del verificador de vínculos que no funcionan.

```
let url = "https://amazon.com";
```

```
let syntheticsLink = new SyntheticsLink(url);

// Navigate to the url.
let page = await synthetics.getPage();

// Create a new instance of Synthetics Link
let link = new SyntheticsLink(url)

try {
    const response = await page.goto(url, {waitUntil: 'domcontentloaded', timeout:
 30000});
} catch (ex) {
    // Add failure reason if navigation fails.
    link.withFailureReason(ex);
}

if (response) {
    // Capture screenshot of destination page
    let screenshotResult = await synthetics.takeScreenshot('amazon-home', 'loaded');

    // Add screenshot result to synthetics link
    link.addScreenshotResult(screenshotResult);

    // Add status code and status description to the link
    link.withStatusCode(response.status()).withStatusText(response.statusText())
}

// Add link to broken link checker report.
brokenLinkCheckerReport.addLink(link);
```

4. Agregue el informe a Synthetics. Esto crea un archivo JSON llamado `BrokenLinkCheckerReport.json` en el bucket de S3 para cada ejecución del valor controlado. Se puede ver un informe de enlaces en la consola para cada ejecución de valores controlados junto con capturas de pantalla, registros y archivos HAR.

```
await synthetics.addReport(brokenLinkCheckerReport);
```

Clase SyntheticsLink

Esta clase proporciona métodos para ajustar la información. Solo se admite en canaries que usan la versión `syn-nodejs-2.0-beta` de tiempo de ejecución o posteriores.

Para utilizar `SyntheticsLink`, incluya las siguientes líneas en el script:

```
const SyntheticsLink = require('SyntheticsLink');

const syntheticsLink = new SyntheticsLink("https://www.amazon.com");
```

La función muestra los resultados de `syntheticsLink`*`Object`*.

Definiciones de funciones útiles:

withUrl(***`url`***)

`url` es una cadena de URL. La función muestra los resultados sobre `syntheticsLink`*`Object`*.

withText(***`text`***)

`text` es una cadena que representa el texto de anclaje. La función muestra los resultados de `syntheticsLink`*`Object`*. Añade texto de anclaje correspondiente al enlace.

withParentUrl(***`parentUrl`***)

`parentUrl` es una cadena que representa la URL principal (página fuente). La función muestra los resultados de `syntheticsLink`*`Object`*.

withStatusCode(***`statusCode`***)

`statusCode` es una cadena que representa el código de estado. La función muestra los resultados de `syntheticsLink`*`Object`*.

withFailureReason(***`failureReason`***)

`failureReason` es una cadena que representa la causa del error. La función muestra los resultados de `syntheticsLink`*`Object`*.

addScreenshotResult(***`screenshotResult`***)

`screenshotResult` es un objeto. Es una instancia de `ScreenshotResult` que la función de Synthetics `takeScreenshot` ha mostrado. El objeto incluye lo siguiente:

- `fileName`— Una cadena que representa el `screenshotFileName`
- `pageUrl` (opcional)
- `error` (opcional)

Clases y funciones de biblioteca Node.js que se aplican sólo a los canaries de la API

Las siguientes funciones de biblioteca de CloudWatch Synthetics para Node.js solo son útiles para los API canaries de la UI.

Temas

- executeHttpStep(stepName, requestOptions, [callback], [stepConfig])

executeHttpStep(stepName, requestOptions, [callback], [stepConfig])

Ejecuta la solicitud HTTP proporcionada como un paso y publica `SuccessPercent` (aprobar o no aprobar) y las métricas `Duration`.

executeHttpStep utiliza funciones nativas HTTP o HTTPS que no son visibles a simple vista de acuerdo al protocolo que se ha especificado en la solicitud.

Esta función también agrega un resumen de ejecución de pasos al informe del valor controlado. El resumen incluye detalles sobre cada solicitud HTTP, como los siguientes:

- Hora de inicio
- Hora de finalización
- Estado (APROBADO o NO APROBADO)
- Razón del error, si hubo
- Detalles de llamada HTTP como cabeceras de solicitud o respuesta, cuerpo, código de estado, mensaje de estado y tiempos de rendimiento.

Parámetros

stepName(***`String`***)

Especifica el nombre del paso. Este nombre también se utiliza para publicar métricas de CloudWatch para este paso.

requestOptions(***`Object or String`***)

El valor de este parámetro puede ser una URL, una cadena URL o un objeto. Si es un objeto, entonces debe ser un conjunto de opciones configurables para realizar una solicitud HTTP. Es compatible con todas las opciones en http.request(options[, callback]) en el documento de Node.js.

Además de estas opciones de Node.js, requestOptions admite el parámetro adicional `body`. Puede utilizar el parámetro `body` para pasar datos como un cuerpo de la solicitud.

callback(***response***)

(Opcional) Esta es una función de usuario que se invoca con la respuesta HTTP. La respuesta es del tipo Clase: http.IncomingMessage.

stepConfig(***object***)

(Opcional) Utilice este parámetro para anular configuraciones globales de Synthetics con una configuración diferente para este paso.

Ejemplos de uso de executeHttpStep

La siguiente serie de ejemplos se crean entre sí para ilustrar los diversos usos de esta opción.

Este primer ejemplo configura los parámetros de solicitud. Puede pasar una URL como requestOptions:

```
let requestOptions = 'https://www.amazon.com';
```

O puede pasar un conjunto de opciones:

```
let requestOptions = {
        'hostname': 'myproductsEndpoint.com',
        'method': 'GET',
        'path': '/test/product/validProductName',
        'port': 443,
        'protocol': 'https:'
    };
```

El siguiente ejemplo crea una función de devolución de llamada que acepta una respuesta. De forma predeterminada, si no se especifica callback (devolución de llamada), CloudWatch Synthetics valida que el estado esté entre 200 y 299 inclusive.

```
// Handle validation for positive scenario
    const callback = async function(res) {
        return new Promise((resolve, reject) => {
            if (res.statusCode < 200 || res.statusCode > 299) {
                throw res.statusCode + ' ' + res.statusMessage;
            }
```

```
            let responseBody = '';
            res.on('data', (d) => {
                responseBody += d;
            });

            res.on('end', () => {
                // Add validation on 'responseBody' here if required. For ex, your
 status code is 200 but data might be empty
                resolve();
            });
        });
    };
```

El siguiente ejemplo crea una configuración para este paso que reemplaza la configuración global de CloudWatch Synthetics. La configuración de pasos de este ejemplo permite las cabeceras de solicitud, las cabeceras de respuesta, el cuerpo de la solicitud (datos posteriores) y el cuerpo de la respuesta en el informe y restringe los valores de las cabeceras de 'X-Amz-Security-Token' y de 'Autorización'. De forma predeterminada, estos valores no se incluyen en el informe por motivos de seguridad. Si elige incluirlos, los datos solo se almacenan en su bucket de S3.

```
// By default headers, post data, and response body are not included in the report for
 security reasons.
// Change the configuration at global level or add as step configuration for individual
 steps
let stepConfig = {
    includeRequestHeaders: true,
    includeResponseHeaders: true,
    restrictedHeaders: ['X-Amz-Security-Token', 'Authorization'], // Restricted header
 values do not appear in report generated.
    includeRequestBody: true,
    includeResponseBody: true
};
```

Este último ejemplo pasa su solicitud a executeHttpStep y nombra el paso.

```
await synthetics.executeHttpStep('Verify GET products API', requestOptions, callback,
 stepConfig);
```

Con este conjunto de ejemplos, CloudWatch Synthetics agrega los detalles de cada paso al informe y genera métricas para cada paso mediante stepName.

Se podrán ver `successPercent` y métricas `duration` para el paso `Verify GET products API`. Puede supervisar el rendimiento de la API si supervisa las métricas de los pasos de llamadas a la API.

Para obtener un script completo de ejemplo que utilice estas funciones, consulte Valor controlado de la API de varios pasos.

Funciones de la biblioteca disponibles para los scripts de valores controlados de Python que usan Selenium

Esta sección enumera las funciones de biblioteca Selenium disponibles para los scripts valores controlados de Python.

Temas

- Clases y funciones de biblioteca de Python y Selenium que se aplican a todos los canaries
- Clases y funciones de biblioteca de Python y Selenium que se aplican solo a canaries de la UI

Clases y funciones de biblioteca de Python y Selenium que se aplican a todos los canaries

Las siguientes funciones de biblioteca de CloudWatch Synthetics para Python son útiles para todos los canaries.

Temas

- Clase SyntheticsConfiguration
- Clase SyntheticsLogger

Clase SyntheticsConfiguration

Se puede utilizar la clase SyntheticsConfiguration para configurar el comportamiento de las funciones de biblioteca de Synthetics. Por ejemplo, se puede utilizar esta clase para configurar la función `executeStep()` para no hacer capturas de pantalla.

Se pueden establecer las configuraciones de CloudWatch Synthetics a nivel global.

Definiciones de la función

set_config(options)

```
from aws_synthetics.common import synthetics_configuration
```

options es un objeto, que es un conjunto de opciones configurables para el valor controlado. En las siguientes secciones se explican los posibles campos en *options*.

- `screenshot_on_step_start` (booleano): si se debe tomar una captura de pantalla antes de comenzar un paso.
- `screenshot_on_step_success` (booleano): si se debe tomar una captura de pantalla después de completar un paso correctamente.
- `screenshot_on_step_failure` (booleano): si se debe tomar una captura de pantalla después de que un paso falla.

with_screenshot_on_step_start(screenshot_on_step_start)

Acepta un argumento booleano, que indica si se debe tomar una captura de pantalla antes de iniciar un paso.

with_screenshot_on_step_success(screenshot_on_step_success)

Acepta un argumento booleano, que indica si se debe tomar una captura de pantalla después de completar un paso correctamente.

with_screenshot_on_step_failure(screenshot_on_step_failure)

Acepta un argumento booleano, que indica si se debe tomar una captura de pantalla después de que un paso falla.

get_screenshot_on_step_start()

Muestra si se debe tomar una captura de pantalla antes de iniciar un paso.

get_screenshot_on_step_success()

Muestra si se debe realizar una captura de pantalla después de completar un paso correctamente.

get_screenshot_on_step_failure()

Muestra si se debe tomar una captura de pantalla después de que un paso falla.

disable_step_screenshots()

Desactiva todas las opciones de captura de pantalla (get_screenshot_on_step_start, get_screenshot_on_step_success, y get_screenshot_on_step_failure).

enable_step_screenshots()

Habilita todas las opciones de captura de pantalla (get_screenshot_on_step_start, get_screenshot_on_step_success y get_screenshot_on_step_failure). Estos métodos no están habilitados de forma predeterminada.

setConfig(options) con respecto a las métricas de CloudWatch

Para los valores controlados que utilizan `syn-python-selenium-1.1` o posteriores, las (opciones) para setConfig pueden incluir los siguientes parámetros booleanos que determinan qué métricas publica el valor controlado. El valor predeterminado para cada una de estas opciones es `true`. Las opciones que comienzan con `aggregated` determinan si la métrica se emite sin la dimensión `CanaryName`. Se pueden utilizar estas métricas para ver los resultados agregados de todos los canaries. Las otras opciones determinan si la métrica se emite con la dimensión `CanaryName`. Se pueden usar estas métricas para ver los resultados de cada valor controlado individualmente.

Para obtener una lista de las métricas de CloudWatch que los canaries emiten, consulte Métricas de CloudWatch que los canaries publican.

- `failed_canary_metric` (booleano): si se debe emitir la métrica `Failed` (con la dimensión `CanaryName`) para este valor controlado. El valor predeterminado es `true`.
- `failed_requests_metric` (booleano): si se debe emitir la métrica `Failed requests` (con la dimensión `CanaryName`) para este valor controlado. El valor predeterminado es `true`.
- `2xx_metric` (booleano): si se debe emitir la métrica `2xx` (con la dimensión `CanaryName`) para este valor controlado. El valor predeterminado es `true`.
- `4xx_metric` (booleano): si se debe emitir la métrica `4xx` (con la dimensión `CanaryName`) para este valor controlado. El valor predeterminado es `true`.
- `5xx_metric` (booleano): si se debe emitir la métrica `5xx` (con la dimensión `CanaryName`) para este valor controlado. El valor predeterminado es `true`.
- `step_duration_metric` (booleano): si se debe emitir la métrica `Step duration` (con las dimensiones `CanaryName` y `StepName`) para este valor controlado. El valor predeterminado es `true`.
- `step_success_metric` (booleano): si se debe emitir la métrica `Step success` (con las dimensiones `CanaryName` y `StepName`) para este valor controlado. El valor predeterminado es `true`.
- `aggregated_failed_canary_metric` (booleano): si se debe emitir el métrica `Failed` (sin la dimensión `CanaryName`) para este valor controlado. El valor predeterminado es `true`.

- `aggregated_failed_requests_metric` (booleano): si se debe emitir el métrica `Failed Requests` (sin la dimensión `CanaryName`) para este valor controlado. El valor predeterminado es `true`.
- `aggregated_2xx_metric` (booleano): si se debe emitir el métrica `2xx` (sin la dimensión `CanaryName`) para este valor controlado. El valor predeterminado es `true`.
- `aggregated_4xx_metric` (booleano): si se debe emitir el métrica `4xx` (sin la dimensión `CanaryName`) para este valor controlado. El valor predeterminado es `true`.
- `aggregated_5xx_metric` (booleano): si se debe emitir el métrica `5xx` (sin la dimensión `CanaryName`) para este valor controlado. El valor predeterminado es `true`.

with_2xx_metric(2xx_metric)

Acepta un argumento booleano, que especifica si se emitirá una métrica `2xx` con la dimensión `CanaryName` para este valor controlado.

with_4xx_metric(4xx_metric)

Acepta un argumento booleano, que especifica si se emitirá una métrica `4xx` con la dimensión `CanaryName` para este valor controlado.

with_5xx_metric(5xx_metric)

Acepta un argumento booleano, que especifica si se emitirá una métrica `5xx`con la dimensión `CanaryName` para este valor controlado.

withAggregated2xxMetric(aggregated2xxMetric)

Acepta un argumento booleano, que especifica si se emitirá una métrica `2xx`sin dimensión para este valor controlado.

withAggregated4xxMetric(aggregated4xxMetric)

Acepta un argumento booleano, que especifica si se emitirá una métrica `4xx` sin dimensión para este valor controlado.

with_aggregated_5xx_metric(aggregated_5xx_metric)

Acepta un argumento booleano, que especifica si se emitirá una métrica `5xx`sin dimensión para este valor controlado.

with_aggregated_failed_canary_metric(aggregated_failed_canary_metric)

Acepta un argumento booleano, que especifica si se emitirá una métrica `Failed`sin dimensión para este valor controlado.

with_aggregated_failed_requests_metric(aggregated_failed_requests_metric)

Acepta un argumento booleano, que especifica si se emitirá una métrica `Failed requests`sin dimensión para este valor controlado.

with_failed_canary_metric(failed_canary_metric)

Acepta un argumento booleano, que especifica si se emitirá una métrica `Failed` con la dimensión `CanaryName` para este valor controlado.

with_failed_requests_metric(failed_requests_metric)

Acepta un argumento booleano, que especifica si se emitirá una métrica `Failed requests` con la dimensión `CanaryName` para este valor controlado.

with_step_duration_metric(step_duration_metric)

Acepta un argumento booleano, que especifica si se emitirá una métrica `Duration` con la dimensión `CanaryName` para este valor controlado.

with_step_success_metric(step_success_metric)

Acepta un argumento booleano, que especifica si se emitirá una métrica `StepSuccess` con la dimensión `CanaryName` para este valor controlado.

Métodos para habilitar o desactivar métricas

disable_aggregated_request_metrics()

Desactiva que el valor controlado emita todas las métricas de solicitud que se emiten sin dimensión `CanaryName`.

disable_request_metrics()

Deshabilita todas las métricas de solicitud, incluidas las métricas por valor controlado y las métricas agregadas en todos los valores controlados.

disable_step_metrics()

Desactiva todas las métricas de pasos, incluidas las métricas de éxito y de duración de los pasos.

enable_aggregated_request_metrics()

Permite que el valor controlado emita todas las métricas de solicitud que se emiten sin dimensión `CanaryName`.

enable_request_metrics()

Habilita todas las métricas de solicitud, incluidas las métricas por valor controlado y las métricas agregadas en todos los valores controlados.

enable_step_metrics()

Habilita todas las métricas de pasos, incluidas las métricas de éxito y de duración de los pasos.

Uso en valores controlados de la IU

Primero, importe la relación de Synthetics y obtenga la configuración. A continuación, establezca la configuración para cada opción mediante llamadas al método SetConfig con una de las siguientes opciones.

```
from aws_synthetics.common import synthetics_configuration

synthetics_configuration.set_config(
     {
        "screenshot_on_step_start": False,
        "screenshot_on_step_success": False,
        "screenshot_on_step_failure": True
     }
)

or
```

Or (Disyunción)

```
synthetics_configuration.with_screenshot_on_step_start(False).with_screenshot_on_step_success(F
```

Para desactivar todas las capturas de pantalla, utilice la función disableStepScreenshots() como en este ejemplo.

```
synthetics_configuration.disable_step_screenshots()
```

Puede habilitar y desactivar las capturas de pantalla en cualquier punto del código. Por ejemplo, para desactivar las capturas de pantalla solo para un paso, se deben desactivar antes de ejecutar ese paso y habilitarlas después del paso.

set_config(options) para los valores controlados de la interfaz de usuario

A partir de `syn-python-selenium-1.1`, para los valores controlados de la interfaz de usuario, `set_config` puede incluir los siguientes parámetros booleanos:

- `continue_on_step_failure` (booleano): si se debe continuar con la ejecución del script valor controlado después de que un paso falle (esto se refiere a la función executeStep). Si algún paso falla, la ejecución del valor controlado seguirá marcándose como fallida. El valor predeterminado es `false`.

Clase SyntheticsLogger

`synthetics_logger` ingresa los registros tanto en la consola como en un archivo de registros local, en el mismo nivel de registro. Este archivo de registro se escribe en ambas ubicaciones solo si el nivel de registro coincide con el deseado para la función de registro a la que se llamó o está por debajo de este.

Los valores “DEBUG: ”, “INFO: ”, etc. se anteponen a las instrucciones de registro del archivo de registro local para que coincidan con el nivel de registro de la función a la que se llamó.

No es necesario utilizar `synthetics_logger` para crear un archivo de registros que se carga en la ubicación de resultados de Amazon S3. En su lugar, puede crear un archivo de registro distinto en la carpeta `/tmp`. Los archivos creados en la carpeta `/tmp` se cargan en la ubicación de resultados del bucket de S3 como artefactos.

Para utilizar `synthetics_logger`.

```
from aws_synthetics.common import synthetics_logger
```

Definiciones de funciones útiles:

Obtenga el nivel de registro:

```
log_level = synthetics_logger.get_level()
```

Establezca el nivel de registro:

```
synthetics_logger.set_level()
```

Registre un mensaje con un nivel especificado. El nivel puede ser `DEBUG`, `INFO`, `WARN` o `ERROR`, como en los siguientes ejemplos de sintaxis:

```
synthetics_logger.debug(message, *args, **kwargs)
```

```
synthetics_logger.info(message, *args, **kwargs)
```

```
synthetics_logger.log(message, *args, **kwargs)
```

```
synthetics_logger.warn(message, *args, **kwargs)
```

```
synthetics_logger.error(message, *args, **kwargs)
```

Para obtener información acerca de los parámetros de depuración, consulte los documentos estándar de Python en logging.debug

En estas funciones de registro, el `message` es la cadena del formato del mensaje. Los `args` son los argumentos que se fusionan en `msg` que usan el operador de formato de cadena.

Hay tres argumentos de palabras clave en `kwargs`:

- `exc_info`: si no se evalúa como false, agrega información de excepción al mensaje de registro.
- `stack_info`: el valor predeterminado es false. Si es VERDADERO, agrega información de pila al mensaje de registro, incluida la llamada de registro real.
- `extra`: el tercer argumento opcional de palabra clave, que se puede utilizar para especificar un diccionario que se utiliza para rellenar el `__dict__` del `LogRecord` que se ha creado para el evento de registro con atributos definidos por el usuario.

Ejemplos:

Registre un mensaje con el nivel `DEBUG`:

```
synthetics_logger.debug('Starting step - login.')
```

Registre un mensaje con el nivel `INFO`. `logger.log` es sinónimo de `logger.info`:

```
synthetics_logger.info('Successfully completed step - login.')
```

o

```
synthetics_logger.log('Successfully completed step - login.')
```

Registre un mensaje con el nivel `WARN`:

```
synthetics_logger.warn('Warning encountered trying to publish %s', 'CloudWatch Metric')
```

Registre un mensaje con el nivel `ERROR`:

```
synthetics_logger.error('Error encountered trying to publish %s', 'CloudWatch Metric')
```

Registre una excepción:

```
synthetics_logger.exception(message, *args, **kwargs)
```

Registre un mensaje con nivel `ERROR`. La información de excepción se agrega al mensaje de registro. Debe llamar a esta función solo desde un controlador de excepciones.

Para obtener información acerca de los parámetros de excepción, consulte los documentos estándar de Python en logging.exception

El `message` es la cadena en formato de mensaje. Los `args` son los argumentos, que se fusionan en `msg` mediante el operador de formato de cadena.

Hay tres argumentos de palabras clave en `kwargs`:

- `exc_info`: si no se evalúa como false, agrega información de excepción al mensaje de registro.
- `stack_info`: el valor predeterminado es false. Si es VERDADERO, agrega información de pila al mensaje de registro, incluida la llamada de registro real.
- `extra`: el tercer argumento opcional de palabra clave, que se puede utilizar para especificar un diccionario que se utiliza para rellenar el `__dict__` del `LogRecord` que se ha creado para el evento de registro con atributos que el usuario ha definido.

Ejemplo:

```
synthetics_logger.exception('Error encountered trying to publish %s', 'CloudWatch
 Metric')
```

Clases y funciones de biblioteca de Python y Selenium que se aplican solo a canaries de la UI

Las siguientes funciones de la biblioteca de Selenium de CloudWatch Synthetics para Python solo son útiles para los valores controlados de la UI.

Temas

- Clase SyntheticsBrowser
- Clase SyntheticsWebDriver

Clase SyntheticsBrowser

Cuando se crea una instancia de navegador mediante una llamada a `synthetics_webdriver.Chrome()`, la instancia del navegador devuelta es del tipo `SyntheticsBrowser`. La clase `SyntheticsBrowser` controla el ChromeDriver y habilita el script valor controlado para manejar el navegador, lo que permite al WebDriver Selenium trabajar con Synthetics.

Además de los métodos estándar de Selenium, también proporciona los siguientes métodos.

set_viewport_size(ancho, alto)

Establece la ventana gráfica del navegador. Ejemplo:

```
browser.set_viewport_size(1920, 1080)
```

save_screenshot(nombre del archivo, sufijo)

Guarda capturas de pantalla en el directorio de `/tmp`. Las capturas de pantalla se cargan desde allí a la carpeta de artefactos de valores controlados en el bucket de S3.

nombre del archivo es el nombre del archivo para la captura de pantalla, y sufijo es una cadena opcional que se utilizará para nombrar la captura de pantalla.

Ejemplo:

```
browser.save_screenshot('loaded.png', 'page1')
```

Clase SyntheticsWebDriver

Para utilizar esta clase, utilice lo siguiente en su script:

```
from aws_synthetics.selenium import synthetics_webdriver
```

add_execution_error(errorMessage, ex);

`errorMessage` describe el error y `ex` es la excepción que se encuentra

`add_execution_error` puede usarse para establecer errores de ejecución para el valor controlado. Se produce un error en el valor controlado sin interrumpir la ejecución del script. Tampoco afecta a las métricas de `successPercent`.

Debe realizar un seguimiento de los errores como errores de ejecución sólo si no son importantes para indicar el éxito o el error del script valor controlado.

A continuación, se muestra un ejemplo del uso de un `add_execution_error`. Está supervisando la disponibilidad de su punto de conexión y tomando capturas de pantalla después de que la página se haya cargado. Debido a que el hecho de no tomar una captura de pantalla no determina la disponibilidad del punto de enlace, puede detectar cualquier error que encuentre al tomar capturas de pantalla y agregarlos como errores de ejecución. Las métricas de disponibilidad seguirán indicando que el punto de conexión está activo y en ejecución, pero el estado del valor controlado se marcará como fallido. El siguiente bloque de código de muestra detecta dicho error y lo agrega como un error de ejecución.

```
try:
    browser.save_screenshot("loaded.png")
except Exception as ex:
   self.add_execution_error("Unable to take screenshot", ex)
```

add_user_agent(user_agent_str)

Añade el valor de `user_agent_str` a la cabecera del agente de usuario del navegador. Debe asignar `user_agent_str` antes de crear la instancia del navegador.

Ejemplo:

```
synthetics_webdriver.add_user_agent('MyApp-1.0')
```

execute_step(step_name, function_to_execute)

Procesa una función. También hace lo siguiente:

- Registra que el paso se ha iniciado.
- Toma una captura de pantalla denominada `<stepName>-starting`.
- Inicia un temporizador.
- Ejecuta la función proporcionada.
- Si la función devuelve resultados normalmente, cuenta como superada. Si la función falla, cuenta como error.
- Finaliza el temporizador.
- Registra si el paso se ha superado o no.
- Toma una captura de pantalla denominada `<stepName>-succeeded` o `<stepName>-failed`.
- Emite la métrica `stepName SuccessPercent`, 100 para superado o 0 para no superado.
- Emite la métrica `stepName Duration`, con un valor basado en las horas de inicio y de finalización del paso.
- Por último, devuelve el mismo resultado que `functionToExecute` o vuelve a arrojar el mismo error que `functionToExecute`.

Ejemplo:

```
from selenium.webdriver.common.by import By

def custom_actions():
        #verify contains
        browser.find_element(By.XPATH, "//*[@id=\"id_1\"][contains(text(),'login')]")
        #click a button
        browser.find_element(By.XPATH, '//*[@id="submit"]/a').click()

    await synthetics_webdriver.execute_step("verify_click", custom_actions)
```

Chrome()

Lanza una instancia del navegador Chromium y muestra la instancia creada del navegador.

Ejemplo:

```
browser = synthetics_webdriver.Chrome()
```

```
browser.get("https://example.com/)
```

Para lanzar un navegador en modo de incógnito, utilice lo siguiente:

```
add_argument('—-incognito')
```

Para agregar la configuración del proxy, utilice lo siguiente:

```
add_argument('--proxy-server=%s' % PROXY)
```

Ejemplo:

```
from selenium.webdriver.chrome.options import Options
chrome_options = Options()
chrome_options.add_argument("—-incognito")
browser = syn_webdriver.Chrome(chrome_options=chrome_options)
```

Programación de las ejecuciones de valores controlados con cron

El uso de una expresión cron le da flexibilidad cuando programa un valor controlado. Las expresiones Cron contienen cinco o seis campos en el orden que se indica en la siguiente tabla. Los campos están separados por espacios. La sintaxis difiere en función de si está utilizando la consola de CloudWatch para crear el valor controlado o AWS CLI o los SDK de AWS. Cuando utilice la consola, especifique sólo los cinco primeros campos. Cuando utiliza AWS CLI o los SDK de AWS, especifique los seis campos y debe especificar * para el campo del `Year` (año).

Campo	Valores permitidos	Caracteres especiales permitidos
Minutos	0-59	, - * /
Horas	0-23	, - * /
Día del mes	1-31	, - * ? / L W
Mes	1-12 o JAN-DEC	, - * /
Día de la semana	1-7 o SUN-SAT	, - * ? L #
Año	*	

Caracteres especiales

- La , (coma) incluye varios valores en la expresión de un campo. Por ejemplo, en el campo Month (mes), ENERO, FEBRERO, MARZO incluirían enero, febrero y marzo.
- El - (guion) es un carácter especial que especifica intervalos. En el campo Day, 1-15 incluiría los días del 1 al 15 del mes especificado.
- El * (asterisco) es un carácter especial que incluye todos los valores del campo. En el campo (Hours) horas, * incluye cada hora. No puede utilizar * en los campos Day-of-month (Día del mes) y Day-of-week (Día de la semana) en la misma expresión. Si lo utiliza en uno, debe utilizar ? en el otro.
- La / (barra inclinada) especifica incrementos. En el campo Minutes (minutos), puede escribir 1/10 para especificar cada diez minutos, si empieza desde el primer minuto de la hora (por ejemplo, los minutos once, veintiuno y treinta y uno, etc.).
- El ? (signo de interrogación) especifica uno u otro. Si ingresa el número 7 en el campo Day-of-month (Día del mes) y no es importante especificar qué día de la semana es el séptimo, puede escribir ? en el campo Day-of-week (Día del mes).
- El comodín L en los campos Día del mes o Día de la semana especifica el último día del mes o de la semana.
- El comodín **`W`** en el campo Día del mes especifica un día de la semana. En el campo Día del mes, **`3W`** especifica el día de la semana más cercano al tercer día del mes.
- El comodín # en el campo Día de la semana especifica una instancia concreta del día de la semana de un mes. Por ejemplo, 3#2 es el segundo martes del mes. El número 3 hace referencia al martes, ya que es el tercer día de la semana en el calendario anglosajón, mientras que 2 hace referencia al segundo día de ese tipo dentro de un mes.

Limitaciones

- No se pueden especificar los campos Día del mes y Día de la semana en la misma expresión Cron. Si especifica un valor o un `*` (asterisco) en uno de los campos, debe utilizar un ? (signo de interrogación) en el otro.
- No se admiten expresiones cron que produzcan frecuencias superiores a un minuto.
- No se puede configurar un valor controlado para que espere más de un año antes de la ejecución, por lo que solo se puede especificar `*` en el `Year`.

Ejemplos

Puede consultar las siguientes cadenas cron de muestra al crear un valor controlado. Los siguientes ejemplos son la sintaxis correcta para utilizar AWS CLI o SDK de AWS para crear o actualizar un valor controlado. Si está utilizando la consola de CloudWatch, omita el * final en cada ejemplo.

Expression	Significado
`0 10 * * ? *`	Ejecutar a las 10:00 h (UTC) todos los días
`15 12 * * ? *`	Ejecutar a las 12:15 h (UTC) todos los días
`0 18 ? * MON-FRI *`	Ejecutar a las 18:00 h (UTC) de lunes a viernes
`0 8 1 * ? *`	Ejecútelo a las 08.00 h (UTC) el primer día de cada mes
`0/10 * ? * MON-SAT *`	Ejecútelo cada 10 minutos de lunes a sábado de cada semana
`0/5 8-17 ? * MON-FRI *`	Ejecútelo cada 5 minutos de lunes a viernes entre las 8.00 h y las 17.55 h (UTC)

Grupos

Puede crear grupos para asociar valores controlados entre sí, incluidos los valores controlados interregionales. El uso de grupos puede ayudarlo a administrar y automatizar los valores controlados y también puede ver los resultados de ejecución y las estadísticas agregadas de todos los valores controlados de un grupo.

Los grupos son recursos globales. Cuando se crea un grupo, se replica en todas las regiones de AWS que admiten grupos y puede agregarle valores controlados de cualquiera de estas regiones y verlos en cualquiera de ellas. Aunque el formato de ARN del grupo refleja el nombre de la región donde se creó, no se limita a ninguna región. Esto significa que puede colocar valores controlados de varias regiones en el mismo grupo y, a continuación, utilizar ese grupo para ver y administrar todos esos valores controlados en una sola vista.

Los grupos se admiten en todas las regiones, excepto en las que están deshabilitadas de forma predeterminada. Para obtener más información sobre estas regiones, consulte Activar una región.

Cada grupo puede contener hasta 10 valores controlados. Puede tener hasta 20 grupos en la cuenta. Cualquier valor controlado puede ser miembro de hasta 10 grupos.

Creación de un grupo

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Señales de aplicación, Valores controlados de Synthetics.
3. Elija Crear grupo.
4. En Group Name (nombre del grupo), escriba un nombre para el grupo.
5. Seleccione valores controlados para asociarlos a este grupo. Para seleccionar un valor controlado, escriba el nombre completo en Nombre exacto del valor controlado y elija Búsqueda. Seleccione la casilla de verificación que hay junto al nombre del valor controlado. Si hay varios valores controlados con el mismo nombre en diferentes regiones, asegúrese de seleccionar los que desee.

   Puede repetir este paso para asociar hasta 10 valores controlados al grupo.
6. (Opcional) En Tags (Etiquetas), agregue uno o más pares de clave-valor como etiquetas para este grupo. Las etiquetas pueden ayudarle a identificar y organizar sus recursos de AWS y a realizar un seguimiento de sus costos de AWS. Para obtener más información, consulte Etiquetado de los recursos de Amazon CloudWatch.
7. Elija Crear grupo.

Prueba local de un canario

En esta sección se explica cómo modificar, probar y depurar los canarios de CloudWatch Synthetics directamente en el editor de código Microsoft Visual Studio o en el editor de código JetBrains IDE. El entorno de depuración local utiliza un contenedor de Serverless Application Model (SAM) para simular una función de Lambda y emular el comportamiento de un canario Synthetics.

> **Note**
>
> No es práctico realizar una depuración local de canarios que dependan de la supervisión visual. La supervisión visual se basa en hacer capturas de pantalla de base durante una ejecución inicial y, a continuación, compararlas con las capturas de pantalla de las ejecuciones posteriores. En un entorno de desarrollo local, las ejecuciones no se almacenan ni se les realiza un seguimiento, y cada iteración es una ejecución independiente y autónoma. La ausencia de un historial de ejecuciones de canarios hace que no sea práctico depurar los canarios que se basan en la supervisión visual.

Requisitos previos

1. Elija o cree un bucket de Amazon S3 para almacenar artefactos de las ejecuciones de prueba de canarios locales, como archivos HAR y capturas de pantalla. Esto requiere que esté aprovisionado con IAM. Si se salta la configuración de los buckets de Amazon S3, podrá probar su canario de forma local, pero verá un mensaje de error sobre el bucket que falta y no tendrá acceso a los artefactos del canario.

   Si utiliza un bucket de Amazon S3, se recomienda que configure el ciclo de vida del bucket para eliminar objetos al cabo de unos días, a fin de ahorrar costos. Para obtener más información, consulte Administración del ciclo de vida del almacenamiento.
2. Configure un perfil de AWS predeterminado para su cuenta de AWS. Para obtener más información, consulte Opciones de los archivos de configuración y credenciales.
3. Defina la región de AWS predeterminada del entorno de depuración en la región que prefiera, por ejemplo `us-west-2`.
4. Instale la CLI de AWS SAM. Para obtener más información, consulte Instalación de la CLI de AWS SAM.
5. Instale Visual Studio Code Editor o JetBrains IDE. Para obtener más información, consulte Visual Studio Code o JetBrains IDE.
6. Instale Docker para trabajar con la CLI de AWS SAM. Asegúrese de iniciar el docker daemon. Para obtener más información, consulte Instalación de Docker para usarlo con la CLI de AWS SAM.

   Como alternativa, puede instalar otro software de administración de contenedores, por ejemplo Rancher, siempre que utilice el tiempo de ejecución de Docker.
7. Instale una extensión de un conjunto de herramientas de AWS para su editor de preferencia. Para obtener más información, consulte Instalación de AWS Toolkit for Visual Studio Code o Instalación de AWS Toolkit for JetBrains.

Temas

- Configuración del entorno de pruebas y depuración
- Utilizar Visual Studio Code IDE
- Utilizar JetBrains IDE
- Ejecución de un canario de forma local con la CLI de SAM
- Cómo integrar su entorno de pruebas local en un paquete de canarios existente

- Cómo cambiar el tiempo de ejecución de CloudWatch Synthetics
- Errores comunes

Configuración del entorno de pruebas y depuración

Primero, clone el repositorio de Github que proporciona AWS ejecutando el siguiente comando. El repositorio contiene ejemplos de código para canarios de Node.js y de Python.

```
git clone https://github.com/aws-samples/synthetics-canary-local-debugging-sample.git
```

Realice una de las siguientes acciones, en función del idioma de los canarios.

Para los canarios de Node.js

1. Acceda al directorio de origen del canario de Node.js ejecutando el siguiente comando.

   ```
   cd synthetics-canary-local-debugging-sample/nodejs-canary/src
   ```

2. Ingrese el siguiente comando para instalar las dependencias de canario.

   ```
   npm install
   ```

Para canarios de Python

1. Acceda al directorio de origen del canario de Python ejecutando el siguiente comando.

   ```
   cd synthetics-canary-local-debugging-sample/python-canary/src
   ```

2. Ingrese el siguiente comando para instalar las dependencias de canario.

   ```
   pip3 install -r requirements.txt -t .
   ```

Utilizar Visual Studio Code IDE

El archivo de configuración de inicialización Visual Studio se encuentra en `.vscode/launch.json`. Contiene configuraciones que permiten que el código Visual Studio detecte el archivo de plantilla. Define una carga de Lambda con los parámetros necesarios para invocar el canario correctamente. Esta es la configuración de inicialización de un archivo canario de Node.js:

```
{
            ...
            ...
            "lambda": {
                "payload": {
                    "json": {
                        // Canary name. Provide any name you like.
                        "canaryName": "LocalSyntheticsCanary",
                        // Canary artifact location
                        "artifactS3Location": {
                            "s3Bucket": "cw-syn-results-123456789012-us-west-2",
                            "s3Key": "local-run-artifacts",
                        },
                        // Your canary handler name
                        "customerCanaryHandlerName": "heartbeat-canary.handler"
                    }
                },
                // Environment variables to pass to the canary code
                "environmentVariables": {}
            }
        }
    ]
}
```

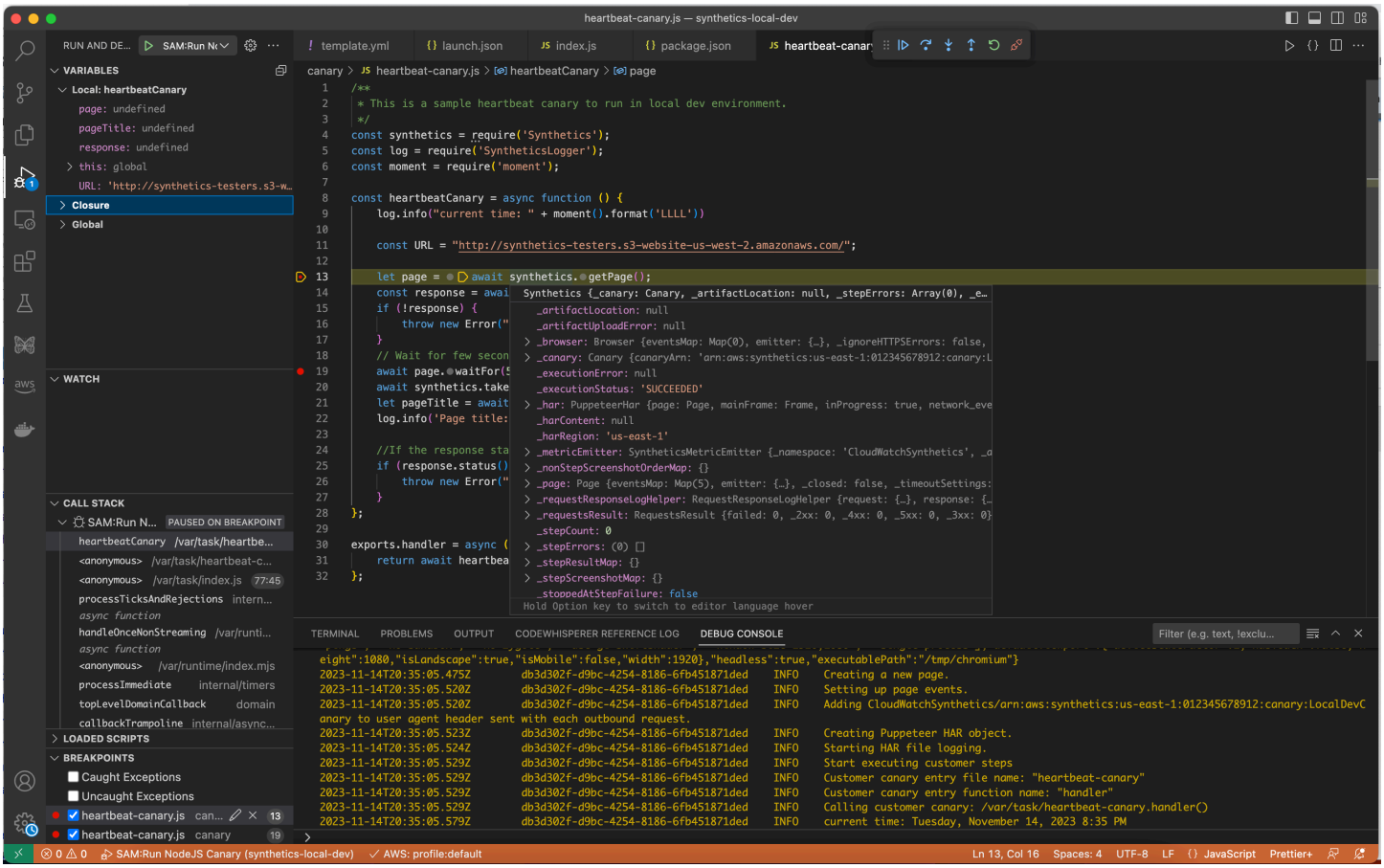
Si lo desea, también puede proporcionar los siguientes campos en el JSON de la carga:

- Valores válidos de `s3EncryptionMode`: `SSE_S3` | `SSE_KMS`
- Valor válido de `s3KmsKeyArn`: *`ARN de clave KMS`*
- Valores válidos de `activeTracing`: `true` | `false`
- Valor válido de `canaryRunId`: *`UUID`* Este parámetro es obligatorio si el seguimiento activo está habilitado.

Para depurar los canarios de Visual Studio, agregue puntos de interrupción en el código de canario donde desee pausar la ejecución. Para agregar un punto de interrupción, seleccione el margen del editor y vaya al modo Ejecutar y depurar en el editor. Ejecute el canario haciendo clic en el botón de reproducción. Cuando se ejecute el canario, los registros se guardarán en la consola de depuración, lo que le proporcionará información en tiempo real sobre el comportamiento del canario. Si agregó puntos de interrupción, la ejecución de canario se detendrá en cada punto de interrupción, lo que le

permitirá revisar el código e inspeccionar los valores de las variables, los métodos de instancia, los atributos de los objetos y la pila de llamadas a funciones.

No se incurre en ningún costo por ejecutar y depurar canarios de forma local, excepto por los artefactos almacenados en el bucket de Amazon S3 y las métricas de CloudWatch generadas por cada ejecución local.



Utilizar JetBrains IDE

Una vez instalada la extensión AWS Toolkit for JetBrains, asegúrese de que el complemento Node.js y el depurador de JavaScript estén habilitados para ejecutarse si va a depurar un canario de Node.js. A continuación, siga estos pasos:

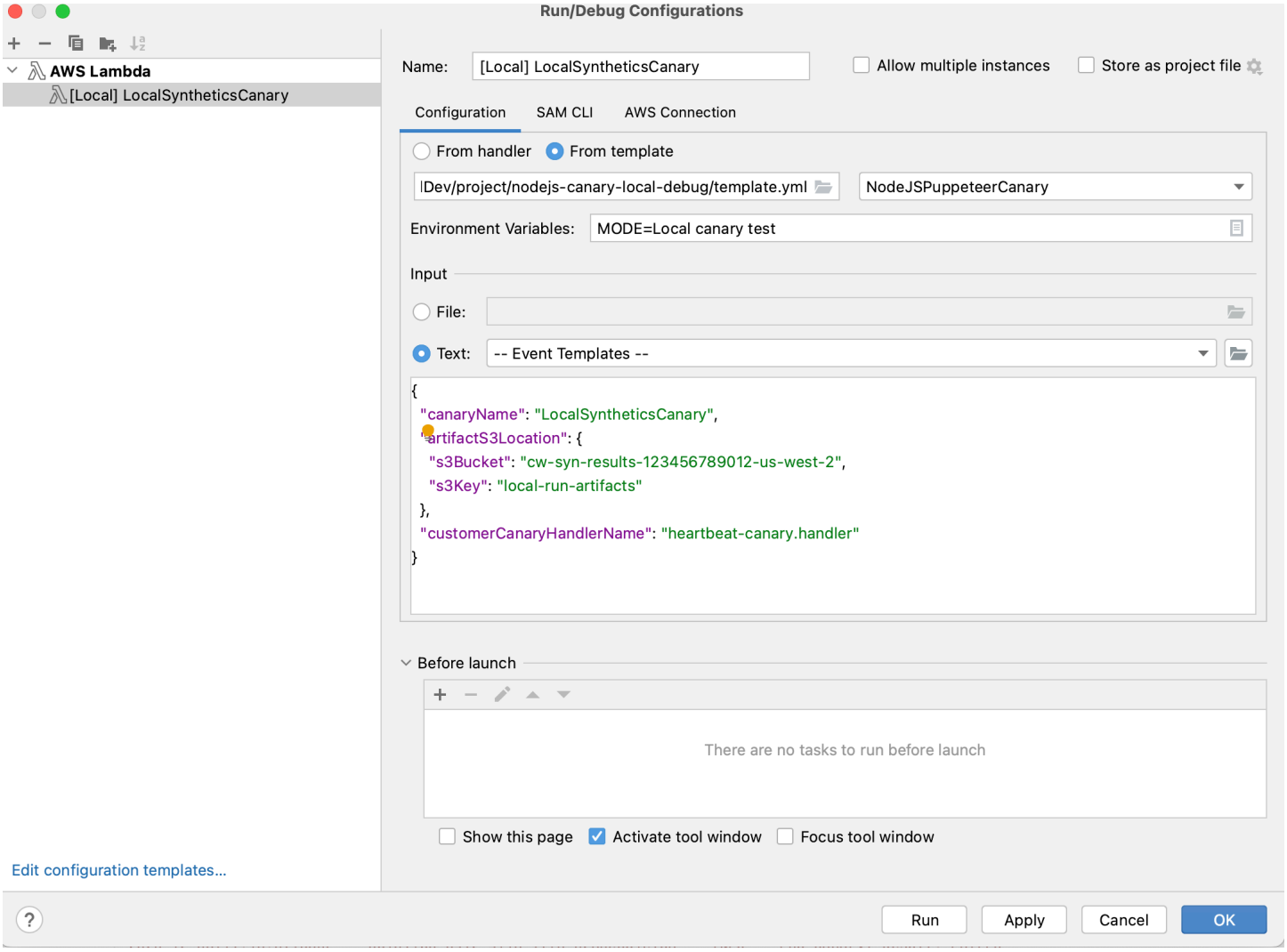
Depuración de un canario con JetBrains IDE

1. En el panel de navegación izquierdo de JetBrains IDE, elija Lambda y, a continuación, elija la plantilla de configuración local.
2. Ingreso de un nombre de configuración de ejecución, por ejemplo **`LocalSyntheticsCanary`**

3. Elija Desde plantilla, seleccione el explorador de archivos en el campo de plantilla y, a continuación, elija el archivo template.yml del proyecto, ya sea del directorio nodejs o del directorio python.
4. En la sección Entrada, introduzca la carga del canario, como se muestra en la siguiente pantalla.

```
{
 "canaryName": "LocalSyntheticsCanary",
 "artifactS3Location": {
     "s3Bucket": "cw-syn-results-123456789012-us-west-2",
     "s3Key": "local-run-artifacts"
 },
 "customerCanaryHandlerName": "heartbeat-canary.handler"
}
```

También puede definir otras variables de entorno en el JSON de carga, como se muestra en Utilizar Visual Studio Code IDE.



Ejecución de un canario de forma local con la CLI de SAM

Utilice uno de los siguientes procedimientos para ejecutar el canario de forma local mediante la CLI de Serverless Application Model (SAM). Asegúrese de especificar el nombre de su propio bucket de Amazon S3 para `s3Bucket` en `event.json`

Cómo usar la CLI de SAM para ejecutar un canario de Node.js

1. Acceda al directorio de origen ejecutando el siguiente comando.

```
cd synthetics-canary-local-debugging-sample/nodejs-canary
```

2. Introduzca los comandos siguientes.

```
sam build
```

```
sam local invoke -e ../event.json
```

Cómo usar la CLI de SAM para ejecutar un canario de Python

1. Acceda al directorio de origen ejecutando el siguiente comando.

```
cd synthetics-canary-local-debugging-sample/python-canary
```

2. Introduzca los comandos siguientes.

```
sam build
sam local invoke -e ../event.json
```

Cómo integrar su entorno de pruebas local en un paquete de canarios existente

Puede integrar la depuración local de canarios en su paquete de canarios existente copiando tres archivos:

- Copie el archivo `template.yml` en la raíz de su paquete de canarios. Asegúrese de modificar la ruta para que `CodeUri` apunte al directorio donde se encuentra su código de canario.
- Si está trabajando con un canario de Node.js, copie el archivo `cw-synthetics.js` en su directorio de origen de canarios. Si está trabajando con un canario de Python, copie el `cw-synthetics.py` en su directorio de origen de canarios.
- Copie el archivo de configuración de inicialización `.vscode/launch.json` en la raíz del paquete. Asegúrese de colocarlo dentro del directorio `.vscode`; créelo si aún no existe.

Cómo cambiar el tiempo de ejecución de CloudWatch Synthetics

Como parte de la depuración, puede intentar ejecutar un canario con un tiempo de ejecución de CloudWatch Synthetics diferente, en lugar del último. Para ello, busque el tiempo de ejecución que desee utilizar en una de las siguientes tablas. Asegúrese de seleccionar el tiempo de ejecución para la región correcta. A continuación, pegue el ARN de ese tiempo de ejecución en el lugar correspondiente del archivo `template.yml` y, a continuación, ejecute el canario.

Tiempos de ejecución de Node.js

ARN para syn-nodejs-puppeteer-7.0

En la siguiente tabla se muestran los ARN que se deben utilizar para la versión `syn-nodejs-puppeteer-7.0` del tiempo de ejecución de CloudWatch Synthetics en cada región de AWS en la que está disponible.

Región	ARN
Este de EE. UU. (Norte de Virginia)	`arn:aws:lambda:us-east-1:378653112637:layer:Synthetics:44`
Este de EE. UU. (Ohio)	`arn:aws:lambda:us-east-2:772927465453:layer:Synthetics:46`
Oeste de EE. UU. (Norte de California)	`arn:aws:lambda:us-west-1:332033056316:layer:Synthetics:44`
Oeste de EE. UU. (Oregón)	`arn:aws:lambda:us-west-2:760325925879:layer:Synthetics:47`
África (Ciudad del Cabo)	`arn:aws:lambda:af-south-1:461844272066:layer:Synthetics:44`
Asia-Pacífico (Hong Kong)	`arn:aws:lambda:ap-east-1:129828061636:layer:Synthetics:45`
Asia-Pacífico (Hyderabad)	`arn:aws:lambda:ap-south-2:280298676434:layer:Synthetics:20`
Asia-Pacífico (Yakarta)	`arn:aws:lambda:ap-southeast-3:246953257743:layer:Synthetics:26`
Asia-Pacífico (Melbourne)	`arn:aws:lambda:ap-southeast-4:200724813040:layer:Synthetics:18`
Asia-Pacífico (Bombay)	`arn:aws:lambda:ap-south-1:724929286329:layer:Synthetics:44`
Asia-Pacífico (Osaka)	`arn:aws:lambda:ap-northeast-3:608016332111:layer:Synthetics:30`

Región	ARN
Asia-Pacífico (Seúl)	arn:aws:lambda:ap-northeast-2:989515803484:layer:Synthetics:46
Asia-Pacífico (Singapur)	arn:aws:lambda:ap-southeast-1:068035103298:layer:Synthetics:49
Asia-Pacífico (Sídney)	arn:aws:lambda:ap-southeast-2:584677157514:layer:Synthetics:44
Asia-Pacífico (Tokio)	arn:aws:lambda:ap-northeast-1:172291836251:layer:Synthetics:44
Canadá (centro)	arn:aws:lambda:ca-central-1:236629016841:layer:Synthetics:44
Oeste de Canadá (Calgary	arn:aws:lambda:ca-west-1:944448206667:layer:Synthetics:76
China (Pekín)	arn:aws-cn:lambda:cn-north-1:422629156088:layer:Synthetics:45
China (Ningxia);	arn:aws-cn:lambda:cn-northwest-1:474974519687:layer:Synthetics:46
Europa (Fráncfort)	arn:aws:lambda:eu-central-1:122305336817:layer:Synthetics:44
Europa (Irlanda)	arn:aws:lambda:eu-west-1:563204233543:layer:Synthetics:46
Europa (Londres)	arn:aws:lambda:eu-west-2:565831452869:layer:Synthetics:44
Europa (Milán)	arn:aws:lambda:eu-south-1:525618516618:layer:Synthetics:45
Europa (París)	arn:aws:lambda:eu-west-3:469466506258:layer:Synthetics:44

Región	ARN
Europa (España)	`arn:aws:lambda:eu-south-2:029793053121:layer:Synthetics:20`
Europa (Estocolmo)	`arn:aws:lambda:eu-north-1:162938142733:layer:Synthetics:44`
Europa (Zúrich)	`arn:aws:lambda:eu-central-2:224218992030:layer:Synthetics:19`
Israel (Tel Aviv)	`arn:aws:lambda:il-central-1:313249807427:layer:Synthetics:17`
Medio Oriente (Baréin)	`arn:aws:lambda:me-south-1:823195537320:layer:Synthetics:44`
Medio Oriente (EAU)	`arn:aws:lambda:me-central-1:239544149032:layer:Synthetics:19`
América del Sur (São Paulo)	`arn:aws:lambda:sa-east-1:783765544751:layer:Synthetics:45`
AWS GovCloud (Este de EE. UU.)	`arn:aws-us-gov:lambda:us-gov-east-1:946759330430:layer:Synthetics:41`
AWS GovCloud (Oeste de EE. UU.)	`arn:aws-us-gov:lambda:us-gov-west-1:946807836238:layer:Synthetics:42`

ARN para syn-nodejs-puppeteer-6.2

En la siguiente tabla se muestran los ARN que se deben utilizar para la versión `syn-nodejs-puppeteer-6.2` del tiempo de ejecución de CloudWatch Synthetics en cada región de AWS en la que está disponible.

Región	ARN
Este de EE. UU. (Norte de Virginia)	arn:aws:lambda:us-east-1:378653112637:layer:Synthetics:41
Este de EE. UU. (Ohio)	arn:aws:lambda:us-east-2:772927465453:layer:Synthetics:43
Oeste de EE. UU. (Norte de California)	arn:aws:lambda:us-west-1:332033056316:layer:Synthetics:41
Oeste de EE. UU. (Oregón)	arn:aws:lambda:us-west-2:760325925879:layer:Synthetics:44
África (Ciudad del Cabo)	arn:aws:lambda:af-south-1:461844272066:layer:Synthetics:41
Asia-Pacífico (Hong Kong)	arn:aws:lambda:ap-east-1:129828061636:layer:Synthetics:42
Asia-Pacífico (Hyderabad)	arn:aws:lambda:ap-south-2:280298676434:layer:Synthetics:17
Asia-Pacífico (Yakarta)	arn:aws:lambda:ap-southeast-3:246953257743:layer:Synthetics:23
Asia-Pacífico (Melbourne)	arn:aws:lambda:ap-southeast-4:200724813040:layer:Synthetics:15
Asia-Pacífico (Bombay)	arn:aws:lambda:ap-south-1:724929286329:layer:Synthetics:41
Asia-Pacífico (Osaka)	arn:aws:lambda:ap-northeast-3:608016332111:layer:Synthetics:27
Asia-Pacífico (Seúl)	arn:aws:lambda:ap-northeast-2:989515803484:layer:Synthetics:42

Región	ARN
Asia-Pacífico (Singapur)	arn:aws:lambda:ap-southeast-1:068035103298:layer:Synthetics:46
Asia-Pacífico (Sídney)	arn:aws:lambda:ap-southeast-2:584677157514:layer:Synthetics:41
Asia-Pacífico (Tokio)	arn:aws:lambda:ap-northeast-1:172291836251:layer:Synthetics:41
Canadá (centro)	arn:aws:lambda:ca-central-1:236629016841:layer:Synthetics:41
Oeste de Canadá (Calgary	arn:aws:lambda:ca-west-1:944448206667:layer:Synthetics:73
China (Pekín)	arn:aws-cn:lambda:cn-north-1:422629156088:layer:Synthetics:42
China (Ningxia);	arn:aws-cn:lambda:cn-northwest-1:474974519687:layer:Synthetics:43
Europa (Fráncfort)	arn:aws:lambda:eu-central-1:122305336817:layer:Synthetics:41
Europa (Irlanda)	arn:aws:lambda:eu-west-1:563204233543:layer:Synthetics:43
Europa (Londres)	arn:aws:lambda:eu-west-2:565831452869:layer:Synthetics:41
Europa (Milán)	arn:aws:lambda:eu-south-1:525618516618:layer:Synthetics:42
Europa (París)	arn:aws:lambda:eu-west-3:469466506258:layer:Synthetics:41
Europa (España)	arn:aws:lambda:eu-south-2:029793053121:layer:Synthetics:17

Región	ARN
Europa (Estocolmo)	`arn:aws:lambda:eu-north-1:162938142733:layer:Synthetics:41`
Europa (Zúrich)	`arn:aws:lambda:eu-central-2:224218992030:layer:Synthetics:16`
Israel (Tel Aviv)	`arn:aws:lambda:il-central-1:313249807427:layer:Synthetics:14`
Medio Oriente (Baréin)	`arn:aws:lambda:me-south-1:823195537320:layer:Synthetics:41`
Medio Oriente (EAU)	`arn:aws:lambda:me-central-1:239544149032:layer:Synthetics:16`
América del Sur (São Paulo)	`arn:aws:lambda:sa-east-1:783765544751:layer:Synthetics:42`
AWS GovCloud (Este de EE. UU.)	`arn:aws-us-gov:lambda:us-gov-east-1:946759330430:layer:Synthetics:39`
AWS GovCloud (Oeste de EE. UU.)	`arn:aws-us-gov:lambda:us-gov-west-1:946807836238:layer:Synthetics:39`

ARN para syn-nodejs-puppeteer-5.2

En la siguiente tabla se muestran los ARN que se deben utilizar para la versión `syn-nodejs-puppeteer-5.2` del tiempo de ejecución de CloudWatch Synthetics en cada región de AWS en la que está disponible.

Región	ARN
Este de EE. UU. (Norte de Virginia)	`arn:aws:lambda:us-east-1:378653112637:layer:Synthetics:42`

Región	ARN
Este de EE. UU. (Ohio)	`arn:aws:lambda:us-east-2:772927465453:layer:Synthetics:44`
Oeste de EE. UU. (Norte de California)	`arn:aws:lambda:us-west-1:332033056316:layer:Synthetics:42`
Oeste de EE. UU. (Oregón)	`arn:aws:lambda:us-west-2:760325925879:layer:Synthetics:45`
África (Ciudad del Cabo)	`arn:aws:lambda:af-south-1:461844272066:layer:Synthetics:42`
Asia-Pacífico (Hong Kong)	`arn:aws:lambda:ap-east-1:129828061636:layer:Synthetics:43`
Asia-Pacífico (Hyderabad)	`arn:aws:lambda:ap-south-2:280298676434:layer:Synthetics:18`
Asia-Pacífico (Yakarta)	`arn:aws:lambda:ap-southeast-3:246953257743:layer:Synthetics:24`
Asia-Pacífico (Melbourne)	`arn:aws:lambda:ap-southeast-4:200724813040:layer:Synthetics:16`
Asia-Pacífico (Bombay)	`arn:aws:lambda:ap-south-1:724929286329:layer:Synthetics:42`
Asia-Pacífico (Osaka)	`arn:aws:lambda:ap-northeast-3:608016332111:layer:Synthetics:28`
Asia-Pacífico (Seúl)	`arn:aws:lambda:ap-northeast-2:989515803484:layer:Synthetics:44`
Asia-Pacífico (Singapur)	`arn:aws:lambda:ap-southeast-1:068035103298:layer:Synthetics:47`
Asia-Pacífico (Sídney)	`arn:aws:lambda:ap-southeast-2:584677157514:layer:Synthetics:42`

Región	ARN
Asia-Pacífico (Tokio)	arn:aws:lambda:ap-northeast-1:172291836251:layer:Synthetics:42
Canadá (centro)	arn:aws:lambda:ca-central-1:236629016841:layer:Synthetics:42
Oeste de Canadá (Calgary	arn:aws:lambda:ca-west-1:944448206667:layer:Synthetics:74
China (Pekín)	arn:aws-cn:lambda:cn-north-1:422629156088:layer:Synthetics:43
China (Ningxia);	arn:aws-cn:lambda:cn-northwest-1:474974519687:layer:Synthetics:44
Europa (Fráncfort)	arn:aws:lambda:eu-central-1:122305336817:layer:Synthetics:42
Europa (Irlanda)	arn:aws:lambda:eu-west-1:563204233543:layer:Synthetics:44
Europa (Londres)	arn:aws:lambda:eu-west-2:565831452869:layer:Synthetics:42
Europa (Milán)	arn:aws:lambda:eu-south-1:525618516618:layer:Synthetics:43
Europa (París)	arn:aws:lambda:eu-west-3:469466506258:layer:Synthetics:42
Europa (España)	arn:aws:lambda:eu-south-2:029793053121:layer:Synthetics:18
Europa (Estocolmo)	arn:aws:lambda:eu-north-1:162938142733:layer:Synthetics:42
Europa (Zúrich)	arn:aws:lambda:eu-central-2:224218992030:layer:Synthetics:17

Región	ARN
Israel (Tel Aviv)	`arn:aws:lambda:il-central-1:313249807427:layer:Synthetics:15`
Medio Oriente (Baréin)	`arn:aws:lambda:me-south-1:823195537320:layer:Synthetics:42`
Medio Oriente (EAU)	`arn:aws:lambda:me-central-1:239544149032:layer:Synthetics:17`
América del Sur (São Paulo)	`arn:aws:lambda:sa-east-1:783765544751:layer:Synthetics:43`
AWS GovCloud (Este de EE. UU.)	`arn:aws-us-gov:lambda:us-gov-east-1:946759330430:layer:Synthetics:40`
AWS GovCloud (Oeste de EE. UU.)	`arn:aws-us-gov:lambda:us-gov-west-1:946807836238:layer:Synthetics:40`

Tiempos de ejecución de Python

ARN para syn-python-selenium-3.0

En la siguiente tabla se muestran los ARN que se deben utilizar para la versión `syn-python-selenium-3.0` del tiempo de ejecución de CloudWatch Synthetics en cada región de AWS en la que está disponible.

Región	ARN
Este de EE. UU. (Norte de Virginia)	`aarn:aws:lambda:us-east-1:378653112637:layer:Synthetics_Selenium:32`
Este de EE. UU. (Ohio)	`arn:aws:lambda:us-east-2:772927465453:layer:Synthetics_Selenium:34`
Oeste de EE. UU. (Norte de California)	`arn:aws:lambda:us-west-1:332033056316:layer:Synthetics_Selenium:32`

Región	ARN
Oeste de EE. UU. (Oregón)	arn:aws:lambda:us-west-2:760325925879:layer:Synthetics_Selenium:34
África (Ciudad del Cabo)	arn:aws:lambda:af-south-1:461844272066:layer:Synthetics_Selenium:32
Asia-Pacífico (Hong Kong)	arn:aws:lambda:ap-east-1:129828061636:layer:Synthetics_Selenium:32
Asia-Pacífico (Hyderabad)	arn:aws:lambda:ap-south-2:280298676434:layer:Synthetics_Selenium:20
Asia-Pacífico (Yakarta)	arn:aws:lambda:ap-southeast-3:246953257743:layer:Synthetics_Selenium:26
Asia-Pacífico (Melbourne)	arn:aws:lambda:ap-southeast-4:200724813040:layer:Synthetics_Selenium:18
Asia-Pacífico (Bombay)	arn:aws:lambda:ap-south-1:724929286329:layer:Synthetics_Selenium:32
Asia-Pacífico (Osaka)	arn:aws:lambda:ap-northeast-3:608016332111:layer:Synthetics_Selenium:30
Asia-Pacífico (Seúl)	arn:aws:lambda:ap-northeast-2:989515803484:layer:Synthetics_Selenium:34
Asia-Pacífico (Singapur)	arn:aws:lambda:ap-southeast-1:068035103298:layer:Synthetics_Selenium:37
Asia-Pacífico (Sídney)	arn:aws:lambda:ap-southeast-2:584677157514:layer:Synthetics_Selenium:32
Asia-Pacífico (Tokio)	arn:aws:lambda:ap-northeast-1:172291836251:layer:Synthetics_Selenium:32
Canadá (centro)	arn:aws:lambda:ca-central-1:236629016841:layer:Synthetics_Selenium:32

Región	ARN
Oeste de Canadá (Calgary	arn:aws:lambda:ca-west-1:944448206667:layer:Synthetics_Selenium:76
China (Pekín)	arn:aws-cn:lambda:cn-north-1:422629156088:layer:Synthetics_Selenium:32
China (Ningxia);	arn:aws-cn:lambda:cn-northwest-1:474974519687:layer:Synthetics_Selenium:32
Europa (Fráncfort)	arn:aws:lambda:eu-central-1:122305336817:layer:Synthetics_Selenium:32
Europa (Irlanda)	arn:aws:lambda:eu-west-1:563204233543:layer:Synthetics_Selenium:34
Europa (Londres)	arn:aws:lambda:eu-west-2:565831452869:layer:Synthetics_Selenium:32
Europa (Milán)	arn:aws:lambda:eu-south-1:525618516618:layer:Synthetics_Selenium:33
Europa (París)	arn:aws:lambda:eu-west-3:469466506258:layer:Synthetics_Selenium:32
Europa (España)	arn:aws:lambda:eu-south-2:029793053121:layer:Synthetics_Selenium:20
Europa (Estocolmo)	arn:aws:lambda:eu-north-1:162938142733:layer:Synthetics_Selenium:32
Europa (Zúrich)	arn:aws:lambda:eu-central-2:224218992030:layer:Synthetics_Selenium:19
Israel (Tel Aviv)	arn:aws:lambda:il-central-1:313249807427:layer:Synthetics_Selenium:17
Medio Oriente (Baréin)	arn:aws:lambda:me-south-1:823195537320:layer:Synthetics_Selenium:32

Región	ARN
Medio Oriente (EAU)	`arn:aws:lambda:me-central-1:239544149032:layer:Synthetics_Selenium:19`
América del Sur (São Paulo)	`arn:aws:lambda:sa-east-1:783765544751:layer:Synthetics_Selenium:33`
AWS GovCloud (Este de EE. UU.)	`arn:aws-us-gov:lambda:us-gov-east-1:946759330430:layer:Synthetics_Selenium:30`
AWS GovCloud (Oeste de EE. UU.)	`arn:aws-us-gov:lambda:us-gov-west-1:946807836238:layer:Synthetics_Selenium:31`

ARN para syn-python-selenium-2.1

En la siguiente tabla se muestran los ARN que se deben utilizar para la versión `syn-python-selenium-2.1` del tiempo de ejecución de CloudWatch Synthetics en cada región de AWS en la que está disponible.

Región	ARN
Este de EE. UU. (Norte de Virginia)	`arn:aws:lambda:us-east-1:378653112637:layer:Synthetics:29`
Este de EE. UU. (Ohio)	`arn:aws:lambda:us-east-2:772927465453:layer:Synthetics:31`
Oeste de EE. UU. (Norte de California)	`arn:aws:lambda:us-west-1:332033056316:layer:Synthetics:29`
Oeste de EE. UU. (Oregón)	`arn:aws:lambda:us-west-2:760325925879:layer:Synthetics:31`
África (Ciudad del Cabo)	`arn:aws:lambda:af-south-1:461844272066:layer:Synthetics:29`

Región	ARN
Asia-Pacífico (Hong Kong)	arn:aws:lambda:ap-east-1:129828061636:layer:Synthetics:29
Asia-Pacífico (Hyderabad)	arn:aws:lambda:ap-south-2:280298676434:layer:Synthetics:17
Asia-Pacífico (Yakarta)	arn:aws:lambda:ap-southeast-3:246953257743:layer:Synthetics:23
Asia-Pacífico (Melbourne)	arn:aws:lambda:ap-southeast-4:200724813040:layer:Synthetics:15
Asia-Pacífico (Bombay)	arn:aws:lambda:ap-south-1:724929286329:layer:Synthetics:29
Asia-Pacífico (Osaka)	arn:aws:lambda:ap-northeast-3:608016332111:layer:Synthetics:27
Asia-Pacífico (Seúl)	arn:aws:lambda:ap-northeast-2:989515803484:layer:Synthetics:30
Asia-Pacífico (Singapur)	arn:aws:lambda:ap-southeast-1:068035103298:layer:Synthetics:34
Asia-Pacífico (Sídney)	arn:aws:lambda:ap-southeast-2:584677157514:layer:Synthetics:29
Asia-Pacífico (Tokio)	arn:aws:lambda:ap-northeast-1:172291836251:layer:Synthetics:29
Canadá (centro)	arn:aws:lambda:ca-central-1:236629016841:layer:Synthetics:29
Oeste de Canadá (Calgary	arn:aws:lambda:ca-west-1:944448206667:layer:Synthetics:73
China (Pekín)	arn:aws-cn:lambda:cn-north-1:422629156088:layer:Synthetics:29

Región	ARN
China (Ningxia);	arn:aws-cn:lambda:cn-northwest-1:474974519687:layer:Synthetics:29
Europa (Fráncfort)	arn:aws:lambda:eu-central-1:122305336817:layer:Synthetics:29
Europa (Irlanda)	arn:aws:lambda:eu-west-1:563204233543:layer:Synthetics:31
Europa (Londres)	arn:aws:lambda:eu-west-2:565831452869:layer:Synthetics:29
Europa (Milán)	arn:aws:lambda:eu-south-1:525618516618:layer:Synthetics:30
Europa (París)	arn:aws:lambda:eu-west-3:469466506258:layer:Synthetics:29
Europa (España)	arn:aws:lambda:eu-south-2:029793053121:layer:Synthetics:17
Europa (Estocolmo)	arn:aws:lambda:eu-north-1:162938142733:layer:Synthetics:29
Europa (Zúrich)	arn:aws:lambda:eu-central-2:224218992030:layer:Synthetics:16
Israel (Tel Aviv)	arn:aws:lambda:il-central-1:313249807427:layer:Synthetics:14
Medio Oriente (Baréin)	arn:aws:lambda:me-south-1:823195537320:layer:Synthetics:29
Medio Oriente (EAU)	arn:aws:lambda:me-central-1:239544149032:layer:Synthetics:16
América del Sur (São Paulo)	arn:aws:lambda:sa-east-1:783765544751:layer:Synthetics:30

Región	ARN
AWS GovCloud (Este de EE. UU.)	`arn:aws-us-gov:lambda:us-gov-east-1:946759330430:layer:Synthetics:29`
AWS GovCloud (Oeste de EE. UU.)	`arn:aws-us-gov:lambda:us-gov-west-1:946807836238:layer:Synthetics:29`

Errores comunes

Error: ejecutar proyectos SAM de AWS de forma local requiere Docker. ¿Lo tiene instalado y se está ejecutando?

Asegúrese de iniciar Docker en su equipo.

Error en la invocación local de SAM: se produjo un error (ExpiredTokenException) al llamar a la operación GetLayerVersion: el token de seguridad incluido en la solicitud ha caducado

Asegúrese de que el perfil predeterminado de AWS esté configurado.

Errores más comunes

Para obtener más información sobre los errores comunes con el SAM, consulte Solución de problemas de la CLI de SAM de AWS.

Solución de problemas de un valor controlado

En caso de que el valor controlado falle, verifique lo siguiente para solucionar el problema.

Solución de problemas generales

- Utilice la página de detalles de valores controlados para encontrar más información. En la consola de CloudWatch, elija Valores controlados en el panel de navegación y, a continuación, elija el nombre del valor controlado para abrir la página de detalles del valor controlado. En la pestaña Availability (Disponibilidad), elija la métrica SuccessPercent para ver si el problema es constante o intermitente.

  Mientras todavía está en la pestaña Availability (Disponibilidad), elija un punto de datos fallido para verlas capturas de pantalla, registros y los informes de pasos (si están disponibles) para esa ejecución fallida.

Si hay un informe de pasos disponible debido a que los pasos forman parte de su script, verifique qué paso ha fallado y vea las capturas de pantalla asociadas para ver el problema que los clientes están viendo.

También puede comprobar los archivos HAR para ver si una o más solicitudes están fallando. Puede profundizar en el uso de registros para analizar a fondo las solicitudes y errores fallidos. Finalmente, puede comparar estos artefactos con los artefactos de un valor controlado exitoso para identificar el problema.

De forma predeterminada, CloudWatch Synthetics toma capturas de pantalla para cada paso en un valor controlado de la UI. Sin embargo, es posible que el script esté configurado para desactivar las capturas de pantalla. Durante la depuración, es posible que desee habilitar las capturas de pantalla de nuevo. Del mismo modo, para los canaries de la API, es posible que desee ver las cabeceras y cuerpos de solicitud y respuesta HTTP durante la depuración. Para obtener más información acerca de cómo incluir los datos en el informe, consulte executeHttpStep(stepName, requestOptions, [callback], [stepConfig]).

- Si ha tenido una implementación reciente en su aplicación, retroceda y ejecute la depuración más tarde.
- Conéctese al punto de enlace de manera manual para verificar si puede reproducir el mismo problema.

Temas

- Problema del intercambio de recursos de origen cruzado (CORS)
- Problemas de condiciones de carrera de los canarios
- Solución de problemas de un valor controlado en una VPC

El valor controlado falla tras la actualización del entorno de Lambda

Los valores controlados de CloudWatch Synthetics se implementan como funciones de Lambda en su cuenta. Estas funciones de Lambda están sujetas a actualizaciones del tiempo de ejecución de Lambda periódicas que incluyen actualizaciones de seguridad, correcciones de errores y otras mejoras. Lambda se esfuerza por proporcionar actualizaciones de tiempo de ejecución que sean compatibles con versiones anteriores a las funciones existentes. Sin embargo, al igual que ocurre con los parches de software, hay casos excepcionales en los que una actualización del tiempo de ejecución puede afectar negativamente a una función ya existente. Si cree que un valor controlado se ha visto afectado por una actualización del tiempo de ejecución de Lambda, puede utilizar el modo manual de administración del tiempo de ejecución de Lambda (en las regiones compatibles) para revertir temporalmente la versión del tiempo de ejecución de Lambda. Esto mantiene al valor controlado en funcionamiento y minimiza las interrupciones, lo que proporciona tiempo para corregir la incompatibilidad antes de volver a la última versión del tiempo de ejecución.

Si el valor controlado falla después de una actualización del tiempo de ejecución de Lambda, la mejor solución es actualizar a uno de los tiempos de ejecución más recientes de Synthetics. Para obtener más información sobre los tiempos de ejecución más recientes, consulte Versiones de tiempo de ejecución de Synthetics.

Como solución alternativa, en las regiones en las que estén disponibles los controles de administración del tiempo de ejecución de Lambda, puede revertir un valor controlado a un tiempo de ejecución administrado por Lambda anterior, mediante el modo manual con los controles de administración del tiempo de ejecución. Puede configurar el modo manual mediante la AWS CLI o la consola de Lambda con los pasos que se indican a continuación en las siguientes secciones.

> **Warning**
>
> Al cambiar la configuración del tiempo de ejecución al modo manual, la función de Lambda no recibirá actualizaciones de seguridad automáticas hasta que vuelva al modo automático. Durante este periodo, la función de Lambda podría estar expuesta a vulnerabilidades de seguridad.

Requisitos previos

- Instalar jq
- Instale la versión más reciente de AWS CLI. Para obtener más información, consulte las instrucciones sobre la instalación y actualización de la AWS CLI.

Paso 1: obtener el ARN de la función de Lambda

Ejecute el siguiente comando para recuperar el campo `EngineArn` de la respuesta. Este `EngineArn` es el ARN de la función de Lambda asociada al valor controlado. Utilice este ARN en los siguientes pasos.

```
aws synthetics get-canary --name my-canary | jq '.Canary.EngineArn'
```

Resultado de ejemplo para `EngingArn`:

```
"arn:aws:lambda:us-west-2:123456789012:function:cwsyn-my-canary-dc5015c2-db17-4cb5-
afb1-EXAMPLE991:8"
```

Paso 2: obtener el ARN de la última versión válida del tiempo de ejecución de Lambda

Para saber si su valor controlado se vio afectado por una actualización del tiempo de ejecución de Lambda, compruebe si la fecha y la hora de los cambios del ARN de la versión del tiempo de ejecución de Lambda en sus registros coinciden con la fecha y la hora en que vio los problemas en su valor controlado. Si no coinciden, probablemente no sea una actualización del tiempo de ejecución de Lambda lo que esté causando los problemas.

Si su valor controlado se ve afectado por una actualización del tiempo de ejecución de Lambda, debe identificar el ARN de la versión del tiempo de ejecución de Lambda en funcionamiento que utilizaba anteriormente. Siga las instrucciones de Identificación de los cambios de versión del tiempo de ejecución para buscar el ARN del tiempo de ejecución anterior. Registre el ARN de la versión del tiempo de ejecución y continúe con el paso 3 para establecer la configuración de administración del tiempo de ejecución.

Si su valor controlado aún no se ha visto afectado por una actualización del entorno de Lambda, puede buscar el ARN de la versión del tiempo de ejecución de Lambda que utiliza actualmente. Ejecute el siguiente comando para recuperar el `RuntimeVersionArn` de la función de Lambda de la respuesta.

```
aws lambda get-function-configuration \
--function-name "arn:aws:lambda:us-west-2:123456789012:function:cwsyn-my-canary-
dc5015c2-db17-4cb5-afb1-EXAMPLE991:8" | jq '.RuntimeVersionConfig.RuntimeVersionArn'
```

Resultado de ejemplo para `RuntimeVersionArn`:

```
"arn:aws:lambda:us-
west-2::runtime:EXAMPLE647b82f490a45d7ddd96b557b916a30128d9dcab5f4972911ec0f"
```

Paso 3: actualizar la configuración de administración del tiempo de ejecución de Lambda

Puede utilizar la AWS CLI o la consola de Lambda para actualizar la configuración de administración del tiempo de ejecución.

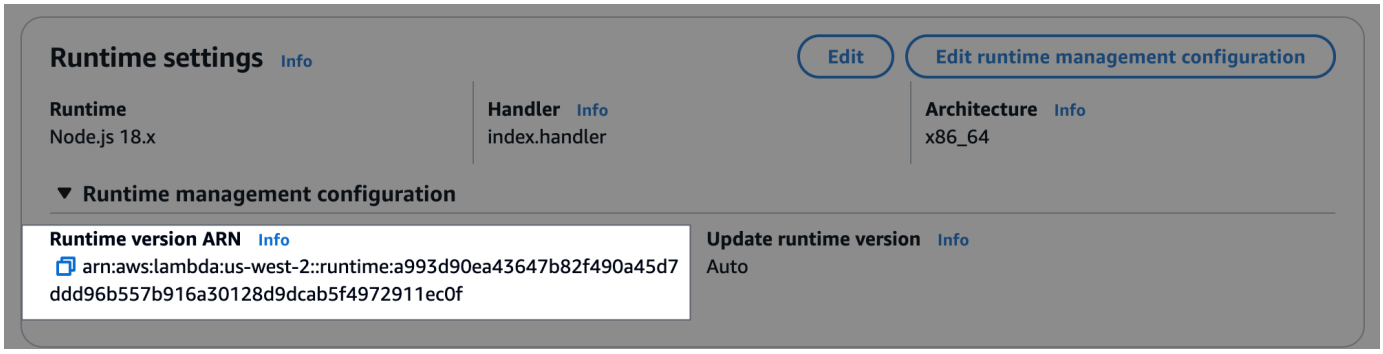
Para establecer el modo manual de la configuración de administración del tiempo de ejecución de Lambda mediante la AWS CLI

Ingrese el siguiente comando para cambiar la administración del tiempo de ejecución de la función de Lambda al modo manual. Asegúrese de reemplazar *`function-name`* y *`qualifier`* por el ARN de la función de Lambda y el número de versión de la función de Lambda, respectivamente, con los valores que encontró en el paso 1. Sustituya también *`runtime-version-arn`* por el ARN de la versión que encontró en el paso 2.

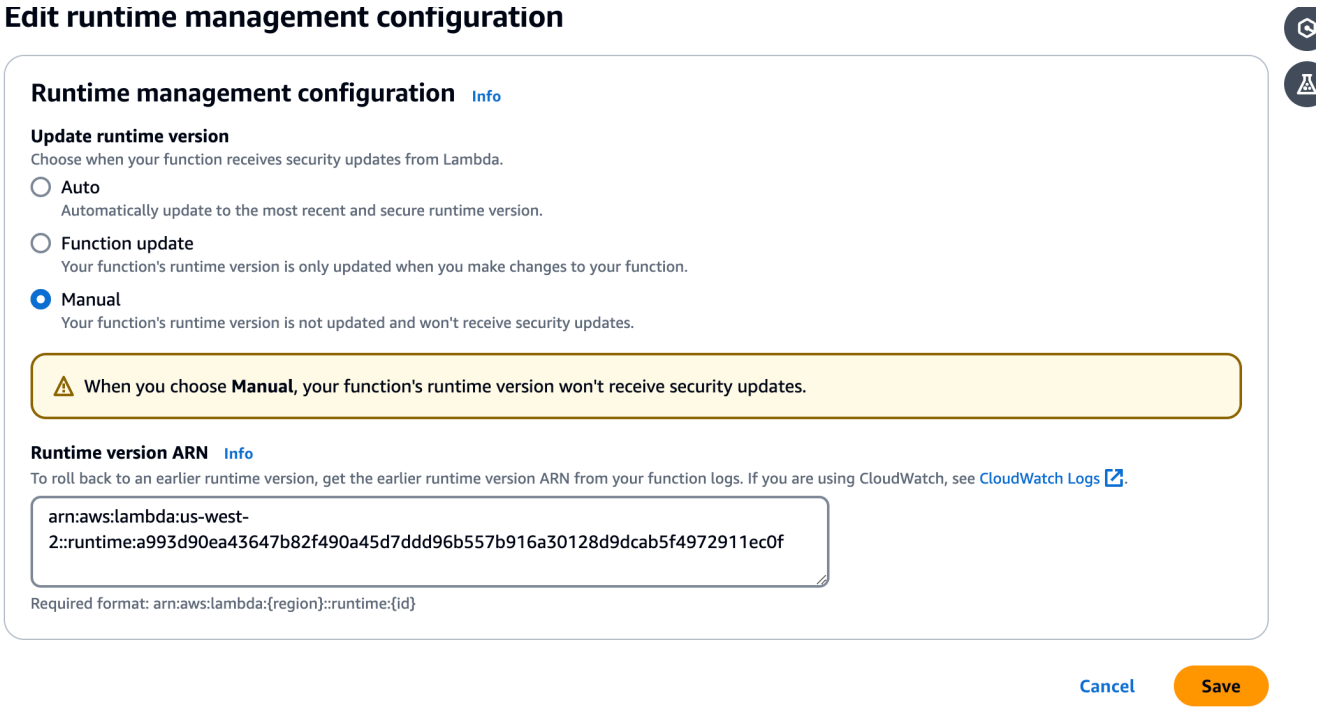
```
aws lambda put-runtime-management-config \
    --function-name "arn:aws:lambda:us-west-2:123456789012:function:cwsyn-my-canary-
dc5015c2-db17-4cb5-afb1-EXAMPLE991" \
    --qualifier 8 \
    --update-runtime-on "Manual" \
    --runtime-version-arn "arn:aws:lambda:us-
west-2::runtime:a993d90ea43647b82f490a45d7ddd96b557b916a30128d9dcab5f4972911ec0f"
```

Para cambiar un valor controlado al modo manual mediante la consola de Lambda

1. Abra la consola de AWS Lambda en https://console.aws.amazon.com/lambda/.
2. Elija la pestaña Versiones, elija el enlace del número de versión que corresponda a su ARN y elija la pestaña Código.
3. Desplácese hacia abajo hasta Configuración de tiempo de ejecución, expanda la Configuración de administración del tiempo de ejecución y copie el ARN de la versión del tiempo de ejecución.



4. Elija Editar la configuración de administración del tiempo de ejecución, elija Manual y pegue el ARN de la versión del tiempo de ejecución que copió anteriormente en el campo ARN de la versión del tiempo de ejecución. A continuación, elija Guardar.



Mi valor controlado está bloqueado por AWS WAF

Para evitar que AWS WAF bloquee su valor controlado, configure una condición de coincidencia de cadena de AWS WAF que permita el uso de la cadena `CloudWatchSynthetics`. Para obtener más información, consulte Trabajar con condiciones de coincidencia de cadena en la documentación de AWS WAF.

En la espera de un elemento

Después de analizar los registros y las capturas de pantalla, si nota que el script está esperando que aparezca un elemento en la pantalla y agota el tiempo de espera, verifique la captura de pantalla correspondiente para ver si el elemento aparece en la página. Verifique el `xpath` para asegurarse de que es correcto.

Para problemas relacionados con el Puppeteer, consulte Puppeteer's GitHub page (Página de GitHub de Puppeteer) o foros de Internet.

El nodo no es visible o no es un HTMLElement para page.click()

Si un nodo no es visible o no es un `HTMLElement` para `page.click()`, verifique primero el `xpath` que está utilizando para hacer clic en el elemento. Además, si el elemento se encuentra en la parte inferior de la pantalla, ajuste la ventana gráfica. CloudWatch Synthetics utiliza de forma predeterminada una ventana gráfica de 1920 x 1080. Puede establecer una ventana gráfica diferente al lanzar el navegador o mediante la función Puppeteer `page.setViewport`.

No se pueden cargar artefactos en S3; excepción: no se puede obtener la ubicación del bucket de S3: acceso denegado

Si el valor controlado no funciona debido a un error de Amazon S3, esto significa que CloudWatch Synthetics no ha podido cargar las capturas de pantalla, los registros o los informes creados para el valor controlado debido a problemas de permisos. Compruebe lo siguiente:

- Verifique que el rol de IAM del valor controlado tenga el permiso `s3:ListAllMyBuckets`, el permiso `s3:GetBucketLocation` para el bucket de Amazon S3 correcto y el permiso `s3:PutObject` para el bucket y donde el valor controlado almacena sus artefactos. Si el valor controlado lleva a cabo una supervisión visual, el rol también necesita el permiso `s3:GetObject` para el bucket. Estos mismos permisos también se requieren en la política del punto de conexión de la puerta de enlace de Amazon VPC S3, si el valor controlado se implementa en una VPC con un punto de conexión de VPC.
- Si el valor controlado utiliza una clave administrada por el cliente de AWS KMS para cifrado en lugar de la clave administrada de AWS estándar (predeterminada), es posible que el rol de IAM del valor controlado no tenga permiso para cifrar o descifrar con esa clave. Para obtener más información, consulte Cifrado de artefactos de un valor controlado.
- Es posible que la política de bucket no permita el mecanismo de cifrado que utiliza el valor controlado. Por ejemplo, si la política de bucket obliga a utilizar un mecanismo de cifrado

específico o una clave de KMS, debe seleccionar el mismo modo de cifrado para su valor controlado.

Si el valor controlado lleva a cabo una supervisión visual, consulte Actualización de la ubicación y el cifrado de artefactos al utilizar supervisión visual para obtener más información.

Error: error de protocolo (Runtime.CallFunctionOn): destino cerrado.

Este error aparece si hay algunas solicitudes de red después de cerrar la página o el navegador. Es posible que haya olvidado esperar una operación asíncrona. Después de ejecutar el script, CloudWatch Synthetics cierra el navegador. La ejecución de cualquier operación asíncrona después de cerrar el navegador puede causar `target closed error`.

Error de valor controlado. Error: no hay punto de datos, el valor controlado muestra error de tiempo de espera

Esto significa que la ejecución del valor controlado superó el tiempo de espera. La ejecución del valor controlado se detuvo antes de que CloudWatch Synthetics pudiera publicar métricas porcentuales exitosas de CloudWatch o actualizar artefactos como archivos HAR, registros y capturas de pantalla. Si su tiempo de espera es demasiado corto, puede aumentarlo.

De forma predeterminada, un valor de tiempo de espera del valor controlado es igual a su frecuencia. Puede ajustar manualmente el valor de tiempo de espera para que sea menor o igual que la frecuencia de los valores controlados. Si la frecuencia del valor controlado es baja, debe aumentar la frecuencia para aumentar el tiempo de espera. Puede ajustar tanto la frecuencia como el valor de tiempo de espera en Programa cuando cree o actualice un valor controlado mediante la consola de CloudWatch Synthetics.

Asegúrese de que el valor de tiempo de espera del valor controlado no sea inferior a 15 segundos para permitir arranques en frío de Lambda y el tiempo que tarda en arrancar la instrumentación de valor controlado.

Los artefactos de valores controlados no se pueden ver en la consola de CloudWatch Synthetics cuando se produce este error. Se puede utilizar Registros de CloudWatch para ver los registros de valores controlados.

Cómo utilizar Registros de CloudWatch para ver los registros de un valor controlado

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.

2. En el panel de navegación a la izquierda, elija Log groups (Grupos de registros).
3. Busque el grupo de registro con el nombre del valor controlado en el cuadro de filtro. Los grupos de registro para valores controlados tienen el nombre /aws/lambda/cwsyn-***canaryName***-randomId.

Acceso a un punto de enlace interno

Si desea que el valor controlado acceda a un punto de conexión de la red interna, se recomienda que configure CloudWatch Synthetics para que utilice VPC. Para obtener más información, consulte Ejecución de un valor controlado en una VPC.

Problemas con la actualización y las versiones anteriores de tiempo de ejecución de valores controlados

Si ha actualizado recientemente el valor controlado de la versión de tiempo de ejecución `syn-1.0` a una versión posterior, puede ocurrir un problema de intercambio de recursos de origen cruzado (CORS). Para obtener más información, consulte Problema del intercambio de recursos de origen cruzado (CORS).

Si recientemente ha instalado una versión anterior de tiempo de ejecución del valor controlado, verifique que las funciones de CloudWatch Synthetics que está utilizando están disponibles en la versión de tiempo de ejecución anterior a la que ha instalado. Por ejemplo, la función `executeHttpStep` está disponible para la versión de tiempo de ejecución `syn-nodejs-2.2` y posteriores. Para verificar la disponibilidad de las funciones, consulte Escritura de un script de valor controlado.

> **Note**
>
> Si planea actualizar o instalar versiones anteriores de tiempo de ejecución de un valor controlado, se recomienda que primero clone el valor controlado y actualice la versión del tiempo de ejecución en el valor controlado clonado. Una vez que haya verificado que el clon con la nueva versión de tiempo de ejecución funciona, puede actualizar la versión de tiempo de ejecución del valor controlado original y eliminar el clon.

Problema del intercambio de recursos de origen cruzado (CORS)

En un valor controlado de la UI, si algunas solicitudes de red fallan con `403` o `net::ERR_FAILED`, verifique si el valor controlado tiene habilitado el rastreo activo y también si utiliza la función Puppeteer `page.setExtraHTTPHeaders` para agregar cabeceras. Si es así, las solicitudes de red fallidas podrían deberse a restricciones de intercambio de recursos de origen cruzado (CORS). Puede confirmar si este es el caso al desactivar el rastreo activo o al eliminar las cabeceras HTTP adicionales.

¿Por qué sucede esto?

Cuando se utiliza el rastreo activo, se agrega una cabecera adicional a todas las solicitudes salientes para rastrear la llamada. La modificación de las cabeceras de solicitud al agregar una cabecera de seguimiento o cabeceras adicionales con Puppeteer's `page.setExtraHTTPHeaders` provoca una verificación de CORS para las solicitudes XMLHttpRequest (XHR).

Si no desea desactivar el rastreo activo o eliminar las cabeceras adicionales, se puede actualizar la aplicación web para permitir el acceso de origen cruzado o puede desactivar la seguridad web mediante el indicador `disable-web-security` cuando lance el navegador Chrome en el script.

Puede anular los parámetros de lanzamiento que CloudWatch Synthetics ha utilizado y pasar los parámetros adicionales del indicador `disable-web-security` mediante la función de lanzamiento de CloudWatch Synthetics. Para obtener más información, consulte Funciones de la biblioteca disponibles para los scripts de valor controlado de Node.js.

> **Note**
>
> Se pueden anular los parámetros de lanzamiento que CloudWatch Synthetics ha utilizado cuando utiliza la versión de tiempo de ejecución `syn-nodejs-2.1` o posteriores.

Problemas de condiciones de carrera de los canarios

Para disfrutar de la mejor experiencia al usar CloudWatch Synthetics, asegúrese de que el código escrito para los canarios sea idempotente. De lo contrario, en raras ocasiones, las ejecuciones de canarios pueden encontrarse con condiciones de carrera cuando el canario interactúa con el mismo recurso en distintas ejecuciones.

Solución de problemas de un valor controlado en una VPC

Si tiene problemas después de crear o actualizar un valor controlado en una VPC, puede encontrar la solución en una de las siguientes secciones.

Un valor controlado nuevo muestra estado de error o no se puede actualizar

Si crea un valor controlado para ejecutarlo en una VPC y este entra inmediatamente en estado de error, o bien no puede actualizar un valor controlado para ejecutarlo en una VPC, es posible que el rol del valor controlado no tenga los permisos correctos. Para ejecutarse en una VPC, un valor controlado debe tener los permisos `ec2:CreateNetworkInterface`, `ec2:DescribeNetworkInterfaces` y `ec2:DeleteNetworkInterface`. Todos estos permisos están contenidos en la política administrada `AWSLambdaVPCAccessExecutionRole`. Para obtener más información, consulte Rol de ejecución y permisos de usuario.

Si el problema se produjo al crear un valor controlado, debe eliminarlo y crear uno nuevo. Si utiliza la consola de CloudWatch para crear el nuevo valor controlado, en Permisos de acceso, seleccione Crear un nuevo rol. Se crea un nuevo rol que incluye todos los permisos necesarios para ejecutar el valor controlado.

Si el problema ocurre al actualizar un valor controlado, puede volver a actualizarlo y proporcionar un nuevo rol que tenga los permisos necesarios.

Error “No se devolvió ningún resultado de prueba”

Si un valor controlado muestra un error de tipo “no se devolvió ningún resultado de prueba”, la causa puede ser uno de los problemas siguientes:

- Si la VPC no tiene acceso a Internet, debe utilizar los puntos de conexión de VPC para proporcionar al valor controlado acceso a CloudWatch y a Amazon S3. Debe habilitar las opciones DNS Resolution (Resolución de DNS) y DNS hostname (Nombre de host DNS) en la VPC para que estas direcciones de punto de enlace se resuelvan correctamente. Para obtener más información, consulte Uso de DNS con su VPC y Uso de CloudWatch y CloudWatch Synthetics con puntos de conexión de VPC de interfaz.
- Los Canaries debe ejecutarse en subredes privadas dentro de una VPC. Para comprobarlo, abra la página Subnets (Subredes) en la consola de VPC. Compruebe las subredes que seleccionó al configurar el valor controlado. Si tienen una ruta a una gateway de Internet (igw-), no son subredes privadas.

Para ayudarle a solucionar estos problemas, consulte los registros del valor controlado.

Cómo ver los eventos de registro de un valor controlado

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, seleccione Grupos de registro.
3. Elija el nombre del grupo de registro del valor controlado. El nombre del grupo de registro comienza por `/aws/lambda/cwsyn-`*`canary-name`*.

Código de muestra para scripts de valores controlados

Esta sección contiene muestras de códigos que ilustran algunas funciones posibles para los scripts valores controlados de CloudWatch Synthetics.

Muestras para Node.js y Puppeteer

Configuración de cookies

Los sitios web se basan en cookies para proporcionar funcionalidad personalizada o realizar un seguimiento de usuarios. Al configurar las cookies en scripts de CloudWatch Synthetics, puede imitar este comportamiento personalizado y validarlo.

Por ejemplo, un sitio web puede mostrar un enlace para Login (Iniciar sesión) para un usuario que vuelva a visitar la página en lugar de un enlace para Register (Registrarse).

```
var synthetics = require('Synthetics');
const log = require('SyntheticsLogger');

const pageLoadBlueprint = async function () {

    let url = "http://smile.amazon.com/";

    let page = await synthetics.getPage();

    // Set cookies.  I found that name, value, and either url or domain are required
 fields.
    const cookies = [{
      'name': 'cookie1',
      'value': 'val1',
      'url': url
    },{
```

```
      'name': 'cookie2',
      'value': 'val2',
      'url': url
    },{
      'name': 'cookie3',
      'value': 'val3',
      'url': url
    }];

    await page.setCookie(...cookies);

    // Navigate to the url
    await synthetics.executeStep('pageLoaded_home', async function (timeoutInMillis =
 30000) {

        var response = await page.goto(url, {waitUntil: ['load', 'networkidle0'],
 timeout: timeoutInMillis});

        // Log cookies for this page and this url
        const cookiesSet = await page.cookies(url);
        log.info("Cookies for url: " + url + " are set to: " +
 JSON.stringify(cookiesSet));
    });

};

exports.handler = async () => {
    return await pageLoadBlueprint();
};
```

Emulación de dispositivos

Puede escribir scripts que emulan varios dispositivos para poder aproximar el aspecto y el comportamiento de una página en esos dispositivos.

El siguiente ejemplo emula un dispositivo iPhone 6. Para obtener más información acerca de la emulación, consulte page.emulate (options) en la documentación de Puppeteer.

```
var synthetics = require('Synthetics');
const log = require('SyntheticsLogger');
const puppeteer = require('puppeteer-core');

const pageLoadBlueprint = async function () {
```

```
    const iPhone = puppeteer.devices['iPhone 6'];

    // INSERT URL here
    const URL = "https://amazon.com";

    let page = await synthetics.getPage();
    await page.emulate(iPhone);

    //You can customize the wait condition here. For instance,
    //using 'networkidle2' may be less restrictive.
    const response = await page.goto(URL, {waitUntil: 'domcontentloaded', timeout:
 30000});
    if (!response) {
        throw "Failed to load page!";
    }

    await page.waitFor(15000);

    await synthetics.takeScreenshot('loaded', 'loaded');

    //If the response status code is not a 2xx success code
    if (response.status() < 200 || response.status() > 299) {
        throw "Failed to load page!";
    }
};

exports.handler = async () => {
    return await pageLoadBlueprint();
};
```

Valor controlado de la API de varios pasos

Este código de muestra demuestra un valor controlado de la API con dos pasos HTTP: prueba la misma API para casos de prueba positivos y negativos. La configuración del paso se pasa para habilitar la generación de informes de cabeceras de solicitud o de respuesta. Además, oculta el encabezado Autorización y X-Amz-Security-Token, ya que contienen credenciales de usuario.

Cuando este script se utiliza como un valor controlado, se pueden ver los detalles sobre cada paso y las solicitudes HTTP asociadas, como el paso aprobar/no aprobar, la duración y las métricas de rendimiento, como el tiempo de búsqueda del DNS y la hora del primer byte. Se puede ver el número de 2xx, 4xx y 5xx para la ejecución del valor controlado.

```
var synthetics = require('Synthetics');
const log = require('SyntheticsLogger');

const apiCanaryBlueprint = async function () {

    // Handle validation for positive scenario
    const validatePositiveCase = async function(res) {
        return new Promise((resolve, reject) => {
            if (res.statusCode < 200 || res.statusCode > 299) {
                throw res.statusCode + ' ' + res.statusMessage;
            }

            let responseBody = '';
            res.on('data', (d) => {
                responseBody += d;
            });

            res.on('end', () => {
                // Add validation on 'responseBody' here if required. For ex, your
 status code is 200 but data might be empty
                resolve();
            });
        });
    };

    // Handle validation for negative scenario
    const validateNegativeCase = async function(res) {
        return new Promise((resolve, reject) => {
            if (res.statusCode < 400) {
                throw res.statusCode + ' ' + res.statusMessage;
            }

            resolve();
        });
    };

    let requestOptionsStep1 = {
        'hostname': 'myproductsEndpoint.com',
        'method': 'GET',
        'path': '/test/product/validProductName',
        'port': 443,
        'protocol': 'https:'
```

```
    };

    let headers = {};
    headers['User-Agent'] = [synthetics.getCanaryUserAgentString(), headers['User-
Agent']].join(' ');

    requestOptionsStep1['headers'] = headers;

    // By default headers, post data and response body are not included in the report
 for security reasons.
    // Change the configuration at global level or add as step configuration for
 individual steps
    let stepConfig = {
        includeRequestHeaders: true,
        includeResponseHeaders: true,
        restrictedHeaders: ['X-Amz-Security-Token', 'Authorization'], // Restricted
 header values do not appear in report generated.
        includeRequestBody: true,
        includeResponseBody: true
    };


    await synthetics.executeHttpStep('Verify GET products API with valid name',
 requestOptionsStep1, validatePositiveCase, stepConfig);

    let requestOptionsStep2 = {
        'hostname': 'myproductsEndpoint.com',
        'method': 'GET',
        'path': '/test/canary/InvalidName(',
        'port': 443,
        'protocol': 'https:'
    };

    headers = {};
    headers['User-Agent'] = [synthetics.getCanaryUserAgentString(), headers['User-
Agent']].join(' ');

    requestOptionsStep2['headers'] = headers;

    // By default headers, post data and response body are not included in the report
 for security reasons.
    // Change the configuration at global level or add as step configuration for
 individual steps
    stepConfig = {
```

```
        includeRequestHeaders: true,
        includeResponseHeaders: true,
        restrictedHeaders: ['X-Amz-Security-Token', 'Authorization'], // Restricted
 header values do not appear in report generated.
        includeRequestBody: true,
        includeResponseBody: true
    };

    await synthetics.executeHttpStep('Verify GET products API with invalid name',
 requestOptionsStep2, validateNegativeCase, stepConfig);

};

exports.handler = async () => {
    return await apiCanaryBlueprint();
};
```

Muestras para Python y Selenium

El siguiente código de Selenium de muestra es un valor controlado que falla con un mensaje de error personalizado cuando no se carga un elemento de destino.

```
from aws_synthetics.selenium import synthetics_webdriver as webdriver
from aws_synthetics.common import synthetics_logger as logger
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
from selenium.webdriver.common.by import By

def custom_selenium_script():
    # create a browser instance
    browser = webdriver.Chrome()
    browser.get('https://www.example.com/')
    logger.info('navigated to home page')
    # set cookie
    browser.add_cookie({'name': 'foo', 'value': 'bar'})
    browser.get('https://www.example.com/')
    # save screenshot
    browser.save_screenshot('signed.png')
    # expected status of an element
    button_condition = EC.element_to_be_clickable((By.CSS_SELECTOR, '.submit-button'))
    # add custom error message on failure
    WebDriverWait(browser, 5).until(button_condition, message='Submit button failed to
 load').click()
```

```
    logger.info('Submit button loaded successfully')
    # browser will be quit automatically at the end of canary run,
    # quit action is not necessary in the canary script
    browser.quit()

# entry point for the canary
def handler(event, context):
    return custom_selenium_script()
```

Canaries y rastreo X-Ray

Puede elegir habilitar el seguimiento activo de AWS X-Ray en los canaries que utilizan `syn-nodejs-2.0` o tiempos de ejecución posteriores. Con el rastreo activado, se envían seguimientos para todas las llamadas que el valor controlado realice y que utilizan el navegador, el SDK de AWS o los módulos HTTP o HTTPS. Los valores controlados con el seguimiento activado aparecen en el Mapa de seguimiento de X-Ray y en Application Signals una vez que lo haya activado para su aplicación.

> **Note**
>
> La activación del rastreo de X-Ray en canaries aún no se admite en Asia Pacífico (Yakarta).

Cuando aparece un valor controlado en un mapa de seguimiento de X-Ray, aparece como un nuevo tipo de nodo cliente. Puede pasar el ratón sobre un nodo de valor controlado para ver datos sobre la latencia, las solicitudes y sobre los errores. También puede elegir el nodo de valor controlado para ver más datos en la parte inferior de la página. Desde esta área de la página, puede elegir Ver en Synthetics para ir a la consola de CloudWatch Synthetics para obtener más detalles sobre el valor controlado, o elija Consultar seguimientos para ver más detalles sobre los rastros de las ejecuciones del valor controlado.

Un valor controlado con seguimiento habilitado también tiene un Seguimiento en la página de detalles, con detalles sobre trazas y segmentos de las ejecuciones del valor controlado.

La activación del rastreo aumenta el tiempo de ejecución del valor controlado entre un 2,5 % y un 7 %.

Un valor controlado con rastreo habilitado debe utilizar un rol con los siguientes permisos. Si se utiliza la consola para crear el rol cuando se crea el valor controlado, se le otorgan estos permisos.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Sid230934",
            "Effect": "Allow",
            "Action": [
                "xray:PutTraceSegments"
            ],
            "Resource": "*"
        }
    ]
}
```

Las trazas generadas por los canaries incurren en cargos. Para obtener más información acerca de los precios de X-Ray, consulte Precios de AWS X-Ray.

Ejecución de un valor controlado en una VPC

Puede ejecutar canaries en los puntos de enlace de una VPC, así como en los puntos de enlace internos públicos. Para ejecutar un valor controlado en una VPC, debe tener habilitadas las opciones DNS Resolution (Resolución de DNS) y DNS hostnames (Nombres de host DNS) en la VPC. Para obtener más información, consulte Utilización de DNS con su VPC.

Cuando ejecute un valor controlado en un punto de conexión de VPC, debe proporcionar una manera en la que este envíe las métricas a CloudWatch y los artefactos a Amazon S3. Si la VPC ya está habilitada para el acceso a Internet, no hay nada más que hacer. El valor controlado se ejecuta en la VPC, pero puede acceder a Internet para cargar sus métricas y artefactos.

Si la VPC no está habilitada aún para el acceso a Internet, tiene dos opciones:

- Habilítela para obtener acceso a Internet. Para obtener más información, consulte la siguiente sección Otorgar acceso a Internet para su valor controlado en una VPC.
- Si desea mantener la VPC como privada, puede configurar el valor controlado para que envíe los datos a CloudWatch y Amazon S3 mediante los puntos de conexión de VPC privados. Si aún no lo ha hecho, debe crear un punto de conexión de VPC para CloudWatch (com.amazonaws.*region*.monitoring) y un punto de conexión de puerta de enlace para Amazon S3. Para obtener más información, consulte Uso de CloudWatch y CloudWatch Synthetics con los puntos de enlace de la VPC de tipo interfaz y Puntos de enlace de Amazon VPC para Amazon S3.

Otorgar acceso a Internet para su valor controlado en una VPC

Siga estos pasos para dar acceso a Internet a su valor controlado de VPC o para asignarle una dirección IP estática

Para otorgar acceso a Internet a un valor controlado en una VPC

1. Cree una puerta de enlace NAT en una subred pública en la VPC. Para obtener instrucciones, consulte Create a NAT gateway(Creación de una puerta de enlace NAT).
2. Agregue una nueva ruta a la tabla de enrutamiento de la subred privada donde se lanza el valor controlado. Especifique lo siguiente:

   - En Destination (Destino), ingrese **`0.0.0.0/0`**
   - En Objetivo, elija Puerta de enlace NAT y luego, el ID de la puerta de enlace NAT que ha creado.
   - Elija Guardar rutas.

   Para obtener más información acerca de la adición de rutas a la tabla de enrutamiento, consulte Add and remove routes from a route table (Agregar y eliminar rutas de la tabla de rutas).

> **Note**
>
> Asegúrese de que las rutas a la puerta de enlace NAT se encuentran en un estado activo. Si se elimina la puerta de enlace NAT y las rutas no están actualizadas, estas están en estado de agujero negro. Para obtener más información, consulte Work with NAT gateways (Trabajar con puertas de enlace NAT).

Cifrado de artefactos de un valor controlado

CloudWatch Synthetics almacena artefactos de un valor controlado como capturas de pantalla, archivos HAR e informes en su bucket de Amazon S3. De forma predeterminada, estos artefactos se cifran en reposo mediante una clave administrada por AWS. Para obtener más información sobre los tipos de claves, consulte Claves de cliente y claves de AWS.

Puede elegir usar una opción de cifrado diferente. CloudWatch Synthetics es compatible con lo siguiente:

- SSE-S3: cifrado de lado del servidor (SSE) con una clave administrada por Amazon S3.
- SSE-KMS: cifrado de lado de servidor (SSE) con una clave administrada por el cliente de AWS KMS.

Si desea utilizar la opción de cifrado predeterminada con una clave administrada por AWS, no necesita permisos adicionales.

Para utilizar el cifrado SSE-S3, debe especificar SE_S3 como modo de cifrado cuando crea o actualiza el valor controlado. No necesita permisos adicionales para utilizar este modo de cifrado. Para obtener más información, consulte Protecting data using server-side encryption with Amazon S3-managed encryption keys (SSE-S3) (Proteger los datos con el cifrado del lado del servidor con claves de cifrado administradas por Amazon S3 [SSE-S3]).

Para utilizar una clave administrada por el cliente de AWS KMS, debe especificar SSE-KMS como el modo de cifrado al crear o actualizar el valor controlado y también proporcionar el nombre de recurso de Amazon (ARN) de la clave. También puede utilizar una clave de KMS entre cuentas.

Para utilizar una clave administrada por el cliente, necesita la siguiente configuración:

- El rol de IAM para el valor controlado debe tener permiso para cifrar los artefactos a través de la clave. Si utiliza la supervisión visual, también debe otorgarle permiso para descifrar artefactos.

```
{
    "Version": "2012-10-17",
    "Statement": {"Effect": "Allow",
        "Action": [
            "kms:GenerateDataKey",
            "kms:Decrypt"
        ],
        "Resource": "Your KMS key ARN"
    }
}
```

- En lugar de agregarle permisos a su rol de IAM, puede agregar su rol de IAM a la política de clave. Si utiliza el mismo rol para varios canaries, debería considerar este enfoque.

```
{
    "Sid": "Enable IAM User Permissions",
    "Effect": "Allow",
    "Principal": {
```

```
            "AWS": "Your synthetics IAM role ARN"
        },
        "Action": [
            "kms:GenerateDataKey",
            "kms:Decrypt"
        ],
        "Resource": "*"
}
```

- Si utiliza una clave KMS entre cuentas, consulte Permitir a los usuarios de otras cuentas utilizar una clave KMS.

Visualización de artefactos de valores controlados cifrados al utilizar una clave administrada por el cliente

Para ver artefactos de valores controlados, actualice la clave administrada por el cliente para otorgarle a AWS KMS el permiso de descifrado para el usuario que ve los artefactos. Como alternativa, agregue permisos de descifrado al usuario o rol de IAM que esté viendo los artefactos.

La política de AWS KMS predeterminada permite que las políticas de IAM de la cuenta permitan el acceso a las claves KMS. Si utiliza una clave de KMS de entre varias cuentas, consulte ¿Por qué los usuarios de varias cuentas reciben errores de acceso denegado cuando intentan acceder a objetos de Amazon S3 cifrados por una clave de AWS KMS personalizada?.

Para obtener más información acerca de la solución de problemas de acceso denegado por una clave KMS, consulte Solución de problemas de acceso a las claves.

Actualización de la ubicación y el cifrado de artefactos al utilizar supervisión visual

Para llevar a cabo una supervisión visual, CloudWatch Synthetics compara las capturas de pantalla con las capturas de pantalla de referencia adquiridas en la ejecución seleccionada como punto de referencia. Si actualiza la ubicación de artefacto o la opción de cifrado, debe realizar una de estas acciones:

- Asegúrese de que el rol de IAM tenga permisos suficientes tanto para la ubicación anterior de Amazon S3 como para la nueva ubicación de Amazon S3 para artefactos. Asegúrese también de que tenga permiso para los métodos de cifrado previos y nuevos y las claves KMS.
- Cree una nueva base de referencia al seleccionar la siguiente ejecución de un valor controlado como nueva base de referencia. Si utiliza esta opción, solo tiene que asegurarse de que su rol de IAM tenga permisos suficientes para la nueva ubicación de artefactos y la opción de cifrado.

Recomendamos la segunda opción de seleccionar la siguiente ejecución como nueva base de referencia. Esto evita depender de una ubicación de artefacto o una opción de cifrado que ya no está usando para el valor controlado.

Por ejemplo, supongamos que el valor controlado utiliza la ubicación del artefacto A y la clave KMS K para cargar artefactos. Si actualiza el valor controlado a la ubicación del artefacto B y la clave KMS L, puede asegurarse de que su rol de IAM tenga permisos para las ubicaciones de artefactos (A y B) y ambas claves KMS (K y L). Como alternativa, puede seleccionar la siguiente ejecución como nueva base de referencia y asegurarse de que el rol de IAM del valor controlado tenga permisos para la ubicación del artefacto B y la clave KMS L.

Visualización de las estadísticas y los detalles de los valores controlados

Puede ver detalles sobre sus canaries y ver estadísticas sobre sus ejecuciones.

Para poder ver todos los detalles sobre los resultados de las ejecuciones de valor controlado, debe iniciar sesión en una cuenta con permisos suficientes. Para obtener más información, consulte Roles y permisos necesarios para los canaries de CloudWatch.

Para ver estadísticas y detalles de los valores controlados

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Señales de aplicación, Valores controlados de Synthetics.

   En los detalles sobre los canaries que ha creado:

   - Status (Estado) muestra visualmente cuántos de sus canaries han superado sus ejecuciones más recientes.
   - Groups (Grupos) muestra los grupos que ha creado y cuántos de ellos tienen valores controlados que fallan o son alarmantes.
   - Slowest performers (Rendimientos más lentos) muestra el grupo y la región con los valores controlados de rendimiento más lento. Se calculan al agregar la duración media de todos los valores controlados (a lo largo del periodo de tiempo seleccionado) dentro de un grupo o región y dividirla por el número de valores controlados en el grupo o región. Si elige la métrica para grupo más lento, la tabla se filtra para mostrar solo los grupos más lentos y los valores controlados. La tabla se ordena por duración media.
   - Cerca de la parte inferior de la página hay una tabla que muestra todos los canaries. Una columna muestra las alarmas que cada valor controlado ha creado. Solo se muestran las

alarmas que cumplen con el estándar de nomenclatura para las alarmas de los valores controlados. Este estándar es `Synthetics-Alarm-`*`canaryName`*`-`*`index`* . Las alarmas de los canarias que crea en la sección Synthetics de la consola de CloudWatch utiliza automáticamente esta convención de nomenclatura. Si crea alarmas para los valores controlados en la sección Alarms (Alarmas) de la consola de CloudWatch o mediante AWS CloudFormation y no utiliza esta convención de nomenclatura, las alarmas funcionan pero no aparecen en esta lista.

3. Para ver más detalles sobre un solo valor controlado, elija su nombre en la tabla valores controlados (Valores controlados).

   En los detalles de ese valor controlado:

   - La pestaña Availability (Disponibilidad) muestra la información sobre las últimas ejecuciones del valor controlado.

     En Ejecuciones de valores controlados, puede elegir una de las líneas para ver detalles sobre esa ejecución.

     En el gráfico, puede elegir Steps (Pasos), Screenshot (Captura de pantalla), Logs (Registros) o HAR file (Archivo HAR) para ver este tipo de detalles. Si el valor controlado tiene activado el rastreo activo, también puede elegir Seguimientos para ver la información de rastreo de las ejecuciones del valor controlado.

     Los registros para las ejecuciones del valor controlado se almacenan en los buckets de S3 y en Registros de CloudWatch.

     Las capturas de pantalla muestran cómo los clientes ven las páginas web. Puede utilizar los archivos HAR (archivos HTTP) para ver en detalle los datos de rendimiento sobre las páginas web. Puede analizar la lista de solicitudes web y detectar problemas de rendimiento, como el tiempo de carga de un elemento. Los archivos de registros muestran el registro de interacciones entre la ejecución del valor controlado y la página web y se pueden utilizar para identificar detalles de errores.

     Si el valor controlado utiliza el tiempo de ejecución `syn-nodejs-2.0-beta` o uno posterior, puede ordenar los archivos HAR por código de estado, tamaño de solicitud o por duración.

     La pestaña Steps (Pasos) muestra una lista de los pasos del valor controlado, el estado de cada paso, el motivo del error, la URL después de la ejecución del paso, capturas de pantalla y la duración de la ejecución del paso. Para canaries de la API con pasos HTTP, puede ver los

pasos y las solicitudes HTTP correspondientes si está utilizando el tiempo de ejecución `syn-nodejs-2.2` o uno posterior.

Elija la pestaña HTTP Requests (Solicitudes HTTP) para ver el registro de cada solicitud HTTP que el valor controlado realiza. Puede ver cabeceras de solicitud o de respuesta, el cuerpo de respuesta, el código de estado y los intervalos de error y rendimiento (duración total, tiempo de conexión del TCP, tiempo de enlace de TLS, tiempo de primer byte y tiempo de transferencia de contenido). Todas las solicitudes HTTP que utilizan el módulo HTTP o HTTPS que no se ven a simple vista se capturan aquí.

De forma predeterminada, en los canaries de la API, la cabecera de solicitud, la de respuesta, el cuerpo de la solicitud y el cuerpo de la respuesta no se incluyen en el informe por razones de seguridad. Si elige incluirlos, los datos se almacenan solo en el bucket de S3. Para obtener más información acerca de los datos que se incluyen en los informes, consulte executeHttpStep(stepName, requestOptions, [callback], [stepConfig]).

Los tipos de contenido del cuerpo de la respuesta que se admiten son texto, HTML y JSON. Los tipos de contenido como texto/HTML, texto/plano, aplicación/JSON y aplicación/x-amz-json-1.0 son compatibles. No se admiten las respuestas comprimidas.

- La pestaña Monitoring (Supervisión) muestra gráficos de las métricas de CloudWatch que el valor controlado publica. Para obtener más información sobre estas métricas, consulte Métricas de CloudWatch que los canaries publican.

  Debajo de los gráficos de CloudWatch que el valor controlado publica se encuentran gráficos de métricas de Lambda relacionadas con el código Lambda del valor controlado.
- La pestaña Configuration (Configuración) muestra información de configuración y programación sobre el valor controlado.
- La pestaña Groups (Grupos) muestra los grupos a los que está asociado este valor controlado, si los hay.
- La pestaña Tags (Etiquetas) muestra las etiquetas asociadas con el valor controlado.

Métricas de CloudWatch que los canaries publican

Los canaries publican las siguientes métricas en CloudWatch en el espacio de nombres `CloudWatchSynthetics`. Para obtener información sobre cómo ver las métricas de CloudWatch, consulte Ver métricas disponibles.

Métrica	Descripción
`SuccessPercent`	El porcentaje de las ejecuciones de este valor controlado que tienen éxito y no encuentran errores. Dimensiones válidas: CanaryName Estadísticas válidas: Promedio Unidades: porcentaje
`Duration`	La duración en milisegundos de la ejecución del valor controlado. Dimensiones válidas: CanaryName Estadísticas válidas: Promedio Unidades: milisegundos
`Errors`	El número de veces que el valor controlado no pudo ejecutar su script completo. Dimensiones válidas: CanaryName Estadísticas válidas: suma
`2xx`	Número de solicitudes de red que el valor controlado ha realizado que devolvieron respuestas OK, con códigos de respuesta entre 200 y 299. Esta métrica se notifica para los canarios de al UI que utilizan la versión de tiempo de ejecución `syn-nodejs-2.0` o una posterior, y se informa de los canaries de la API que utilizan la versión de tiempo de ejecución `syn-nodejs-2.2` o una posterior. Dimensiones válidas: CanaryName Estadísticas válidas: suma Unidades: recuento

Métrica	Descripción
4xx	Número de solicitudes de red que el valor controlado ha realizado que devolvieron respuestas de error, con códigos de respuesta entre 400 y 499. Esta métrica se notifica para canarios de la UI que utilizan la versión de tiempo de ejecución `syn-nodejs-2.0` o una posterior, y se informa de los canaries de la API que utilizan la versión de tiempo de ejecución `syn-nodejs-2.2` o una posterior. Dimensiones válidas: CanaryName Estadísticas válidas: suma Unidades: recuento
5xx	Número de solicitudes de red que el valor controlado ha realizado que devolvieron respuestas a errores, con códigos de respuesta entre 500 y 599. Esta métrica se notifica para canaries de la UI que utilizan la versión de tiempo de ejecución `syn-nodejs-2.0` o una posterior, y se informa de los canaries de la API que utilizan la versión en tiempo de ejecución `syn-nodejs-2.2` o una posterior. Dimensiones válidas: CanaryName Estadísticas válidas: suma Unidades: recuento

Métrica	Descripción
`Failed`	El número de ejecuciones de las ejecuciones del valor controlado que no se ejecutaron. Estos errores están relacionados con el valor controlado en sí. Dimensiones válidas: CanaryName Estadísticas válidas: suma Unidades: recuento
`Failed requests`	El número de solicitudes HTTP que el valor controlado ejecuta en el sitio web de destino que fallaron sin respuesta. Dimensiones válidas: CanaryName Estadísticas válidas: suma Unidades: recuento
`VisualMonitoringSuccessPercent`	Porcentaje de comparaciones visuales que coincidieron correctamente con las capturas de pantalla de línea de base durante una ejecución del valor controlado. Dimensiones válidas: CanaryName Estadísticas válidas: Promedio Unidades: porcentaje
`VisualMonitoringTotalComparisons`	El número total de comparaciones visuales que se produjeron durante una ejecución del valor controlado. Dimensiones válidas: CanaryName Unidades: recuento

> **Note**
>
> Los canaries que utilizan los métodos `executeStep()` o `executeHttpStep()` de la biblioteca de Synthetics también publican las métricas de `SuccessPercent` y `Duration` con las dimensiones `CanaryName` y `StepName` para cada paso.

Edición o eliminación de un valor controlado

Se puede editar o eliminar un valor controlado existente.

Editar el valor controlado

Cuando se edita un valor controlado, aunque no se cambie la programación, se restablece la programación correspondiente a cuando se edita el valor controlado. Por ejemplo, si tiene un valor controlado que se ejecuta cada hora y edita ese valor controlado, el valor controlado se ejecutará inmediatamente después de completar la edición y, a continuación, cada hora después de eso.

Para editar o actualizar un valor controlado

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Señales de aplicación, Valores controlados de Synthetics.
3. Seleccione el botón situado junto al nombre del valor controlado y elija Actions (Acciones), Edit (Editar).
4. (Opcional) Si este canario lleva a cabo la supervisión visual de las capturas de pantalla y desea establecer la siguiente ejecución del canario como línea de base, seleccione Set next run as new baseline (Establecer la siguiente ejecución como nueva línea de base).
5. (Opcional) Si este canario lleva a cabo un supervisión visual de las capturas de pantalla y usted desea eliminar una captura de pantalla de la supervisión visual o desea designar partes de la captura de pantalla que se ignorarán durante las comparaciones visuales, bajo el título Visual Monitoring (Supervisión visual), elija Edit Baseline (Editar línea de base).

   Aparece la captura de pantalla y puede elegir una de las siguientes opciones:

   - Para eliminar la captura de pantalla de ser utilizada para la supervisión visual, seleccione Remove screenshot from visual test baselin (Eliminar captura de pantalla de la línea de base de prueba visual).

- Para designar las partes de la captura de pantalla que se ignorarán durante las comparaciones visuales, haga clic y arrastre para dibujar áreas de la pantalla que se deben ignorar. Una vez que haya hecho esto en todas las áreas que desea ignorar durante las comparaciones, elija Save (Guardar).

6. Realice los demás cambios en el valor controlado que desee y elija Save (Guardar).

Eliminar el valor controlado

Al eliminar un valor controlado, puede elegir si desea eliminar también otros recursos utilizados y creados por el valor controlado. Al eliminar un valor controlado, también debe eliminar lo siguiente:

- Las capas y funciones de Lambda utilizadas por este valor controlado. Su prefijo es `cwsyn-`*`MyCanaryName`*.
- Las alarmas de CloudWatch que se crearon para este valor controlado. Estas alarmas tienen un nombre que comienza con `Synthetics-Alarm-`*`MyCanaryName`*. Para obtener más información sobre la eliminación de alarmas, consulte Edición o eliminación de una alarma de CloudWatch.
- Objetos y buckets de Amazon S3, como la ubicación del artefacto y la ubicación de los resultados del valor controlado.
- Roles de IAM creados para el valor controlado. Tienen el nombre `role/service-role/CloudWatchSyntheticsRole-`*`MyCanaryName`*.
- Grupos de Registros de CloudWatch que se han crearon para el valor controlado. Estos grupos de registros tienen los siguientes nombres: `/aws/lambda/cwsyn-`*`MyCanaryName`*`-`*`randomId`*.

Antes de eliminar un valor controlado, es posible que desee ver los detalles del valor controlado y tomar nota de esta información. De esta forma, puede eliminar los recursos correctos después de eliminar el valor controlado.

Para eliminar un valor controlado

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Señales de aplicación, Valores controlados de Synthetics.
3. Si el valor controlado se encuentra actualmente en el estado `RUNNING`, debe detenerlo. Solo valores controlados en los estados `STOPPED`, `READY(NOT_STARTED)` o bien `ERROR` se pueden eliminar.

Seleccione el botón situado junto al nombre del valor controlado y elija Actions (Acciones), Stop (Detener) para detener el valor controlado.

4. Seleccione el botón situado junto al nombre del valor controlado y elija Actions (Acciones), Delete (Eliminar).
5. Elija si desea eliminar también los demás recursos creados y utilizados por el valor controlado. Esto incluye la función y las capas Lambda, así como el rol de IAM y la política de IAM del valor controlado.

   Para eliminar el rol de IAM del valor controlado y la política de IAM, debe tener permisos suficientes. Para obtener más información, consulte Políticas administradas (predefinidas) de AWS para CloudWatch Synthetics.
6. Escriba **`Delete`** en el cuadro y elija Delete (Eliminar).
7. Elimine los demás recursos utilizados y creados para el valor controlado, como se ha indicado anteriormente en esta sección.

Inicio, detención, eliminación o actualización del tiempo de ejecución de varios valores controlados

Puede detener, iniciar, eliminar o actualizar el tiempo de ejecución de hasta cinco valores controlados con una sola acción. Si actualiza el tiempo de ejecución de un valor controlado, se actualiza al último tiempo de ejecución disponible para el idioma y el marco que utilice el valor controlado.

Si selecciona varios valores controlados y solo algunos de ellos se encuentran en un estado válido para la acción que seleccione, la acción se realizará únicamente en los valores controlados en los que esa acción sea válida. Por ejemplo, si selecciona algunos valores controlados que se estén ejecutando actualmente y otros que no, y selecciona iniciar los valores controlados, se iniciarán los valores controlados que aún no estuvieran en ejecución y los que ya estaban en ejecución no se verán afectados.

Si ninguno de los valores controlados que seleccione es válido para una acción, esa acción no estará disponible en el menú.

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Señales de aplicación, Valores controlados de Synthetics.

3. Seleccione las casillas de verificación situadas junto a los valores controlados que quiera detener, iniciar o eliminar.
4. Elija Actions (Acciones) y, a continuación, elija Start (Iniciar), Stop (Detener), Delete (Eliminar) o Update Runtime (Actualizar tiempo de ejecución).

Supervisión de eventos del valor controlado con Amazon EventBridge

Las reglas de eventos de Amazon EventBridge pueden notificarle cuando los canaries cambian de estado o completan las ejecuciones. EventBridge proporciona un flujo casi en tiempo real de los eventos del sistema que describen cambios en los recursos de AWS. CloudWatch Synthetics envía estos eventos a EventBridge en una base el mejor esfuerzo. La entrega del mejor esfuerzo significa que CloudWatch Synthetics intenta enviar todos los eventos a EventBridge, pero en algunos casos raros es posible que no se entregue un evento. EventBridge procesa todos los eventos que se han recibido al menos una vez. Además, los agentes de escucha de eventos podrían no recibir los eventos en el orden en el que los eventos han ocurrido.

> **Note**
>
> Amazon EventBridge: es un servicio conductor de eventos que se puede utilizar para conectar las aplicaciones con datos de varias fuentes. Para obtener más información, consulte What is Amazon EventBridge? (¿Qué es Amazon EventBridge?) en la Guía del usuario de Amazon EventBridge.

CloudWatch Synthetics emite un evento cuando un valor controlado cambia de estado o completa una ejecución. Se puede crear una regla de EventBridge que incluya un patrón de eventos para que coincida con todos los tipos de eventos que se han enviado desde CloudWatch Synthetics o que coincida únicamente con tipos de eventos específicos. Cuando un valor controlado desencadena una regla, EventBridge invoca las acciones de destino definidas en la regla. Esto le permite enviar notificaciones, capturar información sobre el evento y tomar medidas correctivas en respuesta a un cambio de estado del valor controlado o a la terminación de la ejecución del valor controlado. Por ejemplo, puede crear reglas para los siguientes casos de uso:

- Investigación sobre el fallo de una ejecución de un valor controlado
- Investigación sobre un valor controlado que ha entrado en estado de `ERROR`
- Seguimiento del ciclo de vida de un valor controlado

- Supervisión del éxito o error de ejecución de un canario como parte de un flujo de trabajo

Eventos de muestra de CloudWatch Synthetics

En esta sección se enumeran los eventos de muestra de CloudWatch Synthetics. Para obtener más información sobre el formato del evento, consulte Events and Event Patterns in EventBridge (Eventos y patrones de eventos en EventBridge).

Cambio del estado del valor controlado

En este tipo de evento, los valores de `current-state` y `previous-state` pueden ser los siguientes:

`CREATING` | `READY` | `STARTING` | `RUNNING` | `UPDATING` | `STOPPING` | `STOPPED` | `ERROR`

```
{
                "version": "0",
                "id": "8a99ca10-1e97-2302-2d64-316c5dedfd61",
                "detail-type": "Synthetics Canary Status Change",
                "source": "aws.synthetics",
                "account": "123456789012",
                "time": "2021-02-09T22:19:43Z",
                "region": "us-east-1",
                "resources": [],
                "detail": {
                                "account-id": "123456789012",
                                "canary-id": "EXAMPLE-dc5a-4f5f-96d1-989b75a94226",
                                "canary-name": "events-bb-1",
                                "current-state": "STOPPED",
                                "previous-state": "UPDATING",
                                "source-location": "NULL",
                                "updated-on": 1612909161.767,
                                "changed-config": {
                                                "executionArn": {
                                                                "previous-value":
 "arn:aws:lambda:us-east-1:123456789012:function:cwsyn-events-bb-1-af3e3a05-
dc5a-4f5f-96d1-989EXAMPLE:1",
                                                                "current-value":
 "arn:aws:lambda:us-east-1:123456789012:function:cwsyn-events-bb-1-af3e3a05-
dc5a-4f5f-96d1-989EXAMPLE:2"
                                                },
                                                "vpcId": {
                                                                "current-value": "NULL"
```

```
                                                },
                                                "testCodeLayerVersionArn": {
                                                                "previous-
value": "arn:aws:lambda:us-east-1:123456789012:layer:cwsyn-events-bb-1-af3e3a05-
dc5a-4f5f-96d1-989EXAMPLE:1",
                                                                "current-value":
 "arn:aws:lambda:us-east-1:123456789012:layer:cwsyn-events-bb-1-af3e3a05-
dc5a-4f5f-96d1-989EXAMPLE:2"
                                                }
                                },
                                "message": "Canary status has changed"
                }
}
```

Se ha completado con éxito la ejecución del valor controlado

```
{
                "version": "0",
                "id": "989EXAMPLE-f4a5-57a7-1a8f-d9cc768a1375",
                "detail-type": "Synthetics Canary TestRun Successful",
                "source": "aws.synthetics",
                "account": "123456789012",
                "time": "2021-02-09T22:24:01Z",
                "region": "us-east-1",
                "resources": [],
                "detail": {
                                "account-id": "123456789012",
                                "canary-id": "989EXAMPLE-dc5a-4f5f-96d1-989b75a94226",
                                "canary-name": "events-bb-1",
                                "canary-run-id": "c6c39152-8f4a-471c-9810-989EXAMPLE",
                                "artifact-location": "cw-syn-results-123456789012-us-
east-1/canary/us-east-1/events-bb-1-ec3-28ddbe266797/2021/02/09/22/23-41-200",
                                "test-run-status": "PASSED",
                                "state-reason": "null",
                                "canary-run-timeline": {
                                                "started": 1612909421,
                                                "completed": 1612909441
                                },
                                "message": "Test run result is generated successfully"
                }
}
```

Error en la ejecución del valor controlado

```
{
                "version": "0",
                "id": "2644b18f-3e67-5ebf-cdfd-bf9f91392f41",
                "detail-type": "Synthetics Canary TestRun Failure",
                "source": "aws.synthetics",
                "account": "123456789012",
                "time": "2021-02-09T22:24:27Z",
                "region": "us-east-1",
                "resources": [],
                "detail": {
                                "account-id": "123456789012",
                                "canary-id": "af3e3a05-dc5a-4f5f-96d1-9989EXAMPLE",
                                "canary-name": "events-bb-1",
                                "canary-run-id": "0df3823e-7e33-4da1-8194-
b04e4d4a2bf6",
                                "artifact-location": "cw-syn-results-123456789012-us-
east-1/canary/us-east-1/events-bb-1-ec3-989EXAMPLE/2021/02/09/22/24-21-275",
                                "test-run-status": "FAILED",
                                "state-reason": "\"Error: net::ERR_NAME_NOT_RESOLVED
 \""
                                "canary-run-timeline": {
                                                "started": 1612909461,
                                                "completed": 1612909467
                                },
                                "message": "Test run result is generated successfully"
                }
}
```

Es posible que los eventos se dupliquen o estén desordenados. Para determinar el orden de los eventos, utilice la propiedad `time`.

Prerequisitos para crear las reglas de EventBridge

Antes de crear una regla de EventBridge para CloudWatch Synthetics, debe hacer lo siguiente:

- Familiarizase con los eventos, las reglas y los destinos de EventBridge.
- Cree y configure los destinos que las reglas de EventBridge han invocado. Las reglas pueden invocar muchos tipos de destinos, entre los que se incluyen:
  - Temas de Amazon SNS
  - Funciones de AWS Lambda
  - Flujos de Kinesis

- Colas de Amazon SQS

Para obtener más información, consulte What is Amazon EventBridge? (¿Qué es Amazon EventBridge?) y Getting started with Amazon EventBridge (Introducción a Amazon EventBridge) en la Guía del usuario de Amazon EventBridge.

Cree una regla de EventBridge (CLI)

Los pasos del ejemplo siguiente crean una regla de EventBridge que publica un tema de Amazon SNS cuando el valor controlado que se denomina `my-canary-name` en `us-east-1`completa una ejecución o cambia de estado.

1. Crear la regla.

```
aws events put-rule \
  --name TestRule \
  --region us-east-1 \
  --event-pattern "{\"source\": [\"aws.synthetics\"], \"detail\": {\"canary-name\":
 [\"my-canary-name\"]}}"
```

   Las propiedades que se omiten en el patrón no se tienen en cuenta.

2. Añada el tema como destino de la regla.

   - Reemplace *`tema-arn`* por el nombre de recurso de Amazon (ARN) del tema de Amazon SNS.

```
aws events put-targets \
  --rule TestRule \
  --targets "Id"="1","Arn"="topic-arn"
```

> **Note**
>
> Para permitir que Amazon EventBridge llame al tema de destino, debe agregar en el tema una política basada en recursos. Para obtener más información, consulte Amazon SNS permissions (Permisos de Amazon SNS) en la Guía del usuario de Amazon EventBridge.

Para obtener más información, consulte Events and event patterns in EventBridge (Eventos y patrones de eventos en EventBridge) en la Guía del usuario de Amazon EventBridge.

Realice lanzamientos y experimentos A/B con CloudWatch Evidently

Puede utilizar Amazon CloudWatch Evidently para validar nuevas características de forma segura si se las proporciona a un porcentaje específico de sus usuarios mientras implementa la característica. Puede monitorear el rendimiento de la nueva característica para decidir cuándo aumentar el tráfico hacia los usuarios. Esto ayuda a reducir los riesgos e identificar las consecuencias no deseadas antes de lanzar la característica por completo.

También puede llevar a cabo experimentos A/B para tomar decisiones de diseño de características basadas en pruebas y datos. Un experimento puede realizar pruebas de un máximo de cinco variaciones a la vez. Evidently recopila datos de experimentos y los analiza a través de métodos estadísticos. También brinda recomendaciones claras sobre qué variaciones tienen mejor rendimiento. Puede probar las funciones orientadas al usuario y las características de backend.

Precios de Evidently

Evidently cobra la cuenta en función de eventos Evidently y unidades de análisis Evidently. Los eventos de Evidently incluyen eventos de datos, como clics y vistas de página, así como eventos de asignación que determinan la variación de características que se le presentan a un usuario.

Las unidades de análisis de Evidently se generan a partir de eventos de Evidently, según las reglas que haya creado en Evidently. Las unidades de análisis son la cantidad de coincidencias de reglas en los eventos. Por ejemplo, un evento de clic de usuario podría producir una única unidad de análisis de Evidently, un recuento de clics. Otro ejemplo es un evento de pago de usuario que podría producir dos unidades de análisis de Evidently, el valor del pago y la cantidad de artículos del carrito. Para obtener más información sobre precios, consulte Precios de Amazon CloudWatch.

Cloudwatch Evidently está disponible actualmente en las siguientes regiones:

- US East (Ohio)
- Este de EE. UU. (Norte de Virginia)
- Oeste de EE. UU. (Oregón)
- Asia Pacífico (Singapur)
- Asia-Pacífico (Sídney)

- Asia-Pacífico (Tokio)
- Europa (Fráncfort)
- Europa (Irlanda)
- Europa (Estocolmo)

Temas

Políticas de IAM para usar Evidently

Para administrar Cloudwatch Evidently, debe haber iniciado sesión como usuario o rol de IAM que tenga los siguientes permisos:

- La política de AmazonCloudWatchEvidentlyFullAccess
- La política de ResourceGroupsandTagEditorReadOnlyAccess

Además, para poder crear un proyecto que almacene eventos de evaluación en Amazon S3 o CloudWatch Logs, se necesitan los siguientes permisos:

```
{
    "Version": "2012-10-17",
```

```
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetBucketPolicy",
                "s3:PutBucketPolicy",
                "s3:GetObject",
                "s3:ListBucket"
            ],
            "Resource": "arn:aws:s3:::*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogDelivery",
                "logs:DeleteLogDelivery",
                "logs:DescribeResourcePolicies",
                "logs:PutResourcePolicy"
            ],
            "Resource": [
                "*"
            ]
        }
    ]
}
```

Permisos adicionales para la integración de CloudWatch RUM

Además, si tiene la intención de administrar lanzamientos o experimentos de Evidently que se integran con Amazon CloudWatch RUM y utilizan métricas de Cloudwatch RUM para monitorear, necesita la política de AmazonCloudWatchRUMFullAccess. Para crear un rol de IAM para otorgarle al cliente web de CloudWatch RUM permiso para enviar datos a CloudWatch RUM, necesita los siguientes permisos:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iam:CreateRole",
                "iam:CreatePolicy",
                "iam:AttachRolePolicy"
```

```
            ],
            "Resource": [
                "arn:aws:iam::*:role/service-role/CloudWatchRUMEvidentlyRole-*",
                "arn:aws:iam::*:policy/service-role/CloudWatchRUMEvidentlyPolicy-*"
            ]
        }
    ]
}
```

Permisos de acceso de solo lectura a Evidently

Puede otorgar la política de AmazonCloudWatchEvidentlyReadOnlyAccess para otros usuarios que necesitan ver los datos de Evidently, pero no necesitan crear recursos Evidently.

Creación de proyectos, características, lanzamientos y experimentos

Para empezar a utilizar CloudWatch Evidently, ya sea para un lanzamiento de características o para un experimento A/B, primero cree un proyecto. Un proyecto es una agrupación lógica de recursos. Dentro del proyecto, puede crear características que tienen variaciones que puede probar o lanzar. Puede crear una característica antes de crear un lanzamiento o experimento, o al mismo tiempo.

Temas

Crear un nuevo proyecto de

Siga estos pasos para configurar un proyecto nuevo de CloudWatch Evidently.

Para crear un proyecto nuevo de CloudWatch Evidently

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Señales de aplicación, Evidently.
3. Elija Crear proyecto.

4. En Project name (Nombre del proyecto), ingrese un nombre que se utilizará para identificar este proyecto en la consola de CloudWatch Evidently.

   También puede agregar una descripción opcional.

5. Para Almacenamiento de eventos de evaluación, elija si desea almacenar los eventos de evaluación que recopila con Evidently. Incluso si no almacena estos eventos, Evidently los agrega para crear métricas y otros datos de experimentos que puede ver en el panel de Evidently. Para obtener más información, consulte Almacenamiento de datos del proyecto.

6. En Use client-side evaluation (Uso de la evaluación del cliente), elija si desea habilitar la evaluación del cliente para este proyecto. Con la evaluación del cliente, la aplicación puede asignar variaciones a las sesiones de los usuarios de forma local en lugar de llamar a la operación EvaluateFeature. Esto mitiga los riesgos de latencia y disponibilidad que conlleva una llamada a la API. Para obtener más información, consulte Uso de la evaluación del cliente con tecnología de AWS AppConfig.

   Para crear un proyecto con evaluación del cliente, debe tener el permiso `evidently:ExportProjectAsConfiguration`.

   Si habilita la evaluación del cliente, haga también lo siguiente:

   a. Elija si desea utilizar una aplicación de AWS AppConfig existente o crear una nueva.

   b. Elija si desea utilizar un entorno de AWS AppConfig existente o crear uno nuevo.

      Para obtener más información acerca de las aplicaciones y los entornos de AWS AppConfig, consulte Cómo funciona AWS AppConfig.

7. (Opcional) Para agregar etiquetas a este proyecto, elija Tags (Etiquetas), Add new tag (Agregar nueva etiqueta).

   Luego, en Key (Clave), ingrese un nombre para la etiqueta. Puede agregar un valor opcional para la etiqueta en Valor.

   Para agregar otra etiqueta, vuelva a elegir Add new tag (Agregar nueva etiqueta).

   Para obtener más información, consulte Tagging AWS Resources (Etiquetado de recursos de).

8. Elija Crear proyecto.

Uso de la evaluación del cliente con tecnología de AWS AppConfig

Puede usar la evaluación del cliente con tecnología de AWS AppConfig (evaluación de cliente) en un proyecto, lo que permite que la aplicación asigne variaciones a las sesiones de los usuarios de forma local en lugar de asignar variaciones llamando a la operación EvaluateFeature. Esto mitiga los riesgos de latencia y disponibilidad que conlleva una llamada a la API.

Para utilizar la evaluación del cliente, adjunte la extensión de Lambda AWS AppConfig como capa para sus funciones de Lambda y configure las variables de entorno. La evaluación del cliente se ejecuta como un proceso secundario en el host local. A continuación, puede llamar a las operaciones EvaluationFeature y PutProjectEvent con respecto a `localhost`. El proceso de evaluación del cliente gestiona la asignación de variaciones, el almacenamiento en caché y la sincronización de datos. Para obtener más información sobre cómo funciona AWS AppConfig, consulte Cómo funciona AWS AppConfig.

Al integrar con AWS AppConfig, especifique un ID de aplicación de AWS AppConfig y un ID de entorno de AWS AppConfig a Evidently. Puede usar el mismo ID de aplicación e ID de entorno en todos los proyectos de Evidently.

Al crear un proyecto con la evaluación del cliente habilitada, Evidently crea un perfil de configuración de AWS AppConfig para ese proyecto. El perfil de configuración de cada proyecto será diferente.

Control de acceso de evaluación del cliente

La evaluación del cliente de Evidently utiliza un mecanismo de control de acceso diferente al del resto de Evidently. Le recomendamos encarecidamente que comprenda esto para poder implementar las medidas de seguridad adecuadas.

Con Evidently, puede crear políticas de IAM que limiten las acciones que el usuario puede llevar a cabo con los recursos individuales. Por ejemplo, puede crear un rol que no permita al usuario tener la acción EvaluateFeature. Para obtener más información sobre las acciones de Evidently que se pueden controlar con políticas de IAM, consulte Acciones definidas por Amazon CloudWatch Evidently.

El modelo de evaluación del cliente permite llevar a cabo evaluaciones locales de las características de Evidently que utilizan los metadatos del proyecto. Un usuario de un proyecto con la evaluación del cliente habilitada puede llamar a la API EvaluateFeature con respecto a un punto de conexión de host local, y esta llamada a la API no llega a Evidently y no se autentica mediante las políticas de IAM del servicio de Evidently. Esta llamada se lleva a cabo correctamente incluso si el usuario

no tiene el permiso de IAM para usar la acción EvaluateFeature. Sin embargo, el usuario sigue necesitando el permiso PutProjectEvents para que el agente almacene en búfer los eventos de evaluación o los personalizados y descargue datos a Evidently de forma asíncrona.

Además, el usuario debe tener el permiso `evidently:ExportProjectAsConfiguration` para poder crear un proyecto que utilice la evaluación del cliente. Esto le ayuda a controlar el acceso a acciones EvaluateFeature que se invocan durante la evaluación del cliente.

Si no tiene cuidado, el modelo de seguridad de evaluación del cliente puede subvertir las políticas que ha establecido en el resto de Evidently. Un usuario que tiene el permiso `evidently:ExportProjectAsConfiguration` puede crear un proyecto con la evaluación del cliente habilitada y, a continuación, utilizar la acción EvaluateFeature para la evaluación del cliente con ese proyecto, incluso si se le deniega expresamente la acción EvaluateFeature en una política de IAM.

Introducción a Lambda

Actualmente, Evidently admite la evaluación del cliente mediante el uso de un entorno de AWS Lambda. Para comenzar, primero decida qué aplicación y entorno de AWS AppConfig se va a utilizar. Elija una aplicación y un entorno existentes o cree otros nuevos.

Los siguientes comandos de muestra de la AWS CLI de AWS AppConfig crean una aplicación y un entorno.

```
aws appconfig create-application --name YOUR_APP_NAME
```

```
aws appconfig create-environment --application-id YOUR_APP_ID --
name YOUR_ENVIRONMENT_NAME
```

A continuación, cree un proyecto de Evidently utilizando estos recursos de AWS AppConfig. Para obtener más información, consulte Crear un nuevo proyecto de .

Lambda admite la evaluación del cliente mediante el uso de una capa de Lambda. Se trata de una capa pública que forma parte de `AWS-AppConfig-Extension`, una extensión pública de AWS AppConfig creada por el servicio AWS AppConfig. Para obtener más información acerca de las capas de Lambda, consulte Layer (Capa).

Para usar la evaluación del cliente, debe agregar esta capa a la función de Lambda y configurar los permisos y las variables de entorno.

Para agregar la capa de Lambda de la evaluación del cliente de Evidently a la función de Lambda y configurarla

1. Si aún no lo ha hecho, cree una función de Lambda.
2. Agregue la capa de evaluación del cliente a la función. Puede especificar su ARN o seleccionarlo de la lista de capas de AWS si aún no lo ha hecho. Para obtener más información, consulte Configuración de funciones para utilizar capas y Versiones disponibles de la extensión de Lambda AWS AppConfig.
3. Cree una política de IAM denominada EvidentlyAppConfigCachingAgentPolicy con el siguiente contenido y adjúntela al rol de ejecución de la función. Para obtener más información, consulte Lambda execution role (Rol de ejecución de Lambda).

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": [
                "appconfig:GetLatestConfiguration",
                "appconfig:StartConfigurationSession",
                "evidently:PutProjectEvents"
            ],
            "Resource": "*"
        }
    ]
}
```

4. Agregue la variable de entorno `AWS_APPCONFIG_EXTENSION_EVIDENTLY_CONFIGURATIONS` obligatoria a la función de Lambda. Esta variable de entorno especifica la asignación entre el proyecto de Evidently y los recursos de AWS AppConfig.

   Si utiliza esta función para un proyecto de Evidently, defina el valor de la variable de entorno en: `applications/`*`APP_ID`*`/environments/`*`ENVIRONMENT_ID`*`/configurations/`*`PROJECT_NAME`*

   Si utiliza esta función para varios proyectos de Evidently, utilice una coma para separar los valores, como en el siguiente ejemplo: `applications/`*`APP_ID_1`*`/environments/`*`ENVIRONMENT_ID_1`*`/configurations/`*`PROJECT_NAME_1`*`,`

`applications/APP_ID_2/environments/ENVIRONMENT_ID_2/configurations/PROJECT_NAME_2`

5. (Opcional) Defina otras variables de entorno. Para más información, consulte Configuración de la extensión de Lambda de AWS AppConfig.

6. En la aplicación, envíe `EvaluateFeature` a `localhost` para obtener evaluaciones de Evidently localmente.

   Ejemplo de Python:

```
import boto3
from botocore.config import Config

def lambda_handler(event, context):
    local_client = boto3.client(
        'evidently',
        endpoint_url="http://localhost:2772",
        config=Config(inject_host_prefix=False)
    )
    response = local_client.evaluate_feature(
        project=event['project'],
        feature=event['feature'],
        entityId=event['entityId']
    )
    print(response)
```

   Ejemplo de Node.js:

```
const AWS = require('aws-sdk');
const evidently = new AWS.Evidently({
    region: "us-west-2",
    endpoint: "http://localhost:2772",
    hostPrefixEnabled: false
});

exports.handler = async (event) => {

    const evaluation = await evidently.evaluateFeature({
        project: 'John_ETCProject_Aug2022',
        feature: 'Feature_IceCreamFlavors',
        entityId: 'John'
    }).promise()
```

```
    console.log(evaluation)
    const response = {
        statusCode: 200,
        body: evaluation,
    };
    return response;
};
```

Ejemplo de Kotlin:

```
String localhostEndpoint = "http://localhost:2772/"
public AmazonCloudWatchEvidentlyClient getEvidentlyLocalClient() {
    return AmazonCloudWatchEvidentlyClientBuilder.standard()

 .withEndpointConfiguration(AwsClientBuilder.EndpointConfiguration(localhostEndpoint,
 region))

 .withClientConfiguration(ClientConfiguration().withDisableHostPrefixInjection(true))
        .withCredentials(credentialsProvider)
        .build();
}

AmazonCloudWatchEvidentlyClient evidently = getEvidentlyLocalClient();

// EvaluateFeature via local client.
EvaluateFeatureRequest evaluateFeatureRequest = new
 EvaluateFeatureRequest().builder()
   .withProject(${YOUR_PROJECT}) //Required.
   .withFeature(${YOUR_FEATURE}) //Required.
   .withEntityId(${YOUR_ENTITY_ID}) //Required.
   .withEvaluationContext(${YOUR_EVAL_CONTEXT}) //Optional: a JSON object of
 attributes that you can optionally pass in as part of the evaluation event sent to
 Evidently.
   .build();

EvaluateFeatureResponse evaluateFeatureResponse =
 evidently.evaluateFeature(evaluateFeatureRequest);

// PutProjectEvents via local client.
PutProjectEventsRequest putProjectEventsRequest = new
 PutProjectEventsRequest().builder()
    .withData(${YOUR_DATA})
```

```
    .withTimeStamp(${YOUR_TIMESTAMP})
    .withType(${YOUR_TYPE})
    .build();

PutProjectEvents putProjectEventsResponse =
 evidently.putProjectEvents(putProjectEventsRequest);
```

Configuración de la frecuencia con la que el cliente envía datos a Evidently

Para especificar la frecuencia con la que la evaluación del cliente envía datos a Evidently, tiene la opción de configurar dos variables de entorno.

- `AWS_APPCONFIG_EXTENSION_EVIDENTLY_EVENT_BATCH_SIZE` especifica el número de eventos por proyecto que se van a agrupar antes de enviarlos a Evidently. Los valores válidos son enteros entre 1 y 50, y el valor predeterminado es 40.
- `AWS_APPCONFIG_EXTENSION_EVIDENTLY_BATCH_COLLECTION_DURATION` especifica la duración en segundos de espera de los eventos antes de enviarlos a Evidently. El valor predeterminado es 30.

Solución de problemas

Utilice la siguiente información para poder solucionar los problemas al usar CloudWatch Evidently con la evaluación del cliente con tecnología de AWS AppConfig.

Se produjo un error (BadRequestException) al llamar a la operación EvaluateFeature: método HTTP no admitido para la ruta proporcionada

Es posible que las variables de entorno no estén configuradas correctamente. Por ejemplo, es posible que haya utilizado `EVIDENTLY_CONFIGURATIONS` como el nombre de la variable de entorno en lugar de `AWS_APPCONFIG_EXTENSION_EVIDENTLY_CONFIGURATIONS`.

ResourceNotFoundException: implementación no encontrada

La actualización de los metadatos del proyecto no se implementó en AWS AppConfig. Compruebe si hay una implementación activa en el entorno de AWS AppConfig que utilizó para la evaluación del cliente.

ValidationException: no hay configuración de Evidently para el proyecto

Es posible que la variable de entorno `AWS_APPCONFIG_EXTENSION_EVIDENTLY_CONFIGURATIONS` esté configurada con un nombre de proyecto incorrecto.

Agregue una característica a un proyecto

Una característica en CloudWatch Evidently representa una característica que desea lanzar o de la que desea probar variaciones.

Para poder agregar una característica, debe crear un proyecto. Para obtener más información, consulte Crear un nuevo proyecto de .

Para agregar una característica a un proyecto

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Señales de aplicación, Evidently.
3. Elija el nombre del proyecto.
4. Elija Add feature (Agregar característica).
5. En Feature name (Nombre de la característica), ingrese un nombre que se utilizará para identificar esta característica dentro de este proyecto.

   También puede agregar una descripción opcional de la característica.
6. En Feature variations (Variaciones de características), para Variation type (Tipo de variación), elija Boolean (Booleano), Long (Largo), Double (Doble) o String (Cadena). Para obtener más información, consulte Tipos de variaciones.
7. Agregue un máximo de cinco variaciones para la característica. El Value (Valor) para cada variación debe ser válido para el Variation type (Tipo de variación) que haya seleccionado.

   Seleccione una de las variaciones como predeterminada. Este es el punto de partida con el que se compararán las demás variaciones y debería ser la variación que se está presentando ante los usuarios ahora. Esta es también la variación que se les ofrece a los usuarios que no se agregan a un lanzamiento o experimento para esta característica.
8. Elija Sample code (Código de muestra). En el ejemplo de código, observará lo que necesita agregar a la aplicación para configurar las variaciones y asignarles sesiones de usuario. Puede elegir entre JavaScript, Java y Python para el código.

No es necesario que agregue el código a su aplicación en este momento, pero debe hacerlo antes de iniciar un lanzamiento o un experimento.

Para obtener más información, consulte Agregue código a la aplicación.

9. (Opcional) Para especificar que determinados usuarios siempre vean una variación determinada, elija Overrides (Anulaciones), Add override (Agregar anulación). Luego, especifique un usuario; para ello, ingrese su ID de usuario, ID de cuenta u otro identificador en Identifier (Identificador) y especifique la variación que deben ver.

   Esto puede ser útil para los miembros de su propio equipo de pruebas u otros usuarios internos cuando quiera asegurarse de que vean una variación específica. Las sesiones de los usuarios a los que se les asignan anulaciones no contribuyen a las métricas de lanzamiento o experimentación.

   Puede repetir esta acción para un máximo de 20 usuarios si selecciona Agregar anulación de nuevo.

10. (Opcional) Para agregar etiquetas a esta característica, elija Tags (Etiquetas), Add new tag (Agregar nueva etiqueta).

    Luego, en Key (Clave), ingrese un nombre para la etiqueta. Puede agregar un valor opcional para la etiqueta en Valor.

    Para agregar otra etiqueta, vuelva a elegir Add new tag (Agregar nueva etiqueta).

    Para obtener más información, consulte Tagging AWS Resources (Etiquetado de recursos de).

11. Elija Add feature (Agregar característica).

Tipos de variaciones

Al crear una característica y definir las variaciones, debe seleccionar un variation type (tipo de variación). Los tipos posibles son:

- Booleano
- Entero largo
- Número en coma flotante de precisión doble
- Cadena

El tipo de variación establece cómo se diferencian las distintas variaciones en el código. Puede utilizar el tipo de variación para simplificar la implementación de CloudWatch Evidently y también para simplificar el proceso de modificación de las características de sus lanzamientos y experimentos.

Por ejemplo, si define una característica con el tipo de variación de enteros largos, los enteros especificados para diferenciar las variaciones pueden ser números que se pasan al código de forma directa. Un ejemplo podría ser la prueba del tamaño de píxel de un botón. Los valores de los tipos de variación pueden ser el número de que se utilizan en cada variación. El código de cada variación puede leer el valor del tipo de variación y utilizarlo como tamaño de botón. Para probar un nuevo tamaño de botón, puede cambiar el número utilizado para el valor de la variación, sin realizar ningún otro cambio de código.

Cuando establezca los valores para los tipos de variación dentro de una característica, deberá evitar asignar los mismos valores a múltiples variaciones, a menos que desee realizar pruebas A/A para probar CloudWatch Evidently o tenga otros motivos para hacerlo.

Evidently no tiene soporte nativo para JSON como tipo, pero puede pasar JSON en el tipo de variación String y analizar ese JSON en su código.

Use segmentos para centrar su audiencia

Puede definir los segmentos de la audiencia y usarlos en sus lanzamientos y experimentos. Un segmento es una parte de su audiencia que comparte una o más características. Algunos ejemplos podrían ser los usuarios del navegador Chrome, los usuarios de Europa o los usuarios del navegador Firefox en Europa que también se ajustan a otros criterios que recopila su aplicación, como la edad.

El uso de un segmento en un experimento limita ese experimento para evaluar solo a los usuarios que coinciden con los criterios del segmento. Cuando usa uno o más segmentos en un lanzamiento, puede definir diferentes divisiones de tráfico para los diferentes segmentos de audiencia.

Sintaxis de patrones de reglas del segmento

Para crear un segmento, defina un patrón de la regla de segmento. Especifique los atributos que desea utilizar para evaluar si una sesión de usuario se incluirá en el segmento. El patrón que se crea se compara con el valor de `evaluationContext` que Evidently encuentra en una sesión de usuario. Para obtener más información, consulte Utilizar EvaluateFeature.

Para crear un patrón de reglas de segmentos, especifique los campos con los que desea que coincida el patrón. También puede usar la lógica en su patrón, como `And`, `Or`, `Not` y `Exists`.

Para que un `evaluationContext` coincida con un patrón, el `evaluationContext` debe coincidir con todas las partes del patrón de la regla. Evidently, ignora los campos del `evaluationContext` que no se incluyen en el patrón de reglas.

Los valores que coinciden con los patrones de reglas siguen las reglas JSON. Puede incluir cadenas entre comillas (“), números y palabras clave `true`, `false` y `null`.

Para las cadenas, Evidently usa coincidencia exacta, de carácter a carácter, sin necesidad de cambio de mayúsculas y minúsculas ni cualquier otra normalización de cadenas. Por lo tanto, las coincidencias de reglas distinguen entre mayúsculas y minúsculas. Por ejemplo, si su `evaluationContext` incluye un atributo `browser`, pero el patrón de la regla busca `Browser`, no coincidirán.

Para los números, Evidently también usa la representación de cadenas. Por ejemplo, 300, 300.0 y 3.0e2 no se consideran iguales.

Al escribir patrones de regla escrita para buscar `evaluationContext`, puede utilizar la API `TestSegmentPattern` o el comando de la CLI `test-segment-pattern` para probar que su patrón coincida con el JSON correcto. Para obtener más información, consulte TestSegmentPattern.

El siguiente resumen muestra todos los operadores de comparación que están disponibles en los patrones de segmentos de Evidently.

Comparación	Ejemplo	Sintaxis de reglas
Nulo	El valor de UserID (ID de usuario) es nulo	`{ "UserID": [ null ] }`
Vacío	LastName (Apellido) está vacío	`{ "LastName": [""] }`
Igual a	El navegador es “Chrome”	`{ "Browser": [ "Chrome" ] }`

Comparación	Ejemplo	Sintaxis de reglas
Y	El país es “Francia” y el dispositivo es “móvil”	`{ "Country": [ "France" ], "Device": ["Mobile"] }`
O (varios valores de un solo atributo)	El navegador es “Chrome” o “Firefox”	`{ "Browser": ["Chrome", "Firefox"] }`
O (atributos diferentes)	El navegador es “Safari” o el dispositivo es “Tablet”	`{ "$or": [ {"Browser": ["Safari"]}, {"Device": ["Tablet"}] ] }`
No	El navegador es cualquier valor menos “Safari”	`{ "Browser": [ { "anything-but": [ "Safari" ] } ] }`
Valor numérico (igual a)	El valor de Price (Precio) es 100	`{ "Price": [ { "numeric": [ "=", 100 ] } ] }`

Comparación	Ejemplo	Sintaxis de reglas
Valor numérico (rango)	El valor de Price (Precio) es superior a 10 e inferior o igual a 20	`{ "Price": [ { "numeric": [ ">", 10, "<=", 20 ] } ] }`
Existe	El campo de edad existe	`{ "Age": [ { "exists": true } ] }`
No existe	El campo de edad no existe	`{ "Age": [ { "exists": false } ] }`
Comienza con un prefijo	La región se encuentra en los Estados Unidos	`{ "Region": [ {"prefix": "us-" } ] }`
Termina con un sufijo	La ubicación tiene el sufijo “Oeste”	`{ "Region": [ {"suffix": "West" } ] }`

Ejemplos de reglas de segmentos

En todos los ejemplos que siguen se presupone que se pasan valores para `evaluationContext` con los mismos valores y etiquetas de campo que utiliza en sus patrones de reglas.

El siguiente ejemplo coincide si `Browser` es Chrome o Firefox y `Location` es US-West.

```
{
```

```
  "Browser": ["Chrome", "Firefox"],
  "Location": ["US-West"]
}
```

El siguiente ejemplo coincide si `Browser` es cualquier navegador excepto Chrome, el `Location` comienza por US y existe un campo `Age`.

```
{
  "Browser": [ {"anything-but": ["Chrome"]}],
  "Location": [{"prefix": "US"}],
  "Age": [{"exists": true}]
}
```

El siguiente ejemplo coincide si `Location` es Japón y `Browser` es Safari o `Device` es Tablet.

```
{
 "Location": ["Japan"],
 "$or": [
   {"Browser": ["Safari"]},
   {"Device": ["Tablet"]}
 ]
}
```

Crear un segmento

Después de crear un segmento, puede usarlo en cualquier lanzamiento o experimento de cualquier proyecto.

Para crear un segmento

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Señales de aplicación, Evidently.
3. Seleccione la pestaña Segments (Segmentos).
4. Seleccione Create segment (Crear segmento).
5. En Segment name (Nombre de segmento), ingrese un nombre que se utilizará para identificar este segmento.

   También puede optar por agregar una descripción.

6. Para Segment pattern (Patrón de segmento), introduzca un bloque JSON que defina el patrón de la regla. Para obtener más información acerca de la sintaxis del patrón de regla, consulte Sintaxis de patrones de reglas del segmento.

Creación de un lanzamiento

Si desea exponer una nueva característica o cambiar a un porcentaje específico de los usuarios, cree un lanzamiento. Puede monitorear las métricas clave, como los tiempos de carga de páginas y las conversiones antes de implementar la característica en todos los usuarios.

Antes de agregar un lanzamiento, debe haber creado un proyecto. Para obtener más información, consulte Crear un nuevo proyecto de .

Al agregar un lanzamiento, puede utilizar una característica que ya haya creado o crear una nueva característica mientras crea el lanzamiento.

Para agregar un lanzamiento a un proyecto

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Señales de aplicación, Evidently.
3. Seleccione el botón situado junto al nombre del proyecto y elija Project actions (Acciones del proyecto), Create launch (Crear lanzamiento).
4. En Launch name (Nombre del lanzamiento), ingrese un nombre que se utilizará para identificar esta característica dentro de este proyecto.

   También puede optar por agregar una descripción.
5. Elija Select from existing features (Seleccionar una de las características existentes) o Add new feature (Agregar nueva característica).

   Si está utilizando una característica existente, selecciónela en Feature name (Nombre de la característica).

   Si elige Add new feature (Agregar nueva característica), realice lo siguiente:

   a. En Feature name (Nombre de la característica), ingrese un nombre que se utilizará para identificar esta característica dentro de este proyecto. También puede optar por agregar una descripción.

b. En Feature variations (Variaciones de características), para Variation type (Tipo de variación), elija Boolean (Booleano), Long (Largo), Double (Doble) o String (Cadena). Para obtener más información, consulte Tipos de variaciones.

c. Agregue un máximo de cinco variaciones para la característica. El Value (Valor) para cada variación debe ser válido para el Variation type (Tipo de variación) que haya seleccionado.

   Seleccione una de las variaciones como predeterminada. Este es el punto de partida con el que se compararán las demás variaciones y debería ser la variación que se está presentando ante los usuarios ahora. Si detiene un experimento, esta variación predeterminada se enviará a todos los usuarios.

d. Elija Sample code (Código de muestra). En el ejemplo de código, observará lo que necesita agregar a la aplicación para configurar las variaciones y asignarles sesiones de usuario. Puede elegir entre JavaScript, Java y Python para el código.

   No es necesario agregar el código a la aplicación en este momento, pero debe hacerlo antes de iniciar el lanzamiento.

   Para obtener más información, consulte Agregue código a la aplicación.

6. En Launch configuration (Configuración de lanzamiento), elija si desea iniciar el lanzamiento inmediatamente o programarlo para que comience más tarde.

7. (Opcional) Para especificar diferentes divisiones de tráfico para los segmentos de audiencia que haya definido, en lugar de la división de tráfico que usará para su público general, seleccione Add Segment Overrides (Agregar anulaciones de segmentos).

   En Segment Overrides (Anulación de segmento), seleccione un segmento y defina la división de tráfico que se utilizará para ese segmento.

   Si lo desea, puede definir más segmentos para definir divisiones de tráfico eligiendo Add Segment Override (Agregar anulación de segmento). Un lanzamiento puede tener hasta seis anulaciones de segmentos.

   Para obtener más información, consulte Use segmentos para centrar su audiencia.

8. Para Traffic configuration (Configuración de tráfico), seleccione el porcentaje de tráfico que desea asignar a cada variación del público general que no coincida con las anulaciones de segmentos. También puede elegir variaciones para que no se envíen a los usuarios.

   En el Traffic summary (Resumen del tráfico) se muestra cuánto tráfico general está disponible para este lanzamiento.

9. Si decide programar el lanzamiento para que comience más tarde, puede agregar varios pasos al lanzamiento. En cada paso, se pueden utilizar porcentajes diferentes para las variaciones. Para ello, elija Add another step (Agregar otro paso) y luego, especifique la programación y los porcentajes de tráfico para el siguiente paso. Puede incluir hasta cinco pasos en un lanzamiento.
10. Si desea hacer un seguimiento del rendimiento de sus características con métricas durante el lanzamiento, elija Metrics (Métricas), Add metric (Agregar métrica). Puede utilizar métricas CloudWatch RUM o métricas personalizadas.

    Para utilizar una métrica personalizada, puede crear la métrica aquí mediante una regla de Amazon EventBridge. Para crear una métrica personalizada, realice lo siguiente:

    - Elija Custom metrics (Métricas personalizadas) e ingrese un nombre para la métrica.
    - En Metric rule (Regla métrica), para Entity ID (ID de la entidad), ingrese la forma de identificar la entidad. Puede ser un usuario o una sesión que realiza una acción que ocasiona que se registre un valor de métrica. Un ejemplo es `userDetails.userID`.
    - Para Value key (Clave de valor), ingrese el valor del que se va a realizar el seguimiento para producir la métrica.
    - Puede optar por ingresar un nombre para las unidades de la métrica. El nombre de esta unidad es solo para fines de visualización, para su uso en gráficos en la consola de Evidently.

    Al ingresar esos campos, en el cuadro se mostrarán ejemplos de cómo codificar la regla de EventBridge para crear la métrica. Para obtener más información sobre EventBridge, consulte ¿Qué es Amazon EventBridge?

    Para utilizar las métricas de RUM, debe tener ya configurado un monitor de aplicaciones RUM para su aplicación. Para obtener más información, consulte Configuración de una aplicación para utilizar CloudWatch RUM.

    > **Note**
    >
    > Si utiliza métricas de RUM y el monitor de aplicaciones no está configurado para muestrear el 100 % de las sesiones de usuario, no todas las sesiones de usuario que participan en el lanzamiento enviarán métricas a Evidently. Para garantizar que las métricas de lanzamiento sean precisas, recomendamos que el monitor de aplicaciones utilice el 100 % de las sesiones de usuario para el muestreo.

11. (Opcional) Si crea al menos una métrica para el lanzamiento, puede asociar una alarma de CloudWatch existente a este lanzamiento. Para ello, elija Associate CloudWatch alarms (Asociar alarmas de CloudWatch).

    Cuando asocia una alarma con un lanzamiento, CloudWatch Evidently debe agregar etiquetas a la alarma con el nombre del proyecto y el nombre del lanzamiento. Esto es para que CloudWatch Evidently pueda mostrar las alarmas correctas en la información de lanzamiento de la consola.

    Para confirmar que CloudWatch Evidently agregará estas etiquetas, elija Allow Evidently to tag the alarm resource identified below with this launch resource (Permitir que Evidently etiquete el recurso de alarma que se identifica a continuación con este recurso de lanzamiento). Luego, elija Associate alarm (Asociar alarma) e ingrese el nombre de la alarma.

    Para obtener más información sobre cómo crear alarmas de CloudWatch, consulte Uso de las alarmas de Amazon CloudWatch.
12. (Opcional) Para agregar etiquetas a este lanzamiento, elija Tags (Etiquetas), Add new tag (Agregar nueva etiqueta).

    Luego, en Key (Clave), ingrese un nombre para la etiqueta. Puede agregar un valor opcional para la etiqueta en Valor.

    Para agregar otra etiqueta, vuelva a elegir Add new tag (Agregar nueva etiqueta).

    Para obtener más información, consulte Tagging AWS Resources (Etiquetado de recursos de).
13. Elija Create launch (Crear lanzamiento).

Creación de un experimento

Utilice experimentos para probar diferentes versiones de una característica o sitio web y recopilar datos de sesiones de usuarios reales. De esta forma, puede tomar decisiones para su aplicación basadas en pruebas y datos.

Para poder agregar un experimento, debe haber creado un proyecto. Para obtener más información, consulte Crear un nuevo proyecto de .

Al agregar un experimento, puede utilizar una característica que ya ha creado o crear una nueva característica mientras crea el experimento.

Para agregar un experimento a un proyecto

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Señales de aplicación, Evidently.
3. Seleccione el botón situado junto al nombre del proyecto y elija Project actions (Acciones del proyecto), Create experiment (Crear experimento).
4. Para Experiment name (Nombre del experimento), ingrese un nombre que se utilizará para identificar esta característica dentro de este proyecto.

   También puede optar por agregar una descripción.
5. Elija Select from existing features (Seleccionar una de las características existentes) o Add new feature (Agregar nueva característica).

   Si está utilizando una característica existente, selecciónela en Feature name (Nombre de la característica).

   Si elige Add new feature (Agregar nueva característica), realice lo siguiente:

   a. En Feature name (Nombre de la característica), ingrese un nombre que se utilizará para identificar esta característica dentro de este proyecto. También puede optar por ingresar una descripción.

   b. En Feature variations (Variaciones de características), para Variation type (Tipo de variación), elija Boolean (Booleano), Long (Largo), Double (Doble) o String (Cadena). El tipo define el tipo de valor que se utiliza para cada variación. Para obtener más información, consulte Tipos de variaciones.

   c. Agregue un máximo de cinco variaciones para la característica. El Value (Valor) para cada variación debe ser válido para el Variation type (Tipo de variación) que haya seleccionado.

      Seleccione una de las variaciones como predeterminada. Este es el punto de partida con el que se compararán las demás variaciones y debería ser la variación que se está presentando ante los usuarios ahora. Si detiene un experimento que utiliza esta característica, se otorgará la variación predeterminada al porcentaje de usuarios que estaban en el experimento anteriormente.

   d. Elija Sample code (Código de muestra). En el ejemplo de código, observará lo que necesita agregar a la aplicación para configurar las variaciones y asignarles sesiones de usuario. Puede elegir entre JavaScript, Java y Python para el código.

No es necesario agregar el código a su aplicación en este momento, pero debe hacerlo antes de iniciar el experimento. Para obtener más información, consulte Agregue código a la aplicación.

6. Para Audience (Audiencia), si lo desea, seleccione un segmento que haya creado si desea que este experimento se aplique solo a los usuarios que coincidan con ese segmento. Para obtener más información acerca de los segmentos, consulte Use segmentos para centrar su audiencia.
7. Para Traffic split for the experiment (División del tráfico para el experimento), especifique el porcentaje del público seleccionado cuyas sesiones se utilizarán en el experimento. Luego, asigne el tráfico para las diferentes variaciones que se utilizan en el experimento.

   Si un lanzamiento y un experimento se están ejecutando al mismo tiempo para la misma característica, el público primero se dirige al lanzamiento. Luego, el porcentaje de tráfico especificado para el lanzamiento se toma de la audiencia general. Después, el porcentaje que especifica en esta etapa es el porcentaje de la audiencia restante que se utiliza para el experimento. Luego, el tráfico restante utiliza la variación predeterminada.
8. Para Metrics (Métricas), elija las métricas que desea utilizar para evaluar las variaciones durante el experimento. Debe utilizar al menos una métrica para efectuar una evaluación.

   a. Para Metric source (Origen de métricas), elija si desea utilizar métricas de CloudWatch RUM o métricas personalizadas.

   b. Ingrese un nombre para la métrica. Para Goal (Meta), elija Increase (Aumentar) si desea un valor superior para que la métrica indique una variación mejor. Elija Decrease (Disminuir) si desea un valor inferior para que la métrica indique una variación mejor.

   c. Si utiliza una métrica personalizada, puede crear la métrica aquí mediante una regla de Amazon EventBridge. Para crear una métrica personalizada, realice lo siguiente:

      - En Metric rule (Regla métrica), para Entity ID (ID de la entidad), ingrese una forma de identificar la entidad. Puede ser un usuario o sesión que realiza una acción que ocasiona que se registre un valor de métrica. Un ejemplo es `userDetails.userID`.
      - Para Value key (Clave de valor), ingrese el valor del que se va a realizar el seguimiento para producir la métrica.
      - Puede optar por ingresar un nombre para las unidades de la métrica. El nombre de esta unidad es solo para fines de visualización, para su uso en gráficos en la consola de Evidently.

Puede utilizar métricas de RUM solo si ha configurado RUM para monitorear esta aplicación. Para obtener más información, consulte Uso de CloudWatch RUM.

> **Note**
>
> Si utiliza métricas de RUM y el monitoreo de aplicaciones no está configurado para muestrear el 100 % de las sesiones de usuario, no todas las sesiones de usuario del experimento enviarán métricas a Evidently. Para garantizar que las métricas del experimento sean precisas, recomendamos que en el monitoreo de aplicaciones se utilice el 100 % de las sesiones de usuario para el muestreo.

d. (Opcional) Para agregar más métricas para evaluar, elija Add metric (Agregar métrica). Puede evaluar un máximo de tres métricas durante el experimento.

9. (Opcional) Si desea crear alarmas de CloudWatch para utilizarlas con este experimento, elija CloudWatch alarms (Alarmas de CloudWatch). Las alarmas pueden monitorear si la diferencia de resultados entre cada variación y la variación predeterminada es mayor que un umbral especificado. Si el rendimiento de una variación es peor que la variación predeterminada y la diferencia es mayor que su umbral, entrará en estado de alarma y usted recibirá una notificación.

   Si crea una alarma en esta etapa, se creará una alarma para cada variación que no sea la variación predeterminada.

   Si crea una alarma, especifique lo siguiente:

   - Para Metric name (Nombre de métrica), elija la métrica del experimento que va a utilizar para la alarma.
   - Para Alarm condition (Condición de alarma), elija qué condición ocasiona que la alarma entre en estado de alarma, cuando los valores de métrica de variación se comparan con los valores de métrica de variación predeterminados. Por ejemplo, elija Greater (Mayor) o Greater/Equal (Mayor/Igual) si los números más altos de la variación indican que está funcionando mal. Esto sería adecuado, por ejemplo, si la métrica mide el tiempo de carga de la página.
   - Ingrese un número para el umbral, que es la diferencia porcentual en el rendimiento que ocasionará que la alarma entre en estado de `ALARM` (alarma).
   - Para Average over period (Promedio durante el periodo), elija cuántos datos métricos de cada variación se agregan juntos antes de compararlos.

Puede elegir Add new alarm (Agregar nueva alarma) otra vez para agregar más alarmas al experimento.

Luego, elija Set notifications for the alarm (Establecer notificaciones para la alarma) y seleccione o cree un tema de Amazon Simple Notification Service para enviarle notificaciones de alarma. Para obtener más información, consulte Configuración de notificaciones de Amazon SNS.

10. (Opcional) Para agregar etiquetas a este experimento, elija Tags (Etiquetas), Add new tag (Agregar nueva etiqueta).

    Luego, en Key (Clave), ingrese un nombre para la etiqueta. Puede agregar un valor opcional para la etiqueta en Valor.

    Para agregar otra etiqueta, vuelva a elegir Add new tag (Agregar nueva etiqueta).

    Para obtener más información, consulte Tagging AWS Resources (Etiquetado de recursos de).
11. Elija Create experiment (Crear experimento).
12. Si aún no lo ha hecho, cree las variaciones de la característica en la aplicación.
13. Seleccione Listo. El experimento no iniciará hasta que lo inicie.

Después de completar los pasos del siguiente procedimiento, el experimento iniciará de forma inmediata.

Para iniciar un experimento que ha creado

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Señales de aplicación, Evidently.
3. Elija el nombre del proyecto.
4. Elija la pestaña Experiments (Experimentos).
5. Elija el botón situado junto al nombre del experimento y Actions (Acciones), Start experiment (Iniciar experimento).
6. (Opcional) Para ver o modificar la configuración del experimento que estableció cuando lo creó, elija Experiment setup (Configuración del experimento).
7. Elija el horario de finalización del experimento.
8. Elija Start experiment (Iniciar experimento).

El experimento iniciará de forma inmediata.

Administración de características, lanzamientos y experimentos

Utilice los procedimientos detallados en estas secciones para administrar las características, los lanzamientos y los experimentos que ha creado.

Temas

Consulte las reglas de evaluación actuales y el tráfico de audiencia de una característica

Puede utilizar la consola de CloudWatch Evidently para ver cómo las reglas de evaluación de la característica asignan el tráfico de audiencia entre los lanzamientos, experimentos y variaciones actuales de la característica.

Para visualizar el tráfico de audiencia de una característica

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Señales de aplicación, Evidently.
3. Elija el nombre del proyecto que contiene la característica.
4. Elija la pestaña Features (Características).
5. Elija el nombre de la característica.

   En la pestaña Evaluation rules (Reglas de evaluación), puede ver el flujo de tráfico de audiencia de su característica, de la siguiente manera:

- En primer lugar, se evalúan las anulaciones. Estas especifican que siempre se utiliza una variación específica con determinados usuarios. Las sesiones de los usuarios a los que se les asignan anulaciones no contribuyen a las métricas de lanzamiento o experimentación.
- Luego, el tráfico restante se encuentra disponible para el lanzamiento en curso, si corresponde. Si hay un lanzamiento en curso, observará el nombre del lanzamiento y el tráfico de lanzamiento dividido entre las variaciones de las características en la tabla de la sección de Launches (Lanzamientos). En el lado derecho de la sección de Launches (Lanzamientos), un indicador de Traffic (Tráfico) muestra la cantidad de la audiencia disponible (después de las anulaciones) que se asigna a este lanzamiento. El resto del tráfico no asignado al lanzamiento fluye hacia el experimento (si corresponde) y, a continuación, a la variación predeterminada.
- Luego, el tráfico restante estará disponible para el experimento en curso, si corresponde. Si hay un experimento en curso, observará el nombre y el progreso del experimento en la sección Experiments (Experimentos). En el lado derecho de la sección Experiments (Experimentos), un indicador de Traffic (Tráfico) muestra la cantidad de la audiencia disponible (después de las anulaciones y los lanzamientos) que se asigna a este lanzamiento. El resto del tráfico no asignado al lanzamiento o al experimento recibe la variación predeterminada de la característica.

Modificación del tráfico de lanzamiento

Puede modificar la asignación de tráfico de un lanzamiento en cualquier momento, incluso mientras el lanzamiento está en curso.

Si tiene un lanzamiento continuo y un experimento continuo para la misma característica, cualquier cambio que se realice en el tráfico de características ocasionará un cambio en el tráfico del experimento. Esto se debe a que la audiencia disponible para el experimento es la parte de la audiencia total que aún no se asignó al lanzamiento. El aumento del tráfico de lanzamiento disminuirá la audiencia disponible para el experimento y la disminución del tráfico de lanzamiento o la finalización del lanzamiento aumentarán la audiencia disponible para el experimento.

Para modificar la asignación de tráfico de un lanzamiento

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Señales de aplicación, Evidently.
3. Elija el nombre del proyecto que contiene el lanzamiento.
4. Elija la pestaña Launches (Lanzamientos).

5. Elija el nombre del lanzamiento.

   Elija Modify launch traffic (Modificar el tráfico de lanzamiento).

6. Para Serve (Enviar), seleccione el nuevo porcentaje de tráfico que desea asignar a cada variación. También puede elegir variaciones para que no se envíen a los usuarios. A medida que cambie estos valores, podrá ver los efectos actualizados en el tráfico general de características enTraffic summary (Resumen del tráfico).

   El Traffic summary muestra cuánto tráfico general está disponible para este lanzamiento y cuánto de ese tráfico disponible se asigna a este lanzamiento.

7. Elija Modificar.

Modificación de los pasos futuros de un lanzamiento

Puede modificar la configuración de los pasos de lanzamiento que aún no se hayan producido y agregar más pasos a un lanzamiento.

Para modificar los pasos de un lanzamiento

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Señales de aplicación, Evidently.
3. Elija el nombre del proyecto que contiene el lanzamiento.
4. Elija la pestaña Launches (Lanzamientos).
5. Elija el nombre del lanzamiento.

   Elija Modify launch traffic (Modificar el tráfico de lanzamiento).

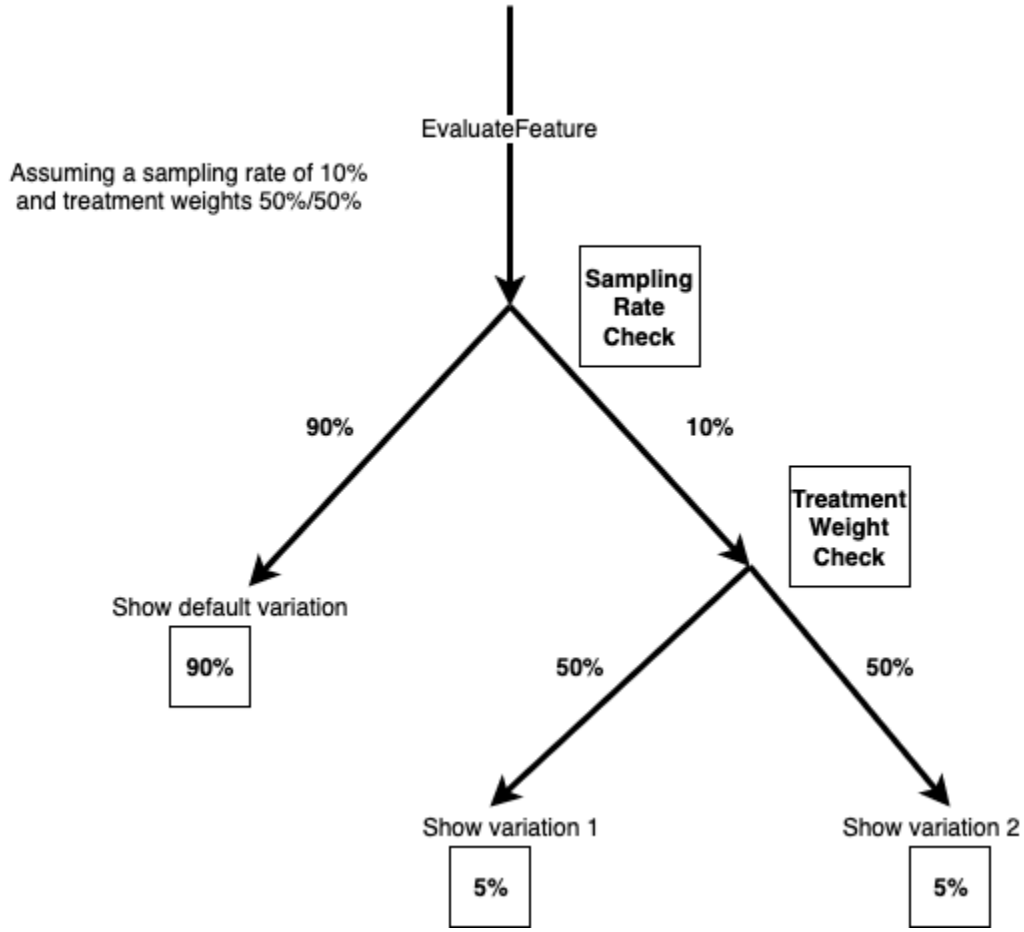
6. Elija Schedule launch (Programar lanzamiento).
7. Para cualquier paso que aún no se haya iniciado, puede modificar el porcentaje de la audiencia disponible que se va a utilizar en el experimento. También puede modificar la forma en la que se asigna su tráfico entre las variaciones.

   Puede agregar más pasos al lanzamiento al seleccionar Add another step (Agregar otro paso). Un lanzamiento puede tener un máximo de cinco pasos.

8. Elija Modificar.

Modificación del tráfico de experimentos

Puede modificar el tráfico de muestra de un experimento en cualquier momento, incluso mientras el experimento está en curso. Sin embargo, no puede actualizar los pesos del tratamiento después de ejecutar un experimento. Por lo tanto, puede cambiar el tráfico total expuesto al experimento después de ejecutarlo, pero no la asignación relativa a cada tratamiento. Si modifica el tráfico de un experimento en curso, le recomendamos que solo aumente la asignación de tráfico, para evitar sesgos.



Para modificar la asignación de tráfico de un experimento

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Application monitoring (Monitoreo de aplicaciones), Evidently.
3. Elija el nombre del proyecto que contiene el lanzamiento.
4. Elija la pestaña Experiments (Experimentos).
5. Elija el nombre del lanzamiento.

6. Elija Modify experiment traffic (Modificar el tráfico de experimentos).
7. Ingrese un porcentaje o utilice el control deslizante para especificar cuánto tráfico disponible se va a asignar a este experimento. El tráfico disponible es la audiencia total menos el tráfico que se asigna a un lanzamiento actual, si corresponde. El tráfico que no se asigna al lanzamiento o al experimento recibe la variación predeterminada.
8. Elija Modificar.

Detener un lanzamiento

Si detiene un lanzamiento en curso, no podrá reanudarlo ni reiniciarlo. Además, no se evaluará como regla para la asignación de tráfico y el tráfico que se asignó al lanzamiento estará disponible para el experimento de la característica, si corresponde. De lo contrario, todo el tráfico se utilizará como variación predeterminada una vez que se detenga el lanzamiento.

Para detener un lanzamiento de forma permanente

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Señales de aplicación, Evidently.
3. Elija el nombre del proyecto que contiene el lanzamiento.
4. Elija la pestaña Launches (Lanzamientos).
5. Elija el botón situado a la izquierda del nombre del lanzamiento.
6. Elija Actions (Acciones), Cancel launch (Cancelar lanzamiento) o Actions (Acciones), Mark as complete (Marcar como completado).

Detener un experimento

Si detiene un experimento en curso, no podrá reanudarlo ni reiniciarlo. La parte del tráfico que se haya utilizado previamente en el experimento recibirá la variación predeterminada.

Cuando un experimento no se detiene de forma manual y excede su fecha de finalización, el tráfico no cambia. La parte del tráfico asignada al experimento sigue destinada al experimento. Para detener esto y ocasionar que el tráfico del experimento se destine, en su lugar, a la variación predeterminada, marque el experimento como completado.

Cuando detiene un experimento, puede elegir cancelarlo o marcarlo como completado. Si cancela, se mostrará como Cancelled (Cancelado) en la lista de experimentos. Si decide marcarlo como completado, se mostrará como Completed (Completado).

Para detener un experimento de forma permanente

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Señales de aplicación, Evidently.
3. Elija el nombre del proyecto que contiene el experimento.
4. Elija la pestaña Experiments (Experimentos).
5. Elija el botón situado a la izquierda del nombre del experimento.
6. Elija Actions (Acciones), Cancel experiment (Cancelar experimento) o Actions (Acciones), Mark as complete (Marcar como completado).

Agregue código a la aplicación

Si desea trabajar con CloudWatch Evidently, agregue código a la aplicación para asignar una variación a cada sesión de usuario y para enviar métricas a Evidently. Utilice la operación `EvaluateFeature` de CloudWatch Evidently para asignar variaciones a las sesiones de usuario y utilice la operación `PutProjectEvents` para enviar eventos a Evidently y utilizarlos para calcular las métricas de sus lanzamientos o experimentos.

Al crear variaciones o métricas personalizadas, la consola de CloudWatch Evidently proporcionará ejemplos del código que necesita agregar.

Si desea obtener un ejemplo integral, consulte Tutorial: A/B testing with the Evidently sample application (Pruebas A/B con la aplicación de muestra de Evidently).

Utilizar EvaluateFeature

Cuando se utilizan variaciones de características en un lanzamiento o experimento, la aplicación utiliza la operación EvaluateFeature para asignar una variación a cada sesión de usuario. La asignación de una variación a un usuario es un evaluation event (evento de evaluación). Cuando utiliza esta operación, sucede lo siguiente:

- Feature name (Nombre de la característica): obligatorio. Evidently procesa la evaluación de acuerdo con las reglas de evaluación de características del lanzamiento o experimento y selecciona una variación para la entidad.
- entityId (ID de la entidad): obligatorio. Representa a un usuario único.
- evaluationContext (Contexto de evaluación): opcional. Un objeto JSON que representa información adicional sobre un usuario. Evidentemente usará este valor para hacer coincidir al usuario con un

segmento de su audiencia durante las evaluaciones de funciones, si ha creado segmentos. Para obtener más información, consulte Use segmentos para centrar su audiencia.

A continuación, se muestra un ejemplo de un valor `evaluationContext` que puede enviar a Evidently.

```
{
    "Browser": "Chrome",
    "Location": {
        "Country": "United States",
        "Zipcode": 98007
    }
}
```

Evaluaciones sticky

CloudWatch Evidently utiliza evaluaciones “sticky”. Una configuración única de `entityId`, característica, configuración de característica y `evaluationContext` siempre recibe la misma asignación de variantes. El único cambio de esta asignación de variaciones se produce cuando se agrega una entidad a una anulación o se marca el tráfico del experimento.

Una configuración de características incluye lo siguiente:

- Las variaciones de características
- La configuración de variación (porcentajes asignados a cada variación) para un experimento en ejecución actualmente de esta característica, si existe.
- La configuración de variantes para un lanzamiento en ejecución actualmente para esta característica, si existe. La configuración de variantes incluye las anulaciones de segmentos definidas, si las hay.

Si se aumenta la asignación de tráfico de un experimento, los `entityId` que se hayan asignado previamente a un grupo de tratamiento experimental seguirán recibiendo el mismo tratamiento. Cualquier `entityId` que se haya asignado previamente al grupo de control podría asignarse a un grupo de tratamiento del experimento, de acuerdo con la configuración de variación especificada para el experimento.

Si se reduce la asignación de tráfico de un experimento, `entityId` puede pasar de un grupo de tratamiento a un grupo de control, pero no pasará a un grupo de tratamiento diferente.

Usar PutProjectEvents

Para codificar una métrica personalizada para Evidently, utilice la operación PutProjectEvents. A continuación, observará un ejemplo de carga.

```
{
    "events": [
        {
            "timestamp": {{$timestamp}},
            "type": "aws.evidently.custom",
            "data": "{\"details\": {\"pageLoadTime\": 800.0}, \"userDetails\":
 {\"userId\": \"test-user\"}}"
        }
    ]
}
```

La `entityIdKey` (clave de ID de identidad) puede ser simplemente un `entityId` (ID de identidad) o puede cambiarle el nombre a cualquier otra cosa, como `userId` (ID del usuario). En un caso real, un `entityId` puede ser un nombre de usuario, un ID de sesión, etc.

```
"metricDefinition":{
    "name": "noFilter",
    "entityIdKey": "userDetails.userId", //should be consistent with jsonValue in
 events "data" fields
    "valueKey": "details.pageLoadTime"
},
```

Para asegurarse de que los eventos estén asociados con el lanzamiento o experimento correctos, debe pasar el mismo `entityId` cuando llame tanto a `EvaluateFeature` como a `PutProjectEvents`. Asegúrese de llamar a `PutProjectEvents` tras la llamada a `EvaluateFeature`; de lo contrario, los datos se eliminan y CloudWatch Evidently no los utilizará.

La operación `PutProjectEvents` no requiere el nombre de la característica como parámetro de entrada. De esta forma, puede utilizar un único evento en múltiples experimentos. Por ejemplo, supongamos que llama `EvaluateFeature` con `entityId` establecido en `userDetails.userId`. Si tiene dos o más experimentos en ejecución, puede hacer que un único evento de la sesión de ese usuario emita métricas para cada uno de esos experimentos. Para ello, llama a `PutProjectEvents` una vez por cada experimento, utilizando ese mismo `entityId`.

Timing

Después de que su aplicación llame a `EvaluateFeature`, hay un periodo de tiempo de una hora en el que los eventos métricos de `PutProjectEvents` se atribuyen según esa evaluación. Si se producen más eventos después del período de una hora, no se les atribuye.

Sin embargo, si el mismo `entityId` se utiliza para una nueva llamada de `EvaluateFeature` durante el período de una hora de esa llamada inicial, se utilizará el último resultado de `EvaluateFeature` en su lugar y se reiniciará el temporizador de una hora. Esto solo puede ocurrir en determinadas circunstancias, como cuando el tráfico de experimentos se marca entre las dos asignaciones, tal y como se explica en la sección anterior, Evaluaciones sticky.

Si desea obtener un ejemplo integral, consulte Tutorial: A/B testing with the Evidently sample application (Pruebas A/B con la aplicación de muestra de Evidently).

Almacenamiento de datos del proyecto

Evidently recopila dos tipos de eventos:

- Los eventos de evaluación están relacionados con la variación de características asignada a una sesión de usuario. Evidently utiliza estos eventos para producir métricas y otros datos de experimento y lanzamiento, que se pueden ver en la consola de Evidently.

  También puede elegir almacenar estos eventos de evaluación en Amazon CloudWatch Logs o Amazon S3.
- Los eventos personalizados se utilizan para generar métricas a partir de acciones del usuario, tales como clics y salidas. Evidently no incluye un método para almacenar eventos personalizados. Si desea guardarlos, debe modificar el código de la aplicación para enviarlo a una opción de almacenamiento externo a Evidently.

Formato de los registros de eventos de evaluación

Si elige almacenar eventos de evaluación en CloudWatch Logs o Amazon S3, cada evento de evaluación se almacena como evento de registro con el siguiente formato:

```
{
    "event_timestamp": 1642624900215,
    "event_type": "evaluation",
    "version": "1.0.0",
    "project_arn": "arn:aws:evidently:us-east-1:123456789012:project/petfood",
    "feature": "petfood-upsell-text",
```

```
    "variation": "Variation1",
    "entity_id": "7",
    "entity_attributes": {},
    "evaluation_type": "EXPERIMENT_RULE_MATCH",
    "treatment": "Variation1",
    "experiment": "petfood-experiment-2"
}
```

A continuación se muestran más detalles sobre el formato de evento de evaluación anterior:

- La marca de hora está en tiempo UNIX con milisegundos
- La variación es el nombre de la variación de la función asignada a esta sesión de usuario.
- El ID de entidad es una cadena.
- Los atributos de entidad son un hash de valores arbitrarios enviados por el cliente. Por ejemplo, si el `entityId` se asigna a azul o verde, entonces, de forma opcional, puede enviar ID de usuario, datos de sesión o cualquier otra cosa que desee desde una perspectiva de correlación y almacén de datos.

Política de IAM y cifrado para el almacenamiento de eventos de evaluación en Amazon S3

Si elige utilizar Amazon S3 para almacenar los eventos de evaluación, debe agregar una política de IAM como la siguiente para permitir que Evidently publique registros en el bucket de Amazon S3. Esto se debe a que los buckets de Amazon S3 y los objetos que contienen son privados y no permiten acceder a otros servicios de forma predeterminada.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AWSLogDeliveryWrite",
            "Effect": "Allow",
            "Principal": {"Service": "delivery.logs.amazonaws.com"},
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::bucket_name/optional_folder/AWSLogs/account_id/
*",
            "Condition": {"StringEquals": {"s3:x-amz-acl": "bucket-owner-full-
control"}}
        },
        {
```

```
            "Sid": "AWSLogDeliveryCheck",
            "Effect": "Allow",
            "Principal": {"Service": "delivery.logs.amazonaws.com"},
            "Action": ["s3:GetBucketAcl", "s3:ListBucket"],
            "Resource": "arn:aws:s3:::bucket_name"
        }
    ]
}
```

Si almacena datos de Evidently en Amazon S3, también puede elegir cifrarlos con cifrado del lado del servidor con claves de AWS Key Management Service (SSE-KMS). Para obtener más información, consulte Protección de los datos con el cifrado del lado del servidor.

Si utiliza una clave administrada por el cliente desde AWS KMS, debe agregar lo siguiente a la política de IAM para su clave. Esto permite que Evidently escriba en el bucket.

```
{
    "Sid": "AllowEvidentlyToUseCustomerManagedKey",
    "Effect": "Allow",
    "Principal": {
        "Service": [
            "delivery.logs.amazonaws.com"
        ]
    },
   "Action": [
       "kms:Encrypt",
       "kms:Decrypt",
       "kms:ReEncrypt*",
       "kms:GenerateDataKey*",
       "kms:DescribeKey"
    ],
    "Resource": "*"
}
```

Cómo calcula Evidently los resultados

Puede utilizar las pruebas A/B de Amazon CloudWatch Evidently como herramienta para la toma de decisiones basada en datos. En una prueba A/B, los usuarios se asignan aleatoriamente al grupo de control (también denominado variación predeterminada) o a uno de los grupos de tratamiento (también denominados variaciones analizadas). Por ejemplo, los usuarios del grupo de control pueden experimentar el sitio web, el servicio o la aplicación de la misma manera que antes de que

comenzara el experimento. Mientras tanto, es posible que los usuarios del grupo de tratamiento experimenten el cambio.

CloudWatch Evidently admite hasta cinco variaciones diferentes en un experimento. Evidently asigna tráfico de forma aleatoria a estas variaciones. De esta forma, puede hacer un seguimiento de las métricas empresariales (como los ingresos) y las métricas de rendimiento (como la latencia) de cada grupo. Evidently hace lo siguiente:

- Compara el tratamiento con el control. (Por ejemplo, compara si los ingresos aumentan o disminuyen con un nuevo proceso de pago).
- Indica si la diferencia observada entre el tratamiento y el control es significativa. Para ello, Evidently ofrece dos enfoques: niveles de importancia frecuentistas y probabilidades bayesianas.

¿Por qué usar enfoques frecuentistas y bayesianos?

Piense en un caso en el que el tratamiento no tenga efecto en comparación con el control o un caso en el que el tratamiento sea idéntico al control (una prueba A/A). Aun así, observaría una pequeña diferencia entre el tratamiento y el control en los datos. Esto se debe a que los participantes de la prueba consisten en una muestra finita de usuarios, que representa un pequeño porcentaje de todos los usuarios del sitio web, el servicio o la aplicación. Los niveles de importancia frecuentistas y las probabilidades bayesianas proporcionan información sobre si la diferencia observada es significativa o se debe al azar.

Evidently considera lo siguiente para determinar si la diferencia observada es significativa:

- El tamaño de la diferencia
- El número de muestras que forman parte de la prueba
- La forma de distribución de los datos

Análisis frecuentista en Evidently

Evidently utiliza pruebas secuenciales, lo que evita los problemas habituales de los vistazos, un error común de las estadísticas frecuentistas. Los vistazos son la práctica de comprobar los resultados de una prueba A/B en curso para detenerla y tomar una decisión basada en los resultados observados. Para obtener más información sobre las pruebas secuenciales, consulte Time-uniform, nonparametric, nonasymptotic confidence sequences (Secuencias de confianza no asintóticas, no paramétricas y uniformes en el tiempo) de Howard et al. (Ann. Statist. 49 (2) 1055 - 1080, 2021).

Dado que los resultados de Evidently son válidos en cualquier momento (resultados válidos en cualquier momento), puede echar un vistazo a los resultados durante el experimento y aun así sacar conclusiones sólidas. Esto puede reducir algunos de los costos de la experimentación, ya que puede detener un experimento antes de la hora programada si los resultados ya son significativos.

Evidently genera niveles de importancia válidos en cualquier momento e intervalos de confianza del 95 % válidos en cualquier momento de la diferencia entre la variación probada y la variación por defecto en la métrica objetivo. La columna Result (Resultado) del experimento indica el rendimiento de la variación probada, que puede ser uno de los siguientes:

- Inconclusive (No concluyente): el nivel de importancia es inferior al 95 %.
- Better (Mejor): el nivel de importancia es del 95 % o superior y se cumple una de las condiciones siguientes:
  - El límite inferior del intervalo de confianza del 95 % es superior a cero y la métrica debería aumentar.
  - El límite superior del intervalo de confianza del 95 % es inferior a cero y la métrica debería disminuir.
- Worse (Peor): el nivel de importancia es del 95 % o superior y se cumple una de las condiciones siguientes:
  - El límite superior del intervalo de confianza del 95 % es superior a cero y la métrica debería aumentar.
  - El límite inferior del intervalo de confianza del 95 % es inferior a cero y la métrica debería disminuir.
- Best (Óptimo): el experimento tiene dos o más variaciones probadas además de la variación predeterminada y se cumplen las siguientes condiciones:
  - La variación cumple las condiciones para la designación Better (Mejor)
  - Se cumple una de las siguientes condiciones:
    - El límite inferior del intervalo de confianza del 95 % es mayor que el límite superior de los intervalos de confianza del 95 % de todas las demás variaciones y la métrica debería aumentar
    - El límite superior del intervalo de confianza del 95 % es menor que el límite inferior de los intervalos de confianza del 95 % de todas las demás variaciones y la métrica debe disminuir

Análisis bayesiano en Evidently

Con el análisis bayesiano, puede calcular la probabilidad de que la media de la variación probada sea mayor o menor que la media de la variación predeterminada. Evidently lleva a cabo la inferencia bayesiana para la media de la métrica objetivo mediante distribuciones a priori de conjugados. Con las distribuciones a priori de conjugados, Evidently puede inferir de manera más eficiente la distribución posterior necesaria para el análisis bayesiano.

Evidently espera hasta la fecha de finalización del experimento para calcular los resultados del análisis bayesiano. En la página de resultados se muestra la siguiente información:

- probability of increase (probabilidad de aumento): la probabilidad de que la media de la métrica en la variación probada sea al menos un 3 % mayor que la media de la variación predeterminada
- probability of decrease (probabilidad de disminución): la probabilidad de que la media de la métrica en la variación probada sea al menos un 3 % menor que la media de la variación predeterminada
- probability of no change (probabilidad de que no haya ningún cambio): la probabilidad de que la media de la métrica en la variación probada esté en un ±3 % de la media de la variación predeterminada

La columna Result (Resultado) indica el rendimiento de la variación probada y puede ser uno de los siguientes:

- Better (Mejor): la probabilidad de aumento es de al menos el 90 % y la métrica debería aumentar, o la probabilidad de disminución es de al menos el 90 % y la métrica debería disminuir
- Worse (Peor): la probabilidad de disminución es de al menos el 90 % y la métrica debe aumentar, o la probabilidad de aumento es de al menos el 90 % y la métrica debe disminuir

Visualización de los resultados del lanzamiento en el panel

Puede ver el progreso y los resultados de las métricas de un experimento mientras esté en curso y, también, una vez que haya finalizado.

Para ver el progreso y los resultados de un lanzamiento

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Señales de aplicación, Evidently.
3. Elija el nombre del proyecto que contiene el lanzamiento.

4. Elija la pestaña Launches (Lanzamientos).
5. Elija el nombre del lanzamiento.
6. Para ver los pasos de lanzamiento y las asignaciones de tráfico de cada paso, elija la pestaña Launch (Lanzamiento).
7. Para ver la cantidad de sesiones de usuario asignadas a cada variación a lo largo del tiempo y para ver las métricas de rendimiento de cada variación del lanzamiento, elija la pestaña Monitoring (Monitoreo).

   En esta vista, también se observa si las alarmas de lanzamiento se pusieron en estado de `ALARM` (alarma) durante el lanzamiento.
8. Para ver las variaciones, métricas, alarmas y etiquetas de este lanzamiento, elija la pestaña Configuration (Configuración).

Ver los resultados del experimento en el panel

Puede ver los resultados estadísticos de un experimento mientras esté en curso y, también, una vez que haya finalizado. Los resultados del experimento estarán disponibles hasta 63 días después del inicio del experimento. No estarán disponibles después de ese periodo por lo establecido en las políticas de retención de datos de CloudWatch.

No se mostrarán resultados estadísticos hasta que cada variación tenga al menos 100 eventos.

Evidently, realiza un análisis de valor p fuera de línea adicional al final del experimento. El análisis de valores p fuera de línea puede detectar significación estadística en algunos casos en los que los valores p utilizados en cualquier momento durante el experimento no encuentran significación estadística.

Para obtener más información acerca de cómo CloudWatch Evidently calcula los resultados de los experimentos, consulte Cómo calcula Evidently los resultados.

Para ver los resultados de un experimento

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Señales de aplicación, Evidently.
3. Elija el nombre del proyecto que contiene el experimento.
4. Elija la pestaña Experiments (Experimentos).
5. Elija el nombre del experimento y luego, la pestaña Results (Resultados).

6. En Variation performance (Rendimiento de variación), hay un control en el que puede seleccionar las estadísticas del experimento que desea que se muestren. Si selecciona más de una estadística, Evidently mostrará un gráfico y una tabla para cada una.

   En cada gráfico y tabla, se muestran los resultados del experimento hasta el momento.

   En cada gráfico, se podrán observar los siguientes resultados. Puede utilizar el control situado a la derecha del gráfico para determinar cuál de los siguientes elementos se muestra:

   - La cantidad de eventos de sesión de usuario registrados para cada variación.
   - El valor promedio de la métrica seleccionada en la parte superior del gráfico, para cada variación.
   - La importancia estadística de los experimentos. Esto compara la diferencia de la métrica seleccionada en la parte superior del gráfico con la variación predeterminada y cada una de las demás variaciones.
   - Los límites de confianza superior e inferior del 95% en la diferencia de la métrica seleccionada, entre cada una de las variaciones y la variación predeterminada.

   En la tabla podrá observar una fila para cada variación. Para cada variación que no es la predeterminada, Evidently muestra si ha recibido suficientes datos para declarar que los resultados son estadísticamente significativos. También muestra si la mejora de la variación en el valor estadístico ha alcanzado un nivel de confianza del 95%.

   Por último, en la columna Result (Resultado), Evidently incluye una recomendación sobre la variación que funciona mejor en función de esta estadística o si los resultados no son concluyentes.

Cómo CloudWatch Evidently recopila y almacena datos

Amazon CloudWatch Evidently recopila y almacena datos relacionados con las configuraciones de proyectos para que los clientes puedan ejecutar experimentos y lanzamientos. Los datos incluyen lo siguiente:

- metadatos sobre proyectos, características, lanzamientos y experimentos
- eventos de métricas
- datos de evaluación.

Los metadatos de recursos se almacenan en Amazon DynamoDB. Los datos se cifran en reposo de forma predeterminada mediante Claves propiedad de AWS. Las claves son una colección de claves de AWS KMS que un Servicio de AWS posee y administra para su uso en varias Cuentas de AWS. Los clientes no pueden ver, administrar ni auditar el uso de estas claves. Los clientes tampoco están obligados a tomar medidas ni a cambiar programas para proteger las claves que cifran sus datos.

Para obtener más información, consulte Claves propiedad de AWS en la guía para desarrolladores de AWS Key Management Service.

Los eventos de métricas de Evidently y los eventos de evaluación se entregan directamente en ubicaciones pertenecientes al cliente.

Los datos en tránsito se cifran automáticamente con HTTPS. Estos datos se entregarán a ubicaciones pertenecientes al cliente.

También puede elegir almacenar eventos de evaluación en Amazon Simple Storage Service o Amazon CloudWatch Logs. Para obtener más información acerca de las formas en las que puede proteger sus datos en estos servicios, consulte Habilitación del cifrado predeterminado de bucket de Amazon S3 y Cifrado de datos de registro en CloudWatch Logs mediante AWS KMS.

Recuperación de datos

Puede recuperar sus datos mediante las API de CloudWatch Evidently. Para recuperar datos del proyecto, utilice GetProject (Obtener proyecto) o ListProjects (Enumerar proyectos).

Para recuperar datos de características, utilice GetFeature (Obtener característica) o ListFeatures (Enumerar características).

Para recuperar datos de lanzamiento, utilice GetLaunch (Obtener lanzamiento) o ListLaunches (Enumerar lanzamientos).

Para recuperar datos de experimentos, utilice GetExperiment (Obtener experimento), ListExperiments (Enumerar experimentos) o GetExperimentResults (Obtener resultados de experimentos).

Modificación y eliminación de datos

Puede modificar y eliminar los datos mediante las API de CloudWatch Evidently. Para los datos del proyecto, utilice UpdateProject (Actualizar proyecto) o DeleteProject (Eliminar proyecto).

Para datos de características, utilice UpdateFeature (Actualizar característica) oDeleteFeature (Eliminar característica).

Para obtener datos de lanzamiento, utilice UpdateLaunch (Actualizar lanzamiento) o DeleteLaunch (Eliminar lanzamiento).

Para obtener datos de experimentos, utilice UpdateExperiment (Actualizar experimento) o DeleteExperiment (Eliminar experimento).

Uso de roles vinculados a servicios para Evidently

CloudWatch Evidently utiliza roles vinculados a servicios de AWS Identity and Access Management (IAM). Un rol vinculado a un servicio es un tipo único de rol de IAM que está vinculado directamente a Evidently. Los roles vinculados a servicios están predefinidos por Evidently e incluyen todos los permisos que el servicio requiere para llamar a otros servicios de AWS en su nombre.

Un rol vinculado a un servicio simplifica la configuración de Evidently porque ya no tendrá que agregar manualmente los permisos necesarios. Evidently define los permisos de sus roles vinculados a servicios y, a menos que esté definido de otra manera, solo Evidently puede asumir sus roles. Los permisos definidos incluyen las políticas de confianza y de permisos y que la política de permisos no se pueda adjuntar a ninguna otra entidad de IAM.

Solo puede eliminar un rol vinculado a servicios después de eliminar sus recursos relacionados. De esta forma, se protegen los recursos de Evidently, ya que se evita que se puedan eliminar accidentalmente permisos de acceso a los recursos.

Para obtener información sobre otros servicios que admiten roles vinculados a servicios, consulte Servicios de AWS que funcionan con IAM y busque los servicios que muestran Yes (Sí) en la columna Service-linked roles (Roles vinculados a servicios). Seleccione una opción Sí con un enlace para ver la documentación acerca del rol vinculado al servicio en cuestión.

Permisos de roles vinculados a servicios de Evidently

Evidently usa el rol vinculado al servicio denominado AWSServiceRoleForCloudWatchEvidently: permite que CloudWatch Evidently administre recursos de AWS asociados en nombre del cliente.

El rol vinculado al servicio AWSServiceRoleForCloudWatchEvidently confía en los siguientes servicios para asumir el rol:

- `CloudWatch Evidently`

La política de permisos del rol denominada AmazonCloudWatchEvidentlyServiceRolePolicy permite que Evidently lleve a cabo las siguientes acciones en los recursos especificados:

- Acciones: `appconfig:StartDeployment`, `appconfig:StopDeployment`, `appconfig:ListDeployments` y `appconfig:TagResource` en clientes pesados de Evidently.

Debe configurar permisos para permitir a una entidad de IAM (como un usuario, grupo o rol) crear, editar o eliminar un rol vinculado a servicios. Para obtener más información, consulte Permisos de roles vinculados a servicios en la Guía del usuario de IAM.

Creación de un rol vinculado a un servicio de Evidently

No necesita crear manualmente un rol vinculado a servicios. Cuando empiece a usar un cliente pesado de Evidently en la AWS Management Console, AWS CLI o la API de AWS, Evidently crea el rol vinculado al servicio.

Si elimina este rol vinculado al servicio y necesita crearlo de nuevo, puede utilizar el mismo proceso para volver a crear el rol en su cuenta. Cuando empiece a usar un cliente pesado de Evidently, Evidently vuelve a crear el rol vinculado al servicio.

Modificación de un rol vinculado a un servicio de Evidently

Evidently no le permite modificar el rol vinculado al servicio AWSServiceRoleForCloudWatchEvidently. Después de crear un rol vinculado a servicios, no puede cambiarle el nombre, ya que varias entidades pueden hacer referencia al mismo. Sin embargo, puede editar la descripción del rol mediante IAM. Para obtener más información, consulte Editar un rol vinculado a servicios en la Guía del usuario de IAM.

Eliminación de un rol vinculado a un servicio de Evidently

Si ya no necesita utilizar una característica o servicio que requiere un rol vinculado a un servicio, recomendamos que elimine dicho rol. De esta forma no tiene una entidad no utilizada que no se monitorice ni mantenga de forma activa. Sin embargo, debe limpiar los recursos del rol vinculado al servicio antes de eliminarlo manualmente. Debe eliminar todos los proyectos de Evidently que utilicen clientes pesados.

> **Note**
>
> Si el servicio de Evidently está utilizando el rol cuando intenta eliminar los recursos, la eliminación podría producir un error. En tal caso, espere unos minutos e intente de nuevo la operación.

Para eliminar los recursos de Evidently utilizados por AWSServiceRoleForCloudWatchEvidently

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Application monitoring (Monitoreo de aplicaciones), Evidently.
3. En la lista de proyectos, seleccione la casilla situada junto a los proyectos que usaban clientes pesados.
4. Elija Project actions (Acciones del proyecto), Delete project (Eliminar el proyecto).

Para eliminar manualmente el rol vinculado a servicios mediante IAM

Utilice la consola de IAM, la AWS CLI o la API de AWS para eliminar el rol vinculado al servicio AWSServiceRoleForCloudWatchEvidently. Para obtener más información, consulte Eliminar un rol vinculado a un servicio en la Guía del usuario de IAM.

Regiones admitidas para los roles vinculados a servicios de Evidently

Evidently admite el uso de roles vinculados a servicios en todas las regiones en las que el servicio está disponible. Para obtener más información, consulte Puntos de enlace y regiones de AWS.

Cuotas de CloudWatch Evidently

CloudWatch Evidently posee las siguientes cuotas.

Recurso	Cuota predeterminada
Proyectos	50 por región y por cuenta Puede solicitar un aumento de cuota.
Segmentos	500 por región y por cuenta Puede solicitar un aumento de cuota.
Cuotas por proyecto	• 100 características totales • 500 lanzamientos en total • 50 lanzamientos en curso • 500 experimentos en total • 50 experimentos en curso

Recurso	Cuota predeterminada
	Puede solicitar un aumento de todas estas cuotas.
Cuotas API (todas las cuotas son por región)	• PutProjectEvents: 1000 transacciones por segundo (TPS) en Este de EE. UU. (Norte de Virginia), Oeste de EE. UU. (Oregón) y Europa (Irlanda). En el resto de las regiones, 200 TPS. • EvaluateFeature: 1000 TPS en Este de EE. UU. (Norte de Virginia), Oeste de EE. UU. (Oregón) y Europa (Irlanda). En el resto de las regiones, 200 TPS. • BatchEvaluateFeature: 50 TPS • API de creación, lectura, actualización, eliminación (CRUD): 10 TPS combinados en todas las API de CRUD Puede solicitar un aumento de todas estas cuotas.

Tutorial: A/B testing with the Evidently sample application (Pruebas A/B con la aplicación de muestra de Evidently)

En esta sección encontrará un tutorial para utilizar Amazon CloudWatch Evidently para pruebas A/B. En este tutorial, se explica cómo usar la aplicación de muestra de Evidently, que es una aplicación de reacción simple. La aplicación de muestra se puede configurar para que muestre una característica de `showDiscount` o no lo haga. Cuando la característica se muestra para un usuario, el precio que se indica en el sitio web de compras aparece con un 20 % de descuento.

Además de cómo hacer para mostrar el descuento a algunos usuarios y a otros no, en este tutorial se indica cómo configurar Evidently para que recopile métricas de tiempo de carga de las páginas con ambas variaciones.

> **Warning**
>
> En este escenario, se requieren usuarios de IAM con acceso programático y credenciales de larga duración, lo que supone un riesgo de seguridad. Para ayudar a mitigar este riesgo, le recomendamos que brinde a estos usuarios únicamente los permisos que necesitan para

realizar la tarea y que los elimine cuando ya no los necesiten. Las claves de acceso se pueden actualizar si es necesario. Para obtener más información, consulte Actualización de claves de acceso en la Guía de usuario de IAM.

Paso 1: Descargar una aplicación de muestra

Comience descargando la aplicación de muestra de Evidently.

Para descargar la aplicación de muestra

1. Descargue la aplicación de muestra del siguiente bucket de Amazon S3:

```
https://evidently-sample-application.s3.us-west-2.amazonaws.com/evidently-sample-shopping-app.zip
```

2. Descomprima el paquete.

Paso 2: Agregar el punto de conexión de Evidently y configurar las credenciales

A continuación, agregue la región y el punto de conexión de Evidently al archivo `config.js` que se encuentra en el directorio `src` del paquete de la aplicación de muestra, como se indica en el siguiente ejemplo:

```
evidently: {
    REGION: "us-west-2",
    ENDPOINT: "https://evidently.us-west-2.amazonaws.com (https://evidently.us-west-2.amazonaws.com/)",
},
```

También debe asegurarse de que la aplicación tenga permiso para llamar a CloudWatch Evidently.

Para otorgar permisos a la aplicación de muestra para llamar a Evidently

1. Fedérela a su cuenta de AWS.
2. Cree un usuario de IAM y adjunte la política AmazonCloudWatchEvidentlyFullAccess a este usuario.
3. Tome nota del ID de clave de acceso y de la clave de acceso secreta del usuario de IAM, ya que los necesitará en el paso siguiente.

4. En el mismo archivo `config.js` que modificó antes en esta sección, ingrese los valores del ID de clave de acceso y de la clave de acceso secreta, como se muestra en el ejemplo siguiente:

```
credential: {
   accessKeyId: "Access key ID",
   secretAccessKey: "Secret key"
}
```

> **Important**
>
> Realizamos este paso para que la aplicación de muestra sea lo más sencilla de probar posible. No es recomendable que incluya su credencial de usuario de IAM en su aplicación de producción real. En lugar de ello, le recomendamos utilizar Amazon Cognito para la autenticación. Para obtener más información, consulte Integración de Amazon Cognito en aplicaciones web y móviles.

Paso 3: Configurar el código para la evaluación de la característica

Cuando utilice CloudWatch Evidently para evaluar una característica, debe utilizar la operación EvaluateFeature (Característica de evaluación) para seleccionar una variación de característica de forma aleatoria para cada sesión de usuario. Esta operación asigna sesiones de usuario a cada variación de la característica, según los porcentajes especificados en el experimento.

Para configurar el código de evaluación de características para la aplicación de demostración de la biblioteca

1. Agregue el creador de clientes al archivo `src/App.jsx` para que la aplicación de muestra pueda llamar a Evidently.

```
import Evidently from 'aws-sdk/clients/evidently';
import config from './config';

const defaultClientBuilder = (
    endpoint,
    region,
) => {
    const credentials = {
        accessKeyId: config.credential.accessKeyId,
        secretAccessKey: config.credential.secretAccessKey
```

```
    }
    return new Evidently({
        endpoint,
        region,
        credentials,
    });
};
```

2. Agregue lo siguiente a la sección del código `const App` para iniciar el cliente.

```
  if (client == null) {
      client = defaultClientBuilder(
        config.evidently.ENDPOINT,
        config.evidently.REGION,
      );
```

3. Forme `evaluateFeatureRequest` agregando el siguiente código. Este código rellena con anticipación el nombre del proyecto y el nombre de la característica que recomendamos más adelante en este tutorial. Puede sustituirlos por los nombres de sus propios proyecto y característica siempre que especifique también esos nombres de proyecto y característica en la consola de Evidently.

```
const evaluateFeatureRequest = {
              entityId: id,
              // Input Your feature name
              feature: 'showDiscount',
              // Input Your project name'
              project: 'EvidentlySampleApp',
            };
```

4. Agregue el código para llamar a Evidently para efectuar la evaluación de la característica. Cuando se envía la solicitud, Evidently asigna aleatoriamente la sesión del usuario para que vea la característica de `showDiscount` o no lo haga.

```
client.evaluateFeature(evaluateFeatureRequest).promise().then(res => {
    if(res.value?.boolValue !== undefined) {
        setShowDiscount(res.value.boolValue);
    }
    getPageLoadTime()
 })
```

Paso 4: Configurar el código para las métricas del experimento

Para la métrica personalizada, utilice la API `PutProjectEvents` de Evidently para enviar resultados de métricas a Evidently. En los siguientes ejemplos, aprenderá a configurar la métrica personalizada y enviar datos de experimentos a Evidently.

Agregue la siguiente función para calcular el tiempo de carga de la página y utilice `PutProjectEvents` para enviar los valores de la métrica a Evidently. Agregue la siguiente función a `Home.tsx` y llame a esta función dentro de la API `EvaluateFeature`:

```
const getPageLoadTime = () => {
    const timeSpent = (new Date().getTime() - startTime.getTime()) * 1.000001;
    const pageLoadTimeData = `{
      "details": {
        "pageLoadTime": ${timeSpent}
      },
      "UserDetails": { "userId": "${id}", "sessionId": "${id}"}
    }`;
    const putProjectEventsRequest = {
      project: 'EvidentlySampleApp',
      events: [
        {
          timestamp: new Date(),
          type: 'aws.evidently.custom',
          data: JSON.parse(pageLoadTimeData)
        },
      ],
    };
    client.putProjectEvents(putProjectEventsRequest).promise();
  }
```

A continuación, se muestra cómo debe verse el archivo `App.js` después de todas las modificaciones que le ha realizado desde que lo descargó.

```
import React, { useEffect, useState } from "react";
import { BrowserRouter as Router, Switch } from "react-router-dom";
import AuthProvider from "contexts/auth";
import CommonProvider from "contexts/common";
import ProductsProvider from "contexts/products";
import CartProvider from "contexts/cart";
import CheckoutProvider from "contexts/checkout";
import RouteWrapper from "layouts/RouteWrapper";
```

```
import AuthLayout from "layouts/AuthLayout";
import CommonLayout from "layouts/CommonLayout";
import AuthPage from "pages/auth";
import HomePage from "pages/home";
import CheckoutPage from "pages/checkout";
import "assets/scss/style.scss";
import { Spinner } from 'react-bootstrap';

import Evidently from 'aws-sdk/clients/evidently';
import config from './config';

const defaultClientBuilder = (
  endpoint,
  region,
) => {
  const credentials = {
    accessKeyId: config.credential.accessKeyId,
    secretAccessKey: config.credential.secretAccessKey
  }
  return new Evidently({
    endpoint,
    region,
    credentials,
  });
};

const App = () => {
  const [isLoading, setIsLoading] = useState(true);
  const [startTime, setStartTime] = useState(new Date());
  const [showDiscount, setShowDiscount] = useState(false);
  let client = null;
  let id = null;

  useEffect(() => {
    id = new Date().getTime().toString();
    setStartTime(new Date());
    if (client == null) {
      client = defaultClientBuilder(
        config.evidently.ENDPOINT,
        config.evidently.REGION,
      );
    }
    const evaluateFeatureRequest = {
      entityId: id,
```

```
      // Input Your feature name
      feature: 'showDiscount',
      // Input Your project name'
      project: 'EvidentlySampleApp',
    };

    // Launch
    client.evaluateFeature(evaluateFeatureRequest).promise().then(res => {
      if(res.value?.boolValue !== undefined) {
        setShowDiscount(res.value.boolValue);
      }
    });

    // Experiment
    client.evaluateFeature(evaluateFeatureRequest).promise().then(res => {
      if(res.value?.boolValue !== undefined) {
        setShowDiscount(res.value.boolValue);
      }
      getPageLoadTime()
    })

    setIsLoading(false);
  },[]);

  const getPageLoadTime = () => {
    const timeSpent = (new Date().getTime() - startTime.getTime()) * 1.000001;
    const pageLoadTimeData = `{
      "details": {
        "pageLoadTime": ${timeSpent}
      },
      "UserDetails": { "userId": "${id}", "sessionId": "${id}"}
    }`;
    const putProjectEventsRequest = {
      project: 'EvidentlySampleApp',
      events: [
        {
          timestamp: new Date(),
          type: 'aws.evidently.custom',
          data: JSON.parse(pageLoadTimeData)
        },
      ],
    };
    client.putProjectEvents(putProjectEventsRequest).promise();
  }
```

```
  return (
    !isLoading? (
    <AuthProvider>
      <CommonProvider>
        <ProductsProvider>
          <CartProvider>
            <CheckoutProvider>
              <Router>
                <Switch>
                  <RouteWrapper
                    path="/"
                    exact
                    component={() => <HomePage showDiscount={showDiscount}/>}
                    layout={CommonLayout}
                  />
                  <RouteWrapper
                    path="/checkout"
                    component={CheckoutPage}
                    layout={CommonLayout}
                  />
                  <RouteWrapper
                    path="/auth"
                    component={AuthPage}
                    layout={AuthLayout}
                  />
                </Switch>
              </Router>
            </CheckoutProvider>
          </CartProvider>
        </ProductsProvider>
      </CommonProvider>
    </AuthProvider> ) : (
      <Spinner animation="border" />
    )
  );
};

export default App;
```

Cada vez que un usuario visita la aplicación de muestra, se envía una métrica personalizada a Evidently para realizar un análisis. Evidently analiza cada métrica y muestra los resultados en tiempo real en el panel de Evidently. En el ejemplo siguiente se muestra una carga métrica:

```
[ {"timestamp": 1637368646.468, "type": "aws.evidently.custom", "data": "{\"details
\":{\"pageLoadTime\":2058.002058},\"userDetails\":{\"userId\":\"1637368644430\",
\"sessionId\":\"1637368644430\"}}" } ]
```

Paso 5: Crear el proyecto, la característica y el experimento

A continuación, cree el proyecto, la característica y el experimento en la consola de CloudWatch Evidently.

Para crear el proyecto, la característica y el experimento de este tutorial

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Señales de aplicación, Evidently.
3. Elija Create project (Crear proyecto) y complete los campos. Debe utilizar **`EvidentlySampleApp`** para que el nombre del proyecto para la muestra funcione correctamente. En Evaluation event storage(Almacenamiento de eventos de evaluación), elija Don't store Evaluation events (No almacenar eventos de evaluación).

   Luego de completar los campos, elija Create project (Crear proyecto).

   Para obtener más información, consulte Crear un nuevo proyecto de .
4. Una vez creado el proyecto, cree una característica en ese proyecto. Nombre la característica como **`showDiscount`** (Mostrar descuento). En esta característica, cree dos variaciones del tipo **`Boolean`** (booleano). Nombre la primera variación como **`disable`** con un valor **`False`** (falso) y nombre la segunda variación como **`enable`** con un valor **`True`** (verdadero).

   Para obtener más información acerca de la creación de una característica, consulte Agregue una característica a un proyecto.
5. Una vez que haya creado la característica, cree un experimento en el proyecto. Nombre el experimento como **`pageLoadTime`** (Tiempo de carga de la página).

   En este experimento se utilizará una métrica personalizada denominada `pageLoadTime`(Tiempo de carga de la página) que mide el tiempo de carga de la página que está bajo prueba. Las métricas personalizadas para experimentos se crean con Amazon EventBridge. Para obtener más información acerca de EventBridge, consulte ¿Qué es Amazon EventBridge?

   Para crear esa métrica personalizada, realice lo siguiente cuando cree el experimento:

- En Metrics (Métricas), para Metric source (Origen de métricas), elija Custom metrics (Métricas personalizadas).
- En Metric name (Nombre de métrica), ingrese **`pageLoadTime`** (Tiempo de carga de página).
- En Meta, elija Decrease (Disminuir). Esto indica que deseamos que un valor inferior de esta métrica indique la mejor variación de la característica.
- En Metric rule (Regla de la métrica), ingrese lo siguiente:
  - En Entity ID (ID de entidad), escriba **`UserDetails.userId`**.
  - En Value key (Clave de valor), ingrese **`details.pageLoadTime`**.
  - En Units (Unidades), ingrese **`ms`**.
- Elija Add metric (Agregar métrica).

En Audiences (Audiencias), seleccione 100 % para que se incluyan todos los usuarios en el experimento. Configure la división de tráfico entre las variaciones para que cada una sea del 50%.

Luego, para crear el experimento, elija Create Experiment (Crear experimento). Una vez que lo haya creado, no iniciará hasta que le indique a Evidently que debe iniciar.

Paso 6: Iniciar el experimento y probar CloudWatch Evidently

Los pasos finales son iniciar el experimento e iniciar la aplicación de muestra.

Para iniciar el experimento tutorial

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Señales de aplicación, Evidently.
3. Elija el proyecto EvidentlySampleApp.
4. Elija la pestaña Experiments (Experimentos).
5. Elija el botón situado junto a PageLoadTime (Tiempo de carga de página) y luego, Actions (Acciones), Start experiment (Iniciar experimento).
6. Elija el horario de finalización del experimento.
7. Elija Start experiment (Iniciar experimento).

   El experimento iniciará de forma inmediata.

Luego, inicie la aplicación de muestra de Evidently con el comando siguiente:

```
npm install -f && npm start
```

Una vez que la aplicación se haya iniciado, se le asignará una de las dos variaciones de características que se están probando. Una variación mostrará un “20 % de descuento” y la otra no lo hará. Siga actualizando la página para ver las diferentes variaciones.

> **Note**
>
> Evidently tiene evaluaciones complicadas. Las evaluaciones de características son deterministas, lo que significa que para el mismo `entityId` y característica, el usuario recibirá la misma asignación de variaciones. El único cambio de asignaciones de variaciones de tiempo se produce cuando se agrega una entidad a una anulación o se marca el tráfico del experimento.
> Sin embargo, para que el tutorial de la aplicación de muestra sea más sencillo para usted, Evidently reasigna la evaluación de la característica de la aplicación de muestra cada vez que se actualiza la página, de modo que pueda experimentar ambas variaciones sin tener que agregar anulaciones.

Solución de problemas

Le recomendamos utilizar la versión 6.14.14 de `npm`. Si aparece algún error sobre la creación o el inicio de la aplicación de muestra y está utilizando una versión diferente de npm, haga lo siguiente.

Para instalar la versión 6.14.14 de **`npm`**

1. Utilice un navegador para conectarse a https://nodejs.org/download/release/v14.17.5/.
2. Descargue node-v14.17.5.pkg y ejecute este paquete para instalar npm.

   Si se indica un error de `webpack not found`, diríjase a la carpeta `evidently-sample-shopping-app` e intente lo siguiente:

   a. Elimine `package-lock.json`
   b. Elimine `yarn-lock.json`
   c. Elimine `node_modules`
   d. Elimine la dependencia de Webpack de `package.json`.

e. Ejecute lo siguiente:

```
npm install -f && npm
```

Uso de CloudWatch RUM

Con CloudWatch RUM, puede llevar a cabo una supervisión real de usuarios para recopilar y ver datos del lado del cliente sobre el rendimiento de su aplicación web desde las sesiones de usuarios reales, casi en tiempo real. Los datos que puede visualizar y analizar incluyen tiempos de carga de páginas, errores del lado del cliente y comportamiento del usuario. Puede ver estos datos todos juntos y también ver desgloses por los navegadores y dispositivos que utilizan sus clientes.

Puede utilizar los datos recopilados para identificar y depurar rápidamente los problemas de rendimiento del lado del cliente. CloudWatch RUM lo ayudará a visualizar anomalías en el rendimiento de la aplicación y a encontrar datos de depuración relevantes, como mensajes de error, seguimientos de pila y sesiones de usuario. También puede utilizar RUM para comprender el intervalo de impacto del usuario final, incluido el número de usuarios, las geolocalizaciones y los navegadores utilizados.

Los datos del usuario final que recopila para CloudWatch RUM se conservan durante 30 días y, a continuación, se eliminan de forma automática. Si desea conservar los eventos de RUM durante más tiempo, puede elegir que la supervisión de aplicaciones envíe copias de los eventos a CloudWatch Logs en su cuenta. A continuación, puede ajustar el periodo de retención de ese grupo de registros.

Si desea usar RUM, cree un monitor de aplicaciones y proporcione información. RUM genera un fragmento de JavaScript para que lo pegue en su aplicación. El fragmento extrae el código del cliente web de RUM. El cliente web de RUM captura datos de un porcentaje de las sesiones de usuario de la aplicación, que se muestran en un panel prediseñado. Puede especificar el porcentaje de sesiones de usuario del cual desea recopilar datos.

CloudWatch RUM se integra con Application Signals, que puede detectar y supervisar los servicios de aplicaciones, los clientes, los valores controlados de Synthetics y las dependencias de los servicios. Use Application Signals para ver una lista o un mapa visual de sus servicios, ver las métricas del estado en función de los objetivos de nivel de servicio (SLO) y profundizar para ver los seguimientos de X-Ray correlacionados para una solución de problemas más detallada. Para ver las solicitudes de páginas de clientes de RUM en Application Signals, active el seguimiento activo de X-Ray al crear un monitor de aplicaciones o al configurar de forma manual el cliente web de RUM. Los

clientes de RUM se muestran en el Asignación de servicio conectado a sus servicios y en la página de Detalles del servicio de los servicios a los que llaman.

El cliente web de RUM es de código abierto. Para obtener más información, consulte CloudWatch RUM web client (Cliente web de CloudWatch RUM).

Consideraciones sobre el rendimiento

Esta sección abarca las consideraciones de rendimiento del uso de CloudWatch.

- Impacto del rendimiento de carga: El cliente web de CloudWatch RUM se puede instalar en la aplicación web como un módulo JavaScript o cargar en la aplicación web de forma asíncrona desde una red de entrega de contenido (CDN, Content Delivery Network). No bloquea el proceso de carga de la aplicación. CloudWatch RUM está diseñado para que no se produzca un impacto perceptible durante el tiempo de carga de la aplicación.
- Impacto del tiempo de ejecución: El cliente web de RUM realiza el procesamiento para registrar y enviar datos de RUM al servicio de CloudWatch RUM. Debido a que los eventos son poco frecuentes y la cantidad de procesamiento es pequeña, CloudWatch RUM está diseñado para que no se produzca un impacto detectable en el rendimiento de la aplicación.
- Impacto de la red: El cliente web de RUM envía datos de forma periódica al servicio de CloudWatch RUM. Los datos se distribuyen en intervalos regulares mientras la aplicación se ejecuta y, también, inmediatamente antes de que el navegador descargue la aplicación. Los datos enviados inmediatamente antes de que el navegador descargue la aplicación se envían como señales, que están diseñadas para no tener un impacto detectable durante el tiempo de descarga de la aplicación.

Precios de RUM

Con CloudWatch RUM, incurre en cargos por cada evento de RUM que recibe CloudWatch RUM. Cada elemento de datos recopilado mediante el cliente web de RUM se considera un evento de RUM. Algunos ejemplos de eventos de RUM incluyen una vista de página, un error de JavaScript y un error HTTP. En algunas opciones, cada supervisión de aplicaciones recopila los tipos de eventos. Puede activar o desactivar opciones para recopilar eventos de telemetría de rendimiento, errores de JavaScript, errores HTTP y seguimientos de X-Ray. Para obtener más información sobre estas opciones, consulte Paso 2: Cree un monitor de aplicaciones y Información recopilada por el cliente web de CloudWatch RUM. Para obtener más información sobre precios, consulte Precios de Amazon CloudWatch.

Disponibilidad por región

Actualmente, CloudWatch RUM está disponible en las siguientes regiones:

- Este de EE. UU. (Norte de Virginia)
- Este de EE. UU. (Ohio)
- Oeste de EE. UU. (Norte de California)
- Oeste de EE. UU. (Oregón)
- África (Ciudad del Cabo)
- Asia-Pacífico (Yakarta)
- Asia-Pacífico (Bombay)
- Asia-Pacífico (Hyderabad)
- Asia-Pacífico (Melbourne)
- Asia-Pacífico (Osaka)
- Asia-Pacífico (Seúl)
- Asia-Pacífico (Singapur)
- Asia-Pacífico (Sídney)
- Asia-Pacífico (Tokio)
- Canadá (centro)
- Europa (Fráncfort)
- Europa (Irlanda)
- Europa (Londres)
- Europa (Milán)
- Europa (París)
- Europa (España)
- Europa (Estocolmo)
- Europa (Zúrich)
- Medio Oriente (Baréin)
- Medio Oriente (EAU)
- América del Sur (São Paulo)

Temas

Políticas de IAM para utilizar CloudWatch RUM

Para poder administrar CloudWatch RUM por completo, debe haber iniciado sesión como usuario o rol de IAM que tenga la política de IAM de AmazonCloudWatchRUMFullAccess (Acceso completo a Amazon CloudWatch RUM). Además, es posible que necesite otras políticas o permisos:

- Para crear una supervisión de aplicaciones que cree un nuevo grupo de identidades de Amazon Cognito para obtener autorización, debe tener el rol de IAM de Admin (Administrador) o la política de IAM de AdministratorAccess (Acceso del administrador).
- Para crear una supervisión de aplicaciones que envíe datos a CloudWatch Logs, debe iniciar sesión en un rol o política de IAM que tenga los siguientes permisos:

```
{
    "Effect": "Allow",
    "Action": [
        "logs:PutResourcePolicy"
    ],
    "Resource": [
        "*"
```

```
    ]
}
```

A otros usuarios que necesitan ver los datos de CloudWatch RUM, pero no necesitan crear recursos de CloudWatch RUM, se les puede conceder la política de AmazonCloudWatchRUMReadOnlyAccess (Acceso de solo lectura a Amazon CloudWatch RUM).

Configuración de una aplicación para utilizar CloudWatch RUM

Siga los pasos de estas secciones para configurar la aplicación con el fin de comenzar a utilizar CloudWatch RUM para recopilar datos de rendimiento de sesiones de usuarios reales.

Temas

Paso 1: Autorice a la aplicación para enviar datos a AWS

Para utilizar CloudWatch RUM, la aplicación debe tener autorización.

Tiene tres opciones para configurar la autorización:

- Permita que CloudWatch RUM cree un nuevo grupo de identidades de Amazon Cognito para la aplicación. Este método requiere el menor esfuerzo para configurarlo. Esta es la opción predeterminada.

  El grupo de identidades contendrá una identidad sin autenticar. Esto permite que el cliente web de CloudWatch RUM envíe datos a CloudWatch RUM sin autenticar al usuario de la aplicación.

  El grupo de identidades de Amazon Cognito tiene un rol de IAM adjunto. La identidad sin autenticar de Amazon Cognito permite que el cliente web asuma el rol de IAM autorizado para enviar datos a CloudWatch RUM.

- Uso de un grupo de identidades de Amazon Cognito existente. En este caso, también debe modificar el rol de IAM que se adjunta al grupo de identidades. Use esta opción para los grupos de identidades que admiten usuarios no autenticados. Solo puede utilizar grupos de identidades dentro de una misma región.
- Utilice la autenticación de un proveedor de identidades existente que ya haya configurado. En este caso, debe obtener credenciales del proveedor de identidad y la aplicación debe reenviar estas credenciales al cliente web de RUM.

  Use esta opción para los grupos de identidades que solo admiten usuarios autenticados.

En las siguientes secciones se detallan estas opciones.

CloudWatch RUM crea un nuevo grupo de identidades de Amazon Cognito

Esta es la opción más sencilla de configurar y, si elige ponerla en práctica, no se requieren más pasos de configuración. Debe disponer de permisos administrativos para utilizar esta opción. Para obtener más información, consulte Políticas de IAM para utilizar CloudWatch RUM.

Con esta opción, CloudWatch RUM crea los siguientes recursos:

- Nuevo grupo de identidades de Amazon Cognito
- Identidad sin autenticar de Amazon Cognito. Esto permite que el cliente web de RUM asuma un rol de IAM sin autenticar al usuario de la aplicación.
- El rol de IAM que asumirá el cliente web de RUM. La política de IAM adjunta a este rol le permite utilizar la API de `PutRumEvents` con el recurso de supervisión de aplicaciones. En otras palabras, permite que el cliente web de RUM envíe datos a RUM.

El cliente web de RUM utiliza la identidad de Amazon Cognito para obtener credenciales de AWS. Las credenciales de AWS se asocian al rol de IAM. El rol de IAM está autorizado para utilizar `PutRumEvents` con el recurso AppMonitor.

Amazon Cognito envía el token de seguridad necesario para permitir que la aplicación envíe datos a CloudWatch RUM. El fragmento de código JavaScript que genera CloudWatch RUM incluye las siguientes líneas para habilitar la autenticación.

```
{
```

```
    identityPoolId: [identity pool id],  // e.g., 'us-west-2:EXAMPLE4a-66f6-4114-902a-
EXAMPLEbad7'
   }
);
```

Uso del grupo de identidades de Amazon Cognito existente

Si elige utilizar un grupo de identidades de Amazon Cognito existente, especifique el grupo de identidades al agregar la aplicación a CloudWatch RUM. El grupo debe ser compatible con el acceso a identidades sin autenticar. Solo puede utilizar grupos de identidades dentro de una misma región.

También debe agregar los siguientes permisos a la política de IAM asociada al rol de IAM que está asociado a este grupo de identidades.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "rum:PutRumEvents",
            "Resource": "arn:aws:rum:[region]:[accountid]:appmonitor/[app monitor
 name]"
        }
    ]
}
```

Entonces, Amazon Cognito enviará el token de seguridad necesario para permitir que la aplicación tenga acceso a CloudWatch RUM.

Proveedor de terceros

Si elige utilizar la autenticación privada de un proveedor externo, debe obtener las credenciales del proveedor de identidad y reenviarlas a AWS. La mejor forma de hacerlo es a través de un proveedor de tokens de seguridad. Puede utilizar cualquier proveedor de tokens de seguridad, incluido Amazon Cognito conAWS Security Token Service. Para obtener más información sobre AWS STS, consulte Bienvenido a la referencia sobre las API de AWS Security Token Service.

Si desea utilizar Amazon Cognito como proveedor de tokens en este contexto, puede configurar Amazon Cognito para que funcione con un proveedor de autenticación. Para obtener más información, consulte Introducción a los grupos de identidades de Amazon Cognito (identidades federadas).

Después de configurar Amazon Cognito para que funcione con su proveedor de identidad, también debe hacer lo siguiente:

- Cree un rol de IAM con los siguientes permisos. Su aplicación utilizará este rol para acceder a AWS.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "rum:PutRumEvents",
            "Resource": "arn:aws:rum:[region]:[accountID]:appmonitor/[app monitor
 name]"
        }
    ]
}
```

- Agregue lo siguiente a su aplicación para que envíe las credenciales de su proveedor a CloudWatch RUM. Inserte la línea para que se ejecute después de que un usuario haya iniciado sesión en la aplicación y la aplicación haya recibido las credenciales que se utilizarán para acceder a AWS.

```
cwr('setAwsCredentials', {/* Credentials or CredentialProvider */});
```

Para obtener más información acerca de los proveedores de credenciales en JavaScript SDK de AWS, consulte Configuración de credenciales en un navegador web en la guía para desarrolladores v3 para JavaScript SDK, Configuración de credenciales en un navegador web en la guía para desarrolladores v2 para JavaScript SDK y @aws-sdk/credential-providers.

También puede utilizar el SDK para el cliente web de CloudWatch RUM para configurar los métodos de autenticación del cliente web. Para obtener más información sobre el SDK del cliente web, consulte.CloudWatch RUM web client SDK (SDK del cliente web de CloudWatch RUM).

Paso 2: Cree un monitor de aplicaciones

Para empezar a utilizar CloudWatch RUM con su aplicación, cree un spp monitor (monitor de aplicaciones). Cuando se crea el monitor de aplicaciones, RUM genera un fragmento de JavaScript para que lo pegue en su aplicación. El fragmento extrae el código del cliente web de RUM. El cliente

web de RUM captura datos de un porcentaje de las sesiones de usuario de la aplicación y los envía a RUM.

Para crear un monitor de aplicaciones

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Señales de aplicación, RUM.
3. Elija Add app monitor (Agregar monitor de aplicaciones).
4. Ingrese la información y la configuración para la aplicación:

   - En App monitor name (Nombre del monitor de aplicaciones), ingrese un nombre que se utilizará para identificar este monitor de aplicaciones en la consola de CloudWatch RUM.
   - En Application domain (Dominio de aplicación), ingrese el nombre de dominio de nivel superior en el que la aplicación tiene autoridad administrativa. Este debe estar en formato de dominio URL.

     Elija Include sub domains (Incluir subdominios) para que el monitor de aplicaciones también recopile datos de todos los subdominios del dominio de nivel superior.
5. En Configure RUM data collection (Configuración de la recopilación de datos de RUM), especifique si desea que el monitor de aplicaciones recopile cada uno de los siguientes elementos:

   - Telemetría de rendimiento: recopila información sobre los tiempos de carga de la página y de carga de recursos.
   - Errores JavaScript: recopila información sobre los errores de JavaScript no controlados generados por su aplicación.
   - Errores HTTP: recopila información sobre los errores HTTP lanzados por la aplicación.

   La selección de estas opciones proporciona más información sobre la aplicación, pero también genera más eventos de CloudWatch RUM y, por lo tanto, incurre en más cargos.

   Si no selecciona ninguno de estos, el monitor de aplicaciones seguirá recopilando eventos de inicio de sesión e ID de página para que pueda ver cuántos usuarios están utilizando la aplicación, incluidos los desgloses por tipo y versión del sistema operativo, tipo y versión del navegador, tipo de dispositivo y ubicación.
6. Seleccione Check this option to allow the CloudWatch RUM Web Client to set cookies (Marque esta opción para permitir que el cliente web de CloudWatch RUM configure cookies) si desea

poder recopilar ID de usuario e ID de sesión de sesiones de usuario de muestra. Los ID de usuario se generan de forma aleatoria mediante RUM. Para obtener más información, consulte Cookies del cliente web de CloudWatch RUM (o tecnologías similares).

7. En Session samples (Muestras de sesiones), ingrese el porcentaje de sesiones de usuario que se utilizarán para recopilar datos de RUM. El valor predeterminado es 100 %. Si reduce este número, obtendrá menos datos, pero reducirá los cargos. Para obtener más información sobre los precios de RUM, consulte Precios de RUM.
8. Los datos del usuario final que recopila para CloudWatch RUM se conservan durante 30 días y, luego, se eliminan. Si desea conservar copias de eventos de RUM en CloudWatch Logs y configurar el tiempo durante el cual se retendrán estas copias, elija Check this option to store your application telemetry data in your CloudWatch Logs account (Marque esta opción para almacenar los datos de telemetría de aplicaciones en su cuenta de CloudWatch Logs) en Data storage (Almacenamiento de datos). El grupo de registros de CloudWatch Logs conserva los datos durante 30 días de forma predeterminada. Puede administrar su periodo de retención de registros en la consola de CloudWatch Logs.
9. En Authorization (Autorización), especifique si desea utilizar un grupo de identidades de Amazon Cognito nuevo o existente, o utilizar otro proveedor de identidad. Crear un nuevo grupo de identidades es la opción más sencilla ya que no requiere ningún otro paso de configuración. Para obtener más información, consulte Paso 1: Autorice a la aplicación para enviar datos a AWS.

   Para llevar a cabo la creación de un nuevo grupo de identidades de Amazon Cognito se necesitan permisos administrativos. Para obtener más información, consulte Políticas de IAM para utilizar CloudWatch RUM.
10. (Opcional) De forma predeterminada, cuando agrega el fragmento de código de RUM a la aplicación, el cliente web inyecta la etiqueta JavaScript para supervisar el uso en el código HTML de todas las páginas de la aplicación. Para cambiar esto, elija Configure pages (Configurar páginas) y luego, Include only these pages (Incluir solo estas páginas) o Exclude these pages (Excluir estas páginas). Luego, especifique las páginas que desea incluir o excluir. Para especificar las páginas que desea incluir o excluir, ingrese las URL completas. Para especificar páginas adicionales, elija Add URL (Agregar URL).
11. Para activar el rastreo AWS X-Ray de las sesiones de usuario que el monitor de aplicaciones ofrece como muestra, seleccione Rastreo activo y, luego, seleccione Rastrear mi servicio con AWS X-Ray.

Si selecciona esta opción, el monitor de aplicaciones rastreará las solicitudes `XMLHttpRequest` y `fetch` realizadas durante las sesiones de usuario de muestra. A continuación, podrá ver los seguimientos y segmentos de estas sesiones del usuario en el panel de RUM y las páginas de detalles de seguimiento y del mapa de seguimiento de X-Ray. Estas sesiones del usuario también se mostrarán como páginas de cliente en Application Signals una vez que las haya habilitado para su aplicación.

Al realizar cambios de configuración adicionales en el cliente web de CloudWatch RUM, puede agregar un encabezado de seguimiento de X-Ray a las solicitudes HTTP para habilitar el seguimiento de extremo a extremo de las sesiones de usuario hasta los servicios administrados de AWS. Para obtener más información, consulte Habilitación del seguimiento integral de X-Ray.

12. (Opcional) Para agregar etiquetas al monitor de aplicaciones, elija Tags (Etiquetas), Add new tag (Agregar nueva etiqueta).

    Luego, en Key (Clave), ingrese un nombre para la etiqueta. Puede agregar un valor opcional para la etiqueta en Valor.

    Para agregar otra etiqueta, vuelva a elegir Add new tag (Agregar nueva etiqueta).

    Para obtener más información, consulte Tagging AWS Resources (Etiquetado de recursos de).
13. Elija Add app monitor (Agregar monitor de aplicaciones).
14. En la sección Sample code (Código de muestra), puede copiar el fragmento de código a usar para agregar a la aplicación. Le recomendamos que elija JavaScript o TypeScript y utilice NPM para instalar el cliente web de CloudWatch RUM como módulo JavaScript.

    De forma alternativa, puede elegir HTML para utilizar una red de entrega de contenido (CDN) para instalar el cliente web de CloudWatch RUM. La desventaja de utilizar una CDN es que el cliente web suele estar bloqueado por bloqueadores de anuncios.
15. Elija Copy (Copiar) o Download (Descargar) y luego elija Done (Hecho).

(Opcional) Paso 3: Modifique el fragmento de código de forma manual para configurar el cliente web de CloudWatch RUM

Puede modificar el fragmento de código antes de insertarlo en la aplicación, si desea activar o desactivar varias opciones. Para obtener más información, consulte la documentación del cliente web de CloudWatch RUM.

Hay tres opciones de configuración que definitivamente debe tener en cuenta, como se detalla en estas secciones.

Impedir la recopilación de URL de recursos que podrían contener información personal

El cliente web de CloudWatch RUM está configurado de forma predeterminada para registrar las URL de los recursos que se descargan en la aplicación. Estos recursos incluyen archivos HTML, imágenes, archivos CSS, archivos JavaScript, entre otros. En algunas aplicaciones, las URL pueden contener información de identificación personal (PII, por sus siglas en inglés).

Si este es el caso de la aplicación, le recomendamos ampliamente que desactive la recopilación de URL de recursos mediante el uso de `recordResourceUrl: false` en la configuración de fragmentos de código, antes de insertarlo en la aplicación.

Registro manual de vistas de página

El cliente web registra las vistas de página de forma predeterminada cuando se carga la página por primera vez y cuando se llama a la API del historial del navegador. El ID de página predeterminado es `window.location.pathname`. Sin embargo, en algunos casos, es posible que desee anular este comportamiento e instrumentar la aplicación para registrar las visitas a las páginas mediante programación. Si lo hace, podrá controlar el ID de la página y cuándo se registrará. Por ejemplo, pensemos en una aplicación web que tenga un URI con un identificador variable, como `/entity/123` o `/entity/456`. De forma predeterminada, CloudWatch RUM genera un evento de vista de página para cada URI con un ID de página distinto que coincida con el nombre de la ruta, pero es posible que desee agruparlos por el mismo ID de página. Para ello, desactive la automatización de las vistas de página del cliente web mediante la configuración de `disableAutoPageView` y utilice el comando `recordPageView` para establecer el ID de página deseado. Para obtener más información, consulte Configuraciones específicas de la aplicación en GitHub.

Ejemplo de script incrustado:

```
cwr('recordPageView', { pageId: 'entityPageId' });
```

Ejemplo de módulo de JavaScript:

```
awsRum.recordPageView({ pageId: 'entityPageId' });
```

Habilitación del seguimiento integral de X-Ray

Al momento de crear el monitor de aplicaciones, seleccione Trace my service with AWS X-Ray; se habilitará el seguimiento de las solicitudes `XMLHttpRequest` y `fetch` hechas durante las sesiones de usuario que el monitor de aplicaciones ofrece como muestra. A continuación, podrá ver los seguimientos de estas solicitudes de HTTP en el panel de CloudWatch RUM y las páginas de detalles de seguimiento y del mapa de seguimiento de X-Ray.

Estos seguimientos del lado del cliente no están conectados a seguimientos posteriores del lado del servidor de forma predeterminada. Para conectar los seguimientos del lado del cliente a los seguimientos del lado del servidor y habilitar el seguimiento de extremo a extremo, establezca la opción `addXRayTraceIdHeader` como `true` (verdadero) en el cliente web. Esto ocasiona que el cliente web de CloudWatch RUM agregue un encabezado de seguimiento de X-Ray a las solicitudes HTTP.

En el siguiente bloque de código, observará un ejemplo de adición de seguimientos del lado del cliente. Algunas opciones de configuración se omiten en este ejemplo para su legibilidad.

```
<script>
    (function(n,i,v,r,s,c,u,x,z){...})(
        'cwr',
        '00000000-0000-0000-0000-000000000000',
        '1.0.0',
        'us-west-2',
        'https://client.rum.us-east-1.amazonaws.com/1.0.2/cwr.js',
        {
            enableXRay: true,
            telemetries: [
                'errors',
                'performance',
                [ 'http', { addXRayTraceIdHeader: true } ]
            ]
        }
    );
</script>
```

> **Warning**
>
> La configuración del cliente web de CloudWatch RUM para agregar un encabezado de seguimiento de X-Ray a las solicitudes HTTP puede ocasionar que el uso compartido de

recursos de origen cruzado (CORS) falle o invalide la firma de la solicitud si la solicitud está firmada con SigV4. Para obtener más información, consulte la documentación del cliente web de CloudWatch RUM. Le recomendamos ampliamente que pruebe la aplicación antes de agregar un encabezado de seguimiento de X-Ray del lado del cliente en un entorno de producción.

Para obtener más información, consulte la documentación del cliente web de CloudWatch RUM.

Paso 4: Inserte el fragmento de código en la aplicación

A continuación, insertará en la aplicación el fragmento de código que haya creado en la sección anterior.

> **Warning**
>
> El cliente web, descargado y configurado mediante el fragmento de código, utiliza cookies (o tecnologías similares) para recopilar datos del usuario final. Antes de insertar el fragmento de código, consulte Filtrar por atributos de metadatos en la consola.

Si no tiene el fragmento de código que se generó previamente, puede encontrarlo si sigue las instrucciones que se describen en ¿Cómo encuentro un fragmento de código que ya he generado?.

Para insertar el fragmento de código de CloudWatch RUM en la aplicación

1. Inserte el fragmento de código que haya copiado o descargado en la sección anterior dentro del elemento <head> de la aplicación. Insértelo antes del elemento <body> o cualquier otra etiqueta <script>.

   El siguiente es un ejemplo de un fragmento de código generado:

```
<script>
(function (n, i, v, r, s, c, x, z) {
    x = window.AwsRumClient = {q: [], n: n, i: i, v: v, r: r, c: c};
    window[n] = function (c, p) {
        x.q.push({c: c, p: p});
    };
    z = document.createElement('script');
    z.async = true;
```

```
    z.src = s;
    document.head.insertBefore(z, document.getElementsByTagName('script')[0]);
})('cwr',
    '194a1c89-87d8-41a3-9d1b-5c5cd3dafbd0',
    '1.0.0',
    'us-east-2',
    'https://client.rum.us-east-1.amazonaws.com/1.0.2/cwr.js',
    {
        sessionSampleRate: 1,
        identityPoolId: "us-east-2:c90ef0ac-e3b8-4d1a-b313-7e73cfd21443",
        endpoint: "https://dataplane.rum.us-east-2.amazonaws.com",
        telemetries: ["performance", "errors", "http"],
        allowCookies: true,
        enableXRay: false
    });
</script>
```

2. Si la aplicación es una aplicación web de varias páginas, debe repetir el paso 1 para cada página HTML que desee incluir en la recopilación de datos.

Paso 5: Pruebe la configuración del monitor de aplicaciones mediante la generación de eventos de usuario

Después de insertar el fragmento de código y se esté ejecutando la aplicación actualizada, puede probarlo al generar eventos de usuario de forma manual. Le recomendamos que realice las siguientes acciones. Esta prueba incurre en cargos estándar de CloudWatch RUM.

- Navegue por las páginas de su aplicación web.
- Cree varias sesiones de usuario, a través de distintos navegadores y dispositivos.
- Realice solicitudes.
- Ocasione errores de JavaScript.

Después de generar algunos eventos, podrá verlos en el panel de CloudWatch RUM. Para obtener más información, consulte Visualización del panel de CloudWatch RUM.

Los datos de las sesiones de usuario pueden tardar hasta 15 minutos en aparecer en el panel.

Si no puede ver los datos 15 minutos después de haber generado eventos en la aplicación, consulte Solución de problemas de CloudWatch RUM.

Configuración del cliente web de CloudWatch RUM

Las aplicaciones podrán utilizar uno de los fragmentos de código que haya generado CloudWatch RUM para instalar el cliente web de CloudWatch RUM. Los fragmentos generados admiten dos métodos de instalación: como módulo JavaScript a través de NPM o desde una red de entrega de contenido (CDN). Para obtener el máximo desempeño, le recomendamos utilizar el método de instalación de NPM. Para obtener más información sobre cómo utilizar este método, consulte Instalación como módulo JavaScript.

Si utiliza la opción de instalación de CDN, los bloqueadores de anuncios podrían bloquear la CDN predeterminada proporcionada por CloudWatch RUM. Esto deshabilita la supervisión de aplicaciones para los usuarios que tienen bloqueadores de anuncios instalados. Por ello, le recomendamos que utilice la CDN predeterminada solo para la incorporación inicial con CloudWatch RUM. Para obtener más información sobre cómo mitigar este problema, consulte Instrumentar la aplicación.

El fragmento de código se encuentra en la etiqueta <head> de un archivo HTML e instala el cliente web mediante la descarga del cliente web y su configuración para la aplicación que está supervisando. El fragmento es una función de ejecución automática que tiene un aspecto similar a lo que se detalla a continuación. En este ejemplo, se omite el cuerpo de la función del fragmento por motivos de legibilidad.

```
<script>
(function(n,i,v,r,s,c,u,x,z){...})(
'cwr',
'00000000-0000-0000-0000-000000000000',
'1.0.0',
'us-west-2',
'https://client.rum.us-east-1.amazonaws.com/1.0.2/cwr.js',
{ /* Configuration Options Here */ }
);
<script>
```

Argumentos

El fragmento de código acepta seis argumentos:

- un espacio de nombres para ejecutar comandos en el cliente web, como `'cwr'`
- el ID del monitor de aplicaciones, como `'00000000-0000-0000-0000-000000000000'`
- la versión de la aplicación, como `'1.0.0'`

- la región de AWSdel monitor de aplicaciones, como `'us-west-2'`
- la URL del cliente web, como `'https://client.rum.us-east-1.amazonaws.com/1.0.2/cwr.js'`
- opciones de configuración específicas de la aplicación. Para más información, consulte la siguiente sección.

Ignoración de errores

El cliente web CloudWatch RUM escucha todo tipo de errores que se producen en sus aplicaciones. Si su aplicación emite errores de JavaScript que no desea ver en el panel de CloudWatch RUM, puede configurar el cliente web de CloudWatch RUM para que filtre tales errores, de modo que solo le aparezcan los eventos de error relevantes en el panel de CloudWatch RUM. Por ejemplo, puede optar por no ver algunos errores de JavaScript en el panel porque ya ha identificado una solución para ellos y el volumen de tales errores oculta otros errores. También puede ignorar los errores que no pueda corregir porque pertenezcan a una biblioteca que sea propiedad de un tercero.

Para obtener más información sobre cómo instrumentar al cliente web para que filtre errores específicos de JavaScript, consulte el ejemplo de Errors (Errores) en la documentación de Github del cliente web.

Opciones de configuración

Para obtener información sobre las opciones de configuración disponibles para el cliente web de CloudWatch RUM, consulte la documentación del cliente web de CloudWatch RUM.

Regionalización

En esta sección, se ilustran las estrategias para utilizar CloudWatch RUM con aplicaciones de diferentes regiones.

Mi aplicación web está implementada en varias regiones de AWS

Si su aplicación web se implementa en varias regiones de AWS, tiene tres opciones:

- Implemente un monitor de aplicaciones en una región, en una cuenta, para que sirva a todas las regiones.
- Implemente monitores de aplicaciones independientes para cada región, en cuentas únicas.

- Implemente monitores de aplicaciones independientes para cada región, en cuentas únicas.

La ventaja de usar un monitor de aplicaciones es que todos los datos se centralizan en una visualización y todos los registros se escriben en el mismo grupo de registro en registros de CloudWatch. Con un único monitor de aplicaciones existe una pequeña cantidad de latencia adicional para las solicitudes y un único punto de error.

Usar varios monitores de aplicaciones elimina el punto único de error, pero evita que todos los datos se combinen en una sola visualización.

CloudWatch RUM no se ha lanzado en algunas regiones en las que está implementada mi aplicación

CloudWatch RUM se ha lanzado en muchas regiones y tiene una amplia cobertura geográfica. Si configura CloudWatch RUM en las regiones en las que está disponible, podrá disfrutar de sus ventajas. Los usuarios finales pueden estar en cualquier lugar y seguir teniendo sus sesiones incluidas si ha configurado un monitor de aplicaciones en la región a la que se conectan.

Sin embargo, CloudWatch RUM aún no se ha lanzado en AWS GovCloud (Este de EE. UU.), AWS GovCloud (Oeste de EE. UU.) ni en ninguna región de China. No puede enviar datos a CloudWatch RUM desde estas regiones.

Uso de grupos de páginas

Use grupos de páginas para asociar páginas diferentes de su aplicación entre sí, de modo que pueda ver los análisis agregados de los grupos de páginas. Por ejemplo, es posible que desee ver los tiempos de carga de páginas agregados de todas sus páginas de destino.

Para colocar páginas en grupos de páginas, agregue una o más etiquetas a los eventos de visualización de página en el cliente web de CloudWatch RUM. Los siguientes ejemplos ponen la página `/home` en el grupo de páginas denominado `en` y el grupo de páginas denominado `landing`.

Ejemplo de script incrustado

```
cwr('recordPageView', { pageId: '/home', pageTags: ['en', 'landing']});
```

Ejemplo de módulo de JavaScript

```
awsRum.recordPageView({ pageId: '/home', pageTags: ['en', 'landing']});
```

> **Note**
>
> Los grupos de páginas están diseñados para hacer más fácil agregar análisis en diferentes páginas. Para obtener información sobre cómo definir y manipular `pageIds` para su aplicación, consulte la sección Registro manual de las visitas a las páginas en (Opcional) Paso 3: Modifique el fragmento de código de forma manual para configurar el cliente web de CloudWatch RUM.

Especificación de metadatos personalizados

CloudWatch RUM adjunta datos adicionales a cada evento en forma de metadatos. Los metadatos de eventos constan de atributos en forma de pares clave-valor. Puede usar estos atributos para buscar o filtrar eventos en la consola de CloudWatch RUM. De forma predeterminada, CloudWatch RUM crea algunos metadatos que puede usar. Para obtener más información acerca de los metadatos predeterminados, consulte Metadatos del evento de RUM.

También puede utilizar el cliente web de CloudWatch RUM para agregar metadatos personalizados a los eventos de CloudWatch RUM. Algunos metadatos personalizados pueden ser atributos de sesión y atributos de página.

Para agregar metadatos personalizados, debe utilizar la versión 1.10.0 o posterior del cliente web de CloudWatch RUM.

Requisitos y sintaxis

Cada evento puede incluir hasta 10 atributos personalizados en los metadatos. Los requisitos de sintaxis para los atributos personalizados son los siguientes:

- Claves
  - Máximo de 128 caracteres
  - Puede contener caracteres alfanuméricos, caracteres de dos puntos (:) y guiones bajos (_).
  - No puede empezar por `aws:`.
  - No puede constar exclusivamente de ninguna de las palabras clave reservadas que figuren en la siguiente sección. Puede usar esas palabras clave como parte de un nombre de clave más largo.
- Valores

- Máximo de 256 caracteres
- Deben ser cadenas, números o valores booleanos

Palabras clave reservadas

No puede utilizar las siguientes palabras clave reservadas como nombres de clave completos. Puede utilizar las siguientes palabras clave como parte de un nombre de clave más largo, por ejemplo `applicationVersion`.

- `browserLanguage`
- `browserName`
- `browserVersion`
- `countryCode`
- `deviceType`
- `domain`
- `interaction`
- `osName`
- `osVersion`
- `pageId`
- `pageTags`
- `pageTitle`
- `pageUrl`
- `parentPageId`
- `platformType`
- `referrerUrl`
- `subdivisionCode`
- `title`
- `url`
- `version`

> **Note**
>
> CloudWatch RUM elimina los atributos personalizados de los eventos de RUM si un atributo incluye una clave o un valor inválido, o si ya se ha alcanzado el límite de 10 atributos personalizados por evento.

Adición de atributos de sesión

Si configura atributos de sesión personalizados, se agregan a todos los eventos de una sesión. Los atributos de sesión se configuran durante la inicialización del cliente web de CloudWatch RUM o en tiempo de ejecución mediante el comando `addSessionAttributes`.

Por ejemplo, puede agregar la versión de su aplicación como un atributo de sesión. A continuación, en la consola de CloudWatch RUM, puede filtrar los errores por versión para ver si hay un aumento en la tasa de errores asociado a una versión concreta de la aplicación.

Agregar un atributo de sesión en la inicialización, ejemplo de NPM

La sección de código en negrita agrega el atributo de sesión.

```
import { AwsRum, AwsRumConfig } from 'aws-rum-web';

try {
  const config: AwsRumConfig = {
    allowCookies: true,
    endpoint: "https://dataplane.rum.us-west-2.amazonaws.com",
    guestRoleArn: "arn:aws:iam::000000000000:role/RUM-Monitor-us-
west-2-000000000000-00xx-Unauth",
    identityPoolId: "us-west-2:00000000-0000-0000-0000-000000000000",
    sessionSampleRate: 1,
    telemetries: ['errors', 'performance'],
    sessionAttributes: {
        applicationVersion: "1.3.8"
    }
  };

  const APPLICATION_ID: string = '00000000-0000-0000-0000-000000000000';
  const APPLICATION_VERSION: string = '1.0.0';
  const APPLICATION_REGION: string = 'us-west-2';

  const awsRum: AwsRum = new AwsRum(
```

```
    APPLICATION_ID,
    APPLICATION_VERSION,
    APPLICATION_REGION,
    config
  );
} catch (error) {
  // Ignore errors thrown during CloudWatch RUM web client initialization
}
```

Agregar un atributo de sesión en tiempo de ejecución, ejemplo de NPM

```
awsRum.addSessionAttributes({
    applicationVersion: "1.3.8"
})
```

Agregar un atributo de sesión en la inicialización, ejemplo de script incrustado

La sección de código en negrita agrega el atributo de sesión.

```
<script>
    (function(n,i,v,r,s,c,u,x,z){...})(
        'cwr',
        '00000000-0000-0000-0000-000000000000',
        '1.0.0',
        'us-west-2',
        'https://client.rum.us-east-1.amazonaws.com/1.0.2/cwr.js',
        {
            sessionSampleRate:1,
            guestRoleArn:'arn:aws:iam::000000000000:role/RUM-Monitor-us-
west-2-000000000000-00xx-Unauth',
            identityPoolId:'us-west-2:00000000-0000-0000-0000-000000000000',
            endpoint:'https://dataplane.rum.us-west-2.amazonaws.com',
            telemetries:['errors','http','performance'],
            allowCookies:true,
            sessionAttributes: {
                applicationVersion: "1.3.8"
            }
        }
    );
</script>
```

Agregar un atributo de sesión en tiempo de ejecución, ejemplo de script incrustado

```
<script>
    function addSessionAttribute() {
        cwr('addSessionAttributes', {
            applicationVersion: "1.3.8"
        })
    }

</script>
```

Adición de atributos de página

Si configura atributos de página personalizados, se agregan a todos los eventos de la página actual. Los atributos de página se configuran durante la inicialización del cliente web de CloudWatch RUM o en tiempo de ejecución mediante el comando `recordPageView`.

Por ejemplo, puede agregar una plantilla de página como un atributo de página. A continuación, en la consola de CloudWatch RUM, puede filtrar los errores por plantillas de página para ver si hay un aumento en la tasa de errores asociado a una plantilla de página concreta de la aplicación.

Agregar un atributo de página en la inicialización, ejemplo de NPM

La sección de código en negrita agrega el atributo de página.

```
const awsRum: AwsRum = new AwsRum(
    APPLICATION_ID,
    APPLICATION_VERSION,
    APPLICATION_REGION,
    { disableAutoPageView:  true // optional }
);
awsRum.recordPageView({
    pageId:'/home',
    pageAttributes: {
      template: 'artStudio'
    }
});
const credentialProvider = new CustomCredentialProvider();
if(awsCreds) awsRum.setAwsCredentials(credentialProvider);
```

Agregar un atributo de página en tiempo de ejecución, ejemplo de NPM

```
awsRum.recordPageView({
    pageId: '/home',
```

```
    pageAttributes: {
        template: 'artStudio'
    }
});
```

Agregar un atributo de página en la inicialización, ejemplo de script incrustado

La sección de código en negrita agrega el atributo de página.

```
<script>
    (function(n,i,v,r,s,c,u,x,z){...})(
        'cwr',
        '00000000-0000-0000-0000-000000000000',
        '1.0.0',
        'us-west-2',
        'https://client.rum.us-east-1.amazonaws.com/1.0.2/cwr.js',
        {
            disableAutoPageView: true //optional
        }
    );
    cwr('recordPageView', {
       pageId: '/home',
       pageAttributes: {
           template: 'artStudio'
       }
    });
    const awsCreds = localStorage.getItem('customAwsCreds');
    if(awsCreds) cwr('setAwsCredentials', awsCreds)
</script>
```

Agregar un atributo de página en tiempo de ejecución, ejemplo de script incrustado

```
<script>
    function recordPageView() {
        cwr('recordPageView', {
            pageId: '/home',
            pageAttributes: {
                template: 'artStudio'
            }
        });
    }
</script>
```

Filtrar por atributos de metadatos en la consola

Para filtrar las visualizaciones en la consola de CloudWatch RUM con cualquier atributo de metadatos integrado o personalizado, utilice la barra de búsqueda. En la barra de búsqueda, puede especificar hasta 20 términos de filtro en forma de clave=valor para aplicarlos a las visualizaciones. Por ejemplo, para filtrar datos únicamente para el navegador Chrome, puede añadir el término de filtro browserName=Chrome.

De forma predeterminada, la consola RUM de CloudWatch recupera los 100 atributos, claves y valores más comunes para mostrarlos en el menú desplegable de la barra de búsqueda. Para agregar más atributos de metadatos como términos de filtro, ingresa la clave y el valor del atributo completos en la barra de búsqueda.

Un filtro puede incluir hasta 20 términos de filtro y puede guardar hasta 20 filtros por monitor de aplicación. Al guardar un filtro, se guarda en el menú desplegable Saved filters (Filtros guardados). También puede eliminar un filtro guardado.

Enviar eventos personalizados

CloudWatch RUM graba e ingiere los eventos que se enumeran en Información recopilada por el cliente web de CloudWatch RUM. Si usa la versión 1.12.0 o posterior del cliente web de CloudWatch RUM, puede definir, registrar y enviar eventos personalizados adicionales. Defina el nombre del tipo de evento y los datos que se van a enviar para cada tipo de evento que defina. La carga útil de cada evento personalizado puede tener un máximo de 6 KB.

Los eventos personalizados se incorporan solo si el monitor de aplicaciones tiene activados los eventos personalizados. Para actualizar los ajustes de configuración del monitor de aplicaciones, utilice la consola CloudWatch RUM o la API UpdateAppMonitor.

Después de habilitar los eventos personalizados y, a continuación, de definir y enviar eventos personalizados, puede buscarlos. Para buscarlos, utilice la pestaña Events (Eventos) de la consola de CloudWatch RUM. Busque mediante el tipo de evento.

Requisitos y sintaxis

Los eventos personalizados constan de un tipo de evento y detalles del evento. Los requisitos son los siguientes:

- Tipo de evento

  - Puede ser el type (tipo) o el name (nombre) del evento. Por ejemplo, el tipo de evento integrado de CloudWatch RUM denominado JsError tiene un tipo de evento de `com.amazon.rum.js_error_event`.
  - Debe tener entre 1 y 256 caracteres de longitud.
  - Puede ser una combinación de caracteres alfanuméricos, guiones bajos, guiones cortos y puntos.
- Detalles del evento
  - Contiene los datos reales que desee registrar en CloudWatch RUM.
  - Debe ser un objeto que conste de campos y valores.

Ejemplos de grabación de eventos personalizados

Hay dos formas de grabar eventos personalizados en el cliente web de CloudWatch RUM.

- Utilice la API `recordEvent` del cliente web de CloudWatch RUM.
- Utilice un complemento personalizado.

Enviar un evento personalizado mediante la API **`recordEvent`**, ejemplo de NPM

```
awsRum.recordEvent('my_custom_event', {
        location: 'IAD',
        current_url: 'amazonaws.com',
        user_interaction: {
            interaction_1 : "click",
            interaction_2 : "scroll"
        },
        visit_count:10
    }
)
```

Enviar un evento personalizado mediante la API **`recordEvent`**, ejemplo de script incrustado

```
cwr('recordEvent', {
    type: 'my_custom_event',
    data: {
        location: 'IAD',
        current_url: 'amazonaws.com',
        user_interaction: {
```

```
            interaction_1 : "click",
            interaction_2 : "scroll"
        },
        visit_count:10
    }
})
```

Ejemplo de envío de un evento personalizado mediante un complemento personalizado

```
// Example of a plugin that listens to a scroll event, and
// records a 'custom_scroll_event' that contains the timestamp of the event.
class MyCustomPlugin implements Plugin {
    // Initialize MyCustomPlugin.
    constructor() {
        this.enabled;
        this.context;
        this.id = 'custom_event_plugin';
    }
    // Load MyCustomPlugin.
    load(context) {
        this.context = context;
        this.enable();
    }
    // Turn on MyCustomPlugin.
    enable() {
        this.enabled = true;
        this.addEventHandler();
    }
    // Turn off MyCustomPlugin.
    disable() {
        this.enabled = false;
        this.removeEventHandler();
    }
    // Return MyCustomPlugin Id.
    getPluginId() {
        return this.id;
    }
    // Record custom event.
    record(data) {
        this.context.record('custom_scroll_event', data);
    }
    // EventHandler.
    private eventHandler = (scrollEvent: Event) => {
```

```
        this.record({timestamp: Date.now()})
    }
    // Attach an eventHandler to scroll event.
    private addEventHandler(): void {
        window.addEventListener('scroll', this.eventHandler);
    }
    // Detach eventHandler from scroll event.
    private removeEventHandler(): void {
        window.removeEventListender('scroll', this.eventHandler);
    }
}
```

Visualización del panel de CloudWatch RUM

CloudWatch RUM lo ayudará a recopilar datos de las sesiones de usuario sobre el rendimiento de la aplicación, incluidos los tiempos de carga de páginas, la puntuación de Apdex, los navegadores y dispositivos utilizados, la geolocalización de las sesiones de usuario y las sesiones con errores. Toda esta información se muestra en un panel.

Para visualizar el panel de RUM

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Señales de aplicación, RUM.

   En la pestaña Overview (Información general) se observa la información recopilada por uno de los monitores de aplicaciones que ha creado.

   La fila superior de paneles muestra la siguiente información de este monitor de aplicaciones:

   - Cantidad de cargas de página
   - Velocidad promedio de carga de página
   - Puntuación de Apdex
   - Estado de las alarmas asociadas con el monitor de aplicaciones

   La puntuación del índice de rendimiento de la aplicación (Apdex) indica el nivel de satisfacción de los usuarios finales. Las puntuaciones oscilan entre 0 (menos satisfechos) y 1 (más satisfechos). Las puntuaciones se basan únicamente en el rendimiento de la aplicación. No se les solicita a los usuarios que califiquen la aplicación. Para obtener más información sobre

las puntuaciones de Apdex, consulte Cómo establece CloudWatch RUM las puntuaciones de Apdex.

Muchos de estos paneles incluyen enlaces que puede utilizar para examinar los datos en mayor profundidad. Si elige cualquiera de estos enlaces, se mostrará una vista detallada con las pestañas Rendimiento, Errores, Solicitudes HTTP, Sesiones, Eventos, navegadores y dispositivos y Recorrido del usuario en la parte superior de la pantalla.

3. Para centrarse aún más, seleccione la List view (Vista de lista) y, a continuación, seleccione el nombre del monitor de aplicaciones en el que desea centrarse. Se mostrarán las siguientes pestañas del monitor de aplicaciones elegido.

   - En la pestaña Performance (Desempeño) se muestra información sobre el rendimiento de la página, incluidos los tiempos de carga, la información de la sesión, la información de la solicitud, los elementos vitales web y las cargas de la página en el tiempo. En esta vista se incluyen controles para alternar la vista entre Page loads (Cargas de páginas), Requests (Solicitudes) y Location (Ubicación).
   - En la pestaña Errores se muestra información sobre los errores de Javascript, incluido el mensaje de error que ven con más frecuencia los usuarios y los dispositivos y navegadores con más errores. Esta vista incluye un histograma de los errores y una vista de lista de los errores. Puede filtrar la lista de errores por usuario y detalles del evento. Seleccione un mensaje de error para ver más detalles.
   - La pestaña de solicitudes HTTP muestra la información de las solicitudes HTTP, incluida la URL de la solicitud con más errores y los dispositivos y navegadores con más errores. Esta pestaña incluye un histograma de las solicitudes, una vista de lista de las solicitudes y una vista de lista de los errores de red. Puede filtrar las listas por usuario y detalles del evento. Elija un código de respuesta o un mensaje de error para ver más detalles sobre la solicitud o el error de red, respectivamente.
   - La pestaña Sesiones muestra las métricas de la sesión. Esta pestaña incluye un histograma de los eventos de inicio de sesión y una vista de lista de las sesiones. Puede filtrar la lista de sesiones por tipo de evento, detalles de usuario y detalles del evento. Elija un SessionID para ver más detalles sobre una sesión.
   - La pestaña Eventos muestra un histograma de los eventos de RUM y una vista de lista de los eventos. Puede filtrar la lista de eventos por tipo de evento, detalles de usuario y detalles del evento. Elija un evento RUM para ver el evento sin procesar.

- En la pestaña de Browsers & Devices (Navegadores y dispositivos) se muestra información como el rendimiento y el uso de distintos navegadores y dispositivos para acceder a la aplicación. Esta vista incluye controles para alternar la vista entre Navegadores y Dispositivos.

  Si limita el alcance a un solo navegador, verá los datos desglosados por versión del navegador.
- En la pestaña de User Journey (Recorrido del usuario) se muestran las rutas que utilizan los clientes para navegar por la aplicación. Puede ver dónde ingresan los clientes en la aplicación y de qué página salen de la aplicación. También puedes ver las rutas que toman y el porcentaje de clientes que siguen esas rutas. Puede detenerse en un nodo para obtener más detalles sobre esa página. Puede elegir una única ruta para resaltar las conexiones y facilitar la visualización.

4. (Opcional) En cualquiera de las seis primeras pestañas, puede elegir el botón Páginas y seleccionar una página o un grupo de páginas de la lista. Esto reduce los datos que se muestran a una sola página o grupos de páginas de la aplicación. También puede marcar las páginas o grupos de páginas de la lista como favoritas.

Cómo establece CloudWatch RUM las puntuaciones de Apdex

Apdex (Application Performance Index) es un estándar abierto que define un método para informar, comparar y evaluar el tiempo de respuesta de las aplicaciones. Una puntuación de Apdex lo ayudará a comprender e identificar el impacto en el rendimiento de las aplicaciones a lo largo del tiempo.

La puntuación de Apdex indica que el nivel de satisfacción de los usuarios finales oscila entre 0 (menos satisfechos) y 1 (los más satisfechos). Las puntuaciones se basan únicamente en el rendimiento de la aplicación. No se les solicita a los usuarios que califiquen la aplicación.

Cada puntuación individual de Apdex corresponde a uno de los tres umbrales. De acuerdo con el umbral de Apdex y el tiempo de respuesta real de la aplicación, existen tres tipos de rendimiento, como se indica a continuación:

- Satisfecho: el tiempo de respuesta real de la aplicación es inferior o igual al umbral de Apdex. En el caso de CloudWatch RUM, este umbral es de 2000 ms o menos.
- Tolerable: el tiempo de respuesta real de la aplicación es mayor que el umbral de Apdex, pero inferior o igual a cuatro veces el umbral de Apdex. En CloudWatch RUM, este rango es de 2000 a 8000 ms.

- Frustrante: el tiempo real de respuesta de la aplicación es superior a cuatro veces el umbral de Apdex. En CloudWatch RUM, este rango supera los 8000 ms.

La puntuación total de 0 a 1 de Apdex se calcula con la siguiente fórmula:

```
(positive scores + tolerable scores/2)/total scores * 100
```

Métricas de CloudWatch que puede recopilar con CloudWatch RUM

En la tabla de esta sección, se muestran las métricas que se recopilen automáticamente con CloudWatch RUM. También puede ver estas métricas en la consola de CloudWatch. Para obtener más información, consulte Ver métricas disponibles.

De forma opcional, también puede enviar métricas ampliadas a CloudWatch o CloudWatch Evidently. Para obtener más información, consulte Métricas ampliadas.

Estas métricas se publican en el espacio de nombres de la métrica llamado `AWS/RUM`. Todas las métricas siguientes se publican con una dimensión de `application_name`. El valor de esta dimensión es el nombre del supervisor de aplicaciones. Algunas métricas también se publican con dimensiones adicionales, como se indica en la tabla.

Métrica	Unidad	Descripción
`HttpStatusCodeCount`	Recuento	El recuento de respuestas HTTP en la aplicación, según su código de estado de respuesta. Dimensiones adicionales: • `event_details.response.status` es el código de estado de respuesta, como 200, 400, 404, etc.

Métrica	Unidad	Descripción
		• `event_type` es el tipo de evento. Actualmente, el único valor posible para esta dimensión es `http`.
`Http4xxCount`	Recuento	El recuento de respuestas HTTP en la aplicación, según su código de estado de respuesta 4xx. Se calculan en función de los eventos de RUM `http_event` que dan como resultado códigos 4xx.
`Http5xxCount`	Recuento	El recuento de respuestas HTTP en la aplicación, según su código de estado de respuesta 5xx. Se calculan en función de los eventos de RUM `http_event` que dan como resultado códigos 5xx.
`JsErrorCount`	Recuento	El recuento de los eventos de error de JavaScript incorporados.

Métrica	Unidad	Descripción
`NavigationFrustratedCount`	Recuento	El recuento de los eventos de navegación con una `duration` superior al límite frustrante, que es de 8000 ms. Se realiza un seguimiento de la duración de los eventos de navegación en la métrica `PerformanceNavigationDuration` .
`NavigationSatisfiedCount`	Recuento	El recuento de los eventos de navegación con una `duration` menor que el objetivo de Apdex, que es de 2000 ms. Se realiza un seguimiento de la duración de los eventos de navegación en la métrica `PerformanceNavigationDuration` .

Métrica	Unidad	Descripción
`NavigationToleratedCount`	Recuento	El recuento de los eventos de navegación con una `duration` entre 2000 ms y 8000 ms. Se realiza un seguimiento de la duración de los eventos de navegación en la métrica `PerformanceNavigationDuration` .
`PageViewCount`	Recuento	El recuento de los eventos de visualización de páginas incorporados por el monitor de la aplicación. Esto se calcula contando los eventos de RUM `page_view_event` .

Métrica	Unidad	Descripción
`PerformanceResourceDuration`	Milisegundos	La `duration` de un evento de recursos. Dimensiones adicionales: • `event_details.file.type` es el tipo de archivo del evento de recursos, como una hoja de estilo, un documento, una imagen, un texto o una fuente. • `event_type` es el tipo de evento. Actualmente, el único valor posible para esta dimensión es `resource`.
`PerformanceNavigationDuration`	Milisegundos	La `duration` de un evento de navegación.

Métrica	Unidad	Descripción
`RumEventPayloadSize`	Bytes	El tamaño de cada evento incorporado por CloudWatch RUM. También puede utilizar la estadística `SampleCount` de esta métrica para controlar la cantidad de eventos que incorpora un supervisor de aplicaciones.
`SessionCount`	Recuento	El recuento de los eventos de inicio de sesión incorporados por el supervisor de aplicaciones. Es decir, la cantidad de nuevas sesiones iniciadas.
`WebVitalsCumulativeLayoutShift`	Ninguna	Realiza un seguimiento del valor de los eventos de cambio de diseño acumulativos.
`WebVitalsFirstInputDelay`	Milisegundos	Realiza un seguimiento del valor de los primeros eventos de retardo de entradas.

Métrica	Unidad	Descripción
`WebVitalsLargestContentfulPaint`	Milisegundos	Realiza un seguimiento del valor de los eventos de pintura con contenido más grandes.

Métricas personalizadas y ampliadas que puede enviar a CloudWatch y CloudWatch Evidently

De forma predeterminada, los monitores de aplicaciones de RUM envían métricas a CloudWatch. Estas métricas y dimensiones predeterminadas se muestran en Métricas de CloudWatch que puede recopilar con CloudWatch RUM.

También puede configurar un monitor de aplicaciones para que exporte métricas. El monitor de aplicaciones puede enviar métricas ampliadas, métricas personalizadas o ambas. Puede enviarlas a CloudWatch o a CloudWatch Evidently, o a ambos.

- Métricas personalizadas: las métricas personalizadas son métricas que el usuario define. Con las métricas personalizadas, puede usar cualquier nombre y espacio de nombres de métrica. Para derivar las métricas, puedes usar cualquier evento personalizado, evento integrado, atributo personalizado o atributo predeterminado.

  Puede enviar métricas personalizadas tanto a CloudWatch como a CloudWatch Evidently.
- Métricas ampliadas: permite enviar las métricas RUM predeterminadas de CloudWatch a CloudWatch Evidently para utilizarlas en los experimentos de Evidently. También puede enviar cualquiera de las métricas RUM de CloudWatch predeterminadas a CloudWatch con dimensiones adicionales. De este modo, estas métricas pueden ofrecerle una visión más detallada.

Temas

- Métricas personalizadas
- Métricas ampliadas

Métricas personalizadas

Para enviar métricas personalizadas, debe utilizar las API de AWS o AWS CLI en lugar de la consola. Para obtener más información sobre el uso de las API de AWS, consulte PutRumMetricsDestination y BatchCreateRumMetricDefinitions.

Un destino puede contener un máximo de 2000 definiciones de métricas ampliadas y personalizadas. Para cada métrica personalizada o ampliada que envíe a cada destino, cada combinación de nombre y valor de dimensión cuenta para este límite. Esto también cuenta como una métrica personalizada de CloudWatch para fijar precios.

El siguiente ejemplo muestra cómo crear una métrica personalizada derivada de un evento personalizado. Este es el ejemplo de evento personalizado que se utiliza:

```
cwr('recordEvent', {
    type: 'my_custom_event',
    data: {
        location: 'IAD',
        current_url: 'amazonaws.com',
        user_interaction: {
            interaction_1 : "click",
            interaction_2 : "scroll"
        },
        visit_count:10
    }
})
```

Dado este evento personalizado, puede crear una métrica personalizada que cuente el número de visitas a la URL `amazonaws.com` desde navegadores Chrome. La siguiente definición crea una métrica denominada `AmazonVisitsCount` en su cuenta, en el espacio de nombres `RUM/CustomMetrics/PageVisits`.

```
{
    "AppMonitorName":"customer-appMonitor-name",
    "Destination":"CloudWatch",
    "MetricDefinitions":[
        {
            "Name":"AmazonVisitsCount",
            "Namespace":"PageVisit",
            "ValueKey":"event_details.visit_count",
            "UnitLabel":"Count",
```

```
            "DimensionKeys":{
                "event_details.current_url": "URL"
            },
            "EventPattern":"{\"metadata\":{\"browserName\":[\"Chrome\"]},\"event_type
\":[\"my_custom_event\"],\"event_details\": {\"current_url\": [\"amazonaws.com\"]}}"
        }
    ]
}
```

Métricas ampliadas

Si configura métricas ampliadas, puede realizar una o ambas de las siguientes acciones:

- Envíe las métricas CloudWatch RUM predeterminadas a CloudWatch Evidently para utilizarlas en los experimentos de Evidently. Solo se pueden enviar a Evidently las métricas PerformanceNavigationDuration, PerformanceResourceDuration, WebVitalsCumulativeLayoutShift, WebVitalsFirstInputDelay y WebVitalsLargestContentfulPaint.
- Envíe cualquiera de las métricas predeterminadas de CloudWatch RUM a CloudWatch con dimensiones adicionales para que las métricas le ofrezcan una visión más detallada. Por ejemplo, puede ver las métricas específicas de un navegador determinado que utilicen los usuarios o las métricas de los usuarios de una ubicación geográfica específica.

Para obtener más información acerca de las métricas de CloudWatch RUM, consulte Métricas de CloudWatch que puede recopilar con CloudWatch RUM.

Un destino puede contener un máximo de 2000 definiciones de métricas ampliadas y personalizadas. Para cada métrica extendida que envíe a cada destino, cada combinación de nombre y valor de dimensión cuenta como una métrica extendida para este límite. Esto también cuenta como una métrica personalizada de CloudWatch para fijar precios.

Al enviar métricas ampliadas a CloudWatch, puede utilizar la consola de CloudWatch RUM para crear alarmas de CloudWatch en ellas.

Las métricas ampliadas se cobran como métricas personalizadas de CloudWatch. Para más información, consulte Precios de Amazon CloudWatch.

Las siguientes dimensiones son compatibles con las métricas ampliadas de todos los nombres de métricas que pueden enviar los monitores de aplicaciones. Estos nombres de métricas se enumeran en Métricas de CloudWatch que puede recopilar con CloudWatch RUM.

- `BrowserName`

  Ejemplos de valores de dimensión: `Chrome`, `Firefox`, `Chrome Headless`
- `CountryCode`: utiliza el formato ISO-3166, con códigos de dos letras.

  Ejemplos de valores de dimensión: `US`, `JP`, `DE`
- `DeviceType`

  Ejemplos de valores de dimensión: `desktop`, `mobile`, `tablet`, `embedded`
- `FileType`

  Ejemplos de valores de dimensión: `Image`, `Stylesheet`
- `OSName`

  Ejemplos de valores de dimensión: `Linux`, `Windows, iOS`, `Android`
- `PageId`

Configuración de métricas ampliadas con la consola

Para usar la consola para enviar métricas ampliadas a CloudWatch, siga estos pasos.

Para enviar métricas ampliadas a CloudWatch Evidently, debe utilizar las API de AWS o AWS CLI en lugar de la consola. Para obtener información sobre el uso de las API de AWS para enviar métricas ampliadas a CloudWatch o Evidently, consulte PutRumMetricsDestination y BatchCreateRumMetricDefinitions.

Usar la consola para configurar un monitor de aplicaciones y enviar las métricas ampliadas de RUM a CloudWatch

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Señales de aplicación, RUM.
3. Seleccione List view (Vista de lista) y, a continuación, seleccione el nombre del monitor de aplicaciones que vaya a enviar las métricas.
4. Elija la pestaña Configuration (Configuración) y, a continuación, elija RUM extended metrics (Métricas ampliadas de RUM).
5. Elija Send metrics (Enviar métricas).
6. Seleccione uno o más nombres de métricas para enviarlos con dimensiones adicionales.

7. Seleccione uno o más factores para utilizarlos como dimensiones para estas métricas. Al hacer sus elecciones, la cantidad de métricas ampliadas que crean sus elecciones se muestra en Number of extended metrics (Número de métricas ampliadas).

   Este número se calcula multiplicando el número de nombres de métricas elegidos por el número de dimensiones diferentes que se crean. Este número representa la cantidad de métricas personalizadas que se le cobran. Para obtener más información sobre los precios de CloudWatch, consulte Precios de Amazon CloudWatch.

   a. Para enviar una métrica con el ID de página como dimensión, elija Browse for page ID (Buscar el ID de página) y, a continuación, seleccione los ID de página que desee utilizar.
   b. Para enviar una métrica con el tipo de dispositivo como dimensión, elija Desktop devices (Dispositivos de escritorio) o Mobile and tablets (Móviles y tabletas).
   c. Para enviar una métrica con el sistema operativo como dimensión, seleccione uno o más sistemas operativos en Operating system (Sistema operativo).
   d. Para enviar una métrica con el tipo de navegador como dimensión, seleccione uno o más navegadores en Browsers (Navegadores).
   e. Para enviar una métrica con la ubicación geográfica como dimensión, seleccione una o más ubicaciones en Locations (Ubicaciones).

      Solo aparecerán en la lista para elegir las ubicaciones desde las que este monitor de aplicaciones haya informado sobre métricas.
8. Cuando haya terminado con las opciones, elija Send metrics (Enviar métricas).
9. (Opcional) En la lista Extended metrics (Métricas ampliadas), para crear una alarma que controle una de las métricas, elija Create alarm (Crear alarma) en la fila de esa métrica.

   Para obtener información general sobre las alarmas de CloudWatch, consulte Uso de las alarmas de Amazon CloudWatch. Para ver un tutorial sobre cómo configurar una alarma en una métrica ampliada de CloudWatch RUM, consulte Tutorial: crear una métrica ampliada y ponerle una alarma.

Dejar de enviar métricas ampliadas

Usar la consola para dejar de enviar métricas ampliadas

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Señales de aplicación, RUM.

3. Seleccione List view (Vista de lista) y, a continuación, seleccione el nombre del monitor de aplicaciones que vaya a enviar las métricas.
4. Elija la pestaña Configuration (Configuración) y, a continuación, elija RUM extended metrics (Métricas ampliadas de RUM).
5. Seleccione una o más combinaciones de nombre y dimensión de métrica para detener el envío. A continuación, elija Actions (Acciones), Delete (Eliminar).

Tutorial: crear una métrica ampliada y ponerle una alarma

Este tutorial muestra cómo configurar una métrica ampliada para enviarla a CloudWatch y, a continuación, cómo configurar una alarma en esa métrica. En este tutorial, creará una métrica que realice un seguimiento de los errores de JavaScript en el navegador Chrome.

Configuración de esta métrica ampliada y cómo establecer una alarma

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Señales de aplicación, RUM.
3. Seleccione List view (Vista de lista) y, a continuación, seleccione el nombre del monitor de aplicaciones que vaya a enviar la métrica.
4. Elija la pestaña Configuration (Configuración) y, a continuación, elija RUM extended metrics (Métricas ampliadas de RUM).
5. Elija Send metrics (Enviar métricas).
6. Seleccione JSErrorCount.
7. En Browsers (Navegadores), seleccione Chrome.

   Esta combinación de JSErrorCount y Chrome enviará una métrica ampliada a CloudWatch. La métrica solo cuenta los errores de JavaScript en las sesiones de usuario que utilicen el navegador Chrome. El nombre de la métrica será JSErrorCount y el nombre de la dimensión será Browser (Navegador).
8. Elija Send metrics (Enviar métricas).
9. En la lista Extended metrics (Métricas ampliadas), seleccione Create alarm (Crear alarma) en la fila que muestre JSErrorCount en Name (Nombre) y Chrome en BrowserName (Nombre del navegador).

10. En Specify metric and conditions (Especificar métrica y condiciones), confirme que los campos Metric name (Nombre de la métrica) y BrowserName estén rellenados previamente con los valores correctos.
11. En Statistic (Estadística), seleccione la estadística que desee utilizar para la alarma. Average (Promedio) es una buena opción para este tipo de métrica de recuento.
12. En Period (Período), seleccione 5 minutes (5 minutos).
13. En Condiciones, haga lo siguiente:

    - Elija Static (Estático).
    - Elija Greater (Mayor) para especificar que la alarma debe pasar al estado ALARM (ALARMA) cuando el número de errores sea superior al umbral que esté a punto de especificar.
    - En than… (que…), ingrese el número del umbral de la alarma. La alarma entra en estado ALARM cuando el número de errores en un período de 5 minutos supera este número.
14. (Opcional) De forma predeterminada, la alarma pasa al estado ALARM en cuanto el número de errores supera el umbral establecido durante un período de 5 minutos. Si lo desea, puede cambiarlo para que la alarma entre en estado ALARM solo si se supera este número durante más de un período de 5 minutos.

    Para ello, seleccione Additional configuration (Configuración adicional) y, a continuación, en Datapoints to alarm (Puntos de datos para la alarma), especifique cuántos períodos de 5 minutos deben tener el número de errores por encima del umbral para activar la alarma. Por ejemplo, puede seleccionar 2 de 2 para que la alarma se active solo cuando dos períodos consecutivos de 5 minutos superen el umbral, o 2 de 3 para que se active la alarma si 2 de los 3 períodos consecutivos de 5 minutos superan el umbral.

    Para obtener más información acerca de este tipo de evaluación de alarmas, consulte Evaluación de una alarma.
15. Elija Siguiente.
16. En Configure actions (Configurar acciones), especifique lo que debe suceder cuando la alarma entre en estado de alarma. Para recibir una notificación con Amazon SNS, haga lo siguiente:

    - Seleccione Agregar notificación.
    - Elija En alarma.
    - Seleccione un tema de SNS existente o cree uno nuevo. Si crea uno nuevo, especifique un nombre y agregue al menos una dirección de correo electrónico.
17. Elija Siguiente.

18. Ingrese un nombre y, si lo desea, una descripción de la alarma y elija Next (Siguiente).
19. Revise los detalles y seleccione Create alarm (Crear alarma).

Protección de datos y privacidad de datos con CloudWatch RUM

El modelo de responsabilidad compartida de AWS se aplica a la protección y la privacidad de datos en Amazon CloudWatch RUM. Tal como se describe en este modelo, AWS es responsable de proteger la infraestructura global que ejecuta toda la nube de AWS. Usted es responsable de mantener el control sobre el contenido alojado en esta infraestructura. Para obtener más información sobre la privacidad de datos, consulte Data Privacy FAQ (Preguntas frecuentes sobre la privacidad de datos). Para obtener información sobre la protección de datos en Europa, consulte la publicación de blog The AWS Shared Responsibility Model and GDPR (Modelo de responsabilidad compartida y RGPD) en el AWS Security Blog. Para obtener más recursos sobre el cumplimiento de los requisitos del RGPD, consulte el Centro de reglamento general de protección de datos (RGPD).

Amazon CloudWatch RUM genera un fragmento de código para que lo incruste en el código de su sitio web o aplicación web, en función de la entrada de datos de usuario final que desea recopilar. El cliente web, descargado y configurado mediante el fragmento de código, utiliza cookies (o tecnologías similares) para recopilar datos del usuario final. El uso de cookies (o tecnologías similares) está sujeto a las normas de privacidad de datos en determinadas jurisdicciones. Antes de utilizar Amazon CloudWatch RUM, le recomendamos ampliamente que evalúe sus obligaciones de cumplimiento conforme a la legislación aplicable, incluidos los requisitos legales aplicables para proporcionar avisos de privacidad legalmente adecuados y obtener el consentimiento necesario para el uso de cookies y el procesamiento (incluida la recopilación) de datos del usuario final. Para obtener más información sobre la forma en la que el cliente web utiliza las cookies (o tecnologías similares) y qué datos de usuario final recopila, consulte Información recopilada por el cliente web de CloudWatch RUM y Cookies del cliente web de CloudWatch RUM (o tecnologías similares).

Le recomendamos que nunca ingrese información confidencial que lo identifique, como números de cuenta de los usuarios finales, correos electrónicos u otro tipo de información personal, en los campos de formato libre. Cualquier dato que ingrese en Amazon CloudWatch RUM o en otros servicios se puede incluir en los registros de diagnóstico.

Cookies del cliente web de CloudWatch RUM (o tecnologías similares)

El cliente web de CloudWatch RUM recopila determinados datos sobre las sesiones de usuario de forma predeterminada. Puede optar por habilitar las cookies para que el cliente web también recopile

un ID de usuario y un ID de sesión cada vez que se carga la página. El ID de usuario se genera de forma aleatoria mediante RUM.

Si estas cookies están habilitadas, RUM podrá mostrar los siguientes tipos de datos cuando usted vea el panel de RUM de este monitor de aplicaciones.

- Datos agregados basados en ID de usuario, como el número de usuarios únicos y el número de usuarios diferentes que han experimentado un error.
- Datos agregados basados en ID de sesión, como el número de sesiones y el número de sesiones que han experimentado un error.
- El user journey (recorrido del usuario), que es la secuencia de páginas que incluye cada sesión de usuario de muestra.

> **Important**
>
> Si no habilita estas cookies (o tecnologías similares), el cliente web seguirá registrando cierta información sobre las sesiones del usuario final, como el tipo/versión del navegador, el tipo/versión del sistema operativo, el tipo de dispositivo, entre otros. Se recopilan para proporcionar información agregada específica de la página, como web vitals, vistas de página y páginas que han experimentado errores. Para obtener más información acerca de los datos registrados, consulte Información recopilada por el cliente web de CloudWatch RUM.

Información recopilada por el cliente web de CloudWatch RUM

En esta sección se documenta el esquema de PutRumEvents, que define la estructura de los datos que puede recopilar de las sesiones de usuario mediante CloudWatch RUM.

Una solicitud de PutRumEvents envía una estructura de datos con los siguientes campos a CloudWatch RUM.

- El ID de este lote de eventos RUM
- Detalles del monitor de aplicaciones, que incluye lo siguiente:
  - ID del monitor de aplicaciones
  - versión de la aplicación supervisada
- Datos del usuario, que incluye lo siguiente. Esto se recopila solo si el monitor de aplicaciones tiene las cookies habilitadas.

  - ID de usuario generado por el cliente web
  - ID de sesión
- La matriz de eventos de RUM en este lote.

Esquema de eventos de RUM

La estructura de cada evento de RUM incluye los siguientes campos.

- El ID del evento
- Una marca temporal
- El tipo de evento
- El agente de usuario
- Metadatos
- Detalles del evento de RUM

Metadatos del evento de RUM

Los metadatos incluyen metadatos de página, metadatos de agente de usuario, metadatos de geolocalización y metadatos de dominio.

Metadatos de la página

Los metadatos de la página incluyen lo siguiente:

- ID de página
- Título de página
- ID de página principal. Esto se recopila solo si el monitor de aplicaciones tiene las cookies habilitadas.
- Profundidad de interacción: se recopila solo si el monitor de aplicaciones tiene las cookies habilitadas.
- Etiquetas de página: puede añadir etiquetas a los eventos de la página para agrupar las páginas. Para obtener más información, consulte Uso de grupos de páginas.

Metadatos del agente de usuario

Los metadatos del agente de usuario incluyen lo siguiente:

- Idioma del navegador
- Nombre del navegador
- Versión del navegador
- Nombre del sistema operativo
- Versión del sistema operativo
- Tipo de dispositivo
- Tipos de plataformas

Metadatos de geolocalización

Los metadatos de geolocalización incluyen lo siguiente:

- Código de país
- Código de subdivisión

Metadatos del dominio

Los metadatos del dominio incluyen el dominio URL.

Detalles del evento de RUM

Los detalles de un evento cumplen con uno de los siguientes tipos de esquemas, según el tipo de evento.

Evento de inicio de sesión

Este evento no contiene campos. Esto se recopila solo si el monitor de aplicaciones tiene las cookies habilitadas.

Esquema de vista de página

Un evento de Vista de página contiene las siguientes propiedades. Puede desactivar la colección de vistas de página si configura el cliente web. Para obtener más información, consulte la documentación del cliente web de CloudWatch RUM.

Nombre	Tipo	Descripción
ID de página	Cadena	ID que representa de forma exclusiva esta página dentro de la aplicación. De forma predeterminada, esta es la ruta URL.
ID de la página principal	Cadena	Id. de la página en la que se encontraba el usuario cuando navegó a la página actual. Esto se recopila solo si el monitor de aplicaciones tiene las cookies habilitadas.
Profundidad de interacción	Cadena	Esto se recopila solo si el monitor de aplicaciones tiene las cookies habilitadas.

Esquema de errores de JavaScript

Los eventos de error de JavaScript generados por el agente contienen las siguientes propiedades. El cliente web recopila estos eventos solo si ha seleccionado recopilar los errores de telemetría.

Nombre	Tipo	Descripción
Tipo de error	Cadena	El nombre del error, si existe alguno. Para obtener más información, consulte Error.prototype.name. Es posible que algunos navegadores no admitan tipos de errores.
Mensaje de error	Cadena	Mensaje de error. Para obtener más información, consulte Error.prototype.message. Si el campo de error no existe, este es el mensaje del evento de error. Para obtener más información, consulte ErrorEvent. Es posible que los mensajes de error no sean coherentes según los distintos navegadores.
Seguimiento de pila	Cadena	El seguimiento de pila del error, si existe, se trunca a 150 caracteres. Para obtener más información, consulte Error.prototype.stack.

Nombre	Tipo	Descripción
		Es posible que algunos navegadores no admitan seguimientos de pila.

Esquema de eventos DOM

Los eventos del modelo de objeto de documento (DOM) generados por el agente contienen las siguientes propiedades. Estos eventos no se recopilan de forma predeterminada. Solo se recopilan si activa la telemetría de interacciones. Para obtener más información, consulte la documentación del cliente web de CloudWatch RUM.

Nombre	Tipo	Descripción
Evento	Cadena	El tipo de evento DOM, como hacer clic, deslizar o desplazar el ratón. Para obtener más información, consulte Event reference (Referencia de eventos).
Elemento	Cadena	El tipo de elemento DOM
(ID del elemento)	Cadena	Si el elemento que generó el evento tiene un ID, esta propiedad almacena ese ID. Para obtener más informaci ón, consulte Element.id.
Localizador CSS	Cadena	El localizador CSS utilizado para identificar el elemento DOM.
ID de interacción	Cadena	Un identificador único para la interacción entre el usuario y la interfaz de usuario.

Esquema de eventos de navegación

Los eventos de navegación se recopilan solo si el monitor de la aplicación tiene activada la telemetría de rendimiento.

Los eventos de navegación utilizan API de Navigation timing Level 1 (Temporización de navegación Nivel 1) y Navigation timing Level 2 (Temporización de navegación Nivel 2). Las API de nivel 2 no son compatibles con todos los navegadores, por lo que estos campos más recientes son opcionales.

> **Note**
>
> Las métricas de marca temporal se basan en DOMHighResTimestamp. Con las API de nivel 2, todas las temporizaciones están relacionadas a `startTime` de forma predeterminada. Sin embargo, para el nivel 1, la métrica `navigationStart` se descuenta de las métricas de marca temporal para obtener valores relativos. Todos los valores de marca temporal se expresan en milisegundos.

Los eventos de navegación contienen las siguientes propiedades.

Nombre	Tipo	Descripción	Notas
initiatorType	Cadena	Representa el tipo de recurso que inició el evento de rendimiento.	Valor: “navegación” Nivel 1: “navegación” Nivel 2: entryData .initiatorType
navigationType	Cadena	Representa el tipo de navegación. Este atributo no es obligatorio.	Valor: este valor tiene que ser uno de los siguientes: • `navigate` es una navegación que se inicia al elegir un enlace, ingresar una URL

Nombre	Tipo	Descripción	Notas
			en la barra de direcciones de un navegador, enviar formularios o inicializar mediante una operación de script que no sea `reload` ni `back_forward`. • `reload` es una navegación a través de la operación de recarga del navegador o `location.reload()`. • `back_forward` es una navegación a través

Nombre	Tipo	Descripción	Notas
			de la operación transversal del historial del navegador. • `prerender` es una navegación iniciada por un hint de prerender. Para obtener más información, consulte Prerender.
startTime	Número	Indica cuándo se desencadena el evento.	Valor: 0 Nivel 1: entryData.navigationStart - entryData.navigationStart Nivel 2: entryData.startTime

Nombre	Tipo	Descripción	Notas
unloadEventStart	Número	Indica la hora en la que el documento previo de la ventana comenzó a descargarse tras el lanzamiento del evento de `unload` (descarga).	Valor: si no hay un documento previo o si el documento previo o uno de los redirecciones necesarios no son del mismo origen, el valor devuelto será 0. Nivel 1: `entryData.unloadEventStart > 0 ? entryData.unloadEventStart - entryData.navigationStart : 0` Nivel 2: entryData

Nombre	Tipo	Descripción	Notas
			.unloadEventStart
promptForUnload	Número	El tiempo que tarda en descargar el documento. En otras palabras, el tiempo entre `unloadEventStart` y `unloadEventEnd`. `UnloadEventEnd` representa el momento en milisegundos en el que el controlador de eventos de descarga finaliza.	Valor: si no hay un documento previo o si el documento previo o uno de los redirecciones necesarios no son del mismo origen, el valor devuelto será 0. Nivel 1: entryData.unloadEventEnd - entryData.unloadEventStart Nivel 2: entryData.unloadEventEnd - entryData.unloadEventStart

Nombre	Tipo	Descripción	Notas
redirectCount	Número	Número que representa la cantidad de redirecciones desde la última navegación sin redirección en el contexto de búsqueda actual. Este atributo no es obligatorio.	Valor: si no hay redirección o si hay alguna redirección que no sea del mismo origen que el documento de destino, el valor devuelto será 0. Nivel 1: no disponible Nivel 2: entryData.redirectCount

Nombre	Tipo	Descripción	Notas
redirectStart	Número	El momento en que se inicia la primera redirección HTTP.	Valor: si no hay redirección o si hay alguna redirección que no sea del mismo origen que el documento de destino, el valor devuelto será 0. Nivel 1: `entryData.redirectStart > 0 ? entryData.redirectStart - entryData.navigationStart : 0` Nivel 2: entryData.redirectStart

Nombre	Tipo	Descripción	Notas
redirectTime	Número	El tiempo que tarda la redirección HTTP. Esta es la diferencia entre `redirectStart` y `redirectEnd` .	Nivel 1: entryData .redirectEnd - entryData .redirectStart Nivel 2: entryData .redirectEnd - entryData .redirectStart

Nombre	Tipo	Descripción	Notas
workerStart	Número	Esta es una propiedad de la interfaz `PerformanceResourceTiming` . Marca el comienzo de la operación de subprocesos de trabajo. Este atributo no es obligatorio.	Valor: si ya se está ejecutando un subproceso de Service Worker o inmediatamente antes de iniciar el subproceso de Service Worker, esta propiedad devuelve la hora inmediatamente antes de despachar `FetchEvent` . Devuelve 0 si un Service Worker no intercepta el recurso. Nivel 1: no disponible Nivel 2: entryData .workerStart

Nombre	Tipo	Descripción	Notas
workerTime	Número	Si un Service Worker intercepta el recurso,esto devuelve el tiempo necesario para la operación de subprocesos de trabajo. Este atributo no es obligatorio.	Nivel 1: no disponible Nivel 2: `entryData.workerStart > 0 ? entryData.fetchStart - entryData.workerStart : 0`
fetchStart	Número	La hora en la que el navegador está listo para recuperar el documento mediante una solicitud HTTP. Esto es antes de comprobar la caché de cualquier aplicación.	Nivel 1: `: entryData.fetchStart > 0 ? entryData.fetchStart - entryData.navigationStart : 0` Nivel 2: entrydata .fetchStart

Nombre	Tipo	Descripción	Notas
domainLookup	Número	Hora en la que se inicia la búsqueda de dominios.	Valor: si se utiliza una conexión persistente o si la información se almacena en una caché o en un recurso local, el valor será el mismo que el de `fetchStart`. Nivel 1: `entryData.domainLookupStart > 0 ? entryData.domainLookupStart - entryData.navigationStart : 0` Nivel 2: entryData.domainLookupStart

Nombre	Tipo	Descripción	Notas
DNS	Número	El tiempo necesario para la búsqueda de dominios.	Valor: si los recursos y los registros DNS se almacenan en la caché, el valor esperado será 0. Nivel 1: entryData .domainLookupEnd - entryData .domainLookupStart Nivel 2: entryData .domainLookupEnd - entryData .domainLookupStart
nextHopProtocol	Cadena	Cadena que representa el protocolo de red que se utiliza para obtener el recurso. Este atributo no es obligatorio.	Nivel 1: no disponible Nivel 2: entryData .nextHopProtocol

Nombre	Tipo	Descripción	Notas
connectStart	Número	Tiempo inmediatamente antes de que el agente de usuario comience a establecer la conexión con el servidor para recuperar el documento.	Valor: si se utiliza una conexión persistente RFC2616 o si el documento actual se recupera desde cachés de aplicaciones o recursos locales relevantes, este atributo devuelve el valor de `domainLookupEnd` . Nivel 1: `entryData.connectStart > 0 ? entryData.connectStart - entryData.navigationStart : 0`

Nombre	Tipo	Descripción	Notas
			Nivel 2: entryData .connectStart
connect	Número	Mide el tiempo necesario para establecer las conexiones de transporte o para realizar la autenticación SSL. También incluye el tiempo bloqueado que se tarda cuando hay demasiadas solicitudes simultáneas emitidas por el navegador.	Nivel 1: entryData .connectEnd - entryData .connectStart Nivel 2: entryData .connectEnd - entryData .connectStart
secureConnectionStart	Número	Si el esquema de URL de la página actual es “https”, este atributo devuelve el tiempo inmediatamente antes de que el agente de usuario inicie el proceso de enlace para proteger la conexión actual. Si no se utiliza HTTPS, devuelve 0. Para obtener más información sobre los esquemas de URL, consulte.URL representation (Representación de URL).	Fórmula: entryData .secureConnectionStart

Nombre	Tipo	Descripción	Notas
tlsTime	Número	El tiempo que lleva completar un enlace SSL.	Nivel 1: `entryData.secureConnectionStart > 0 ? entryData.connectEnd - entryData.secureConnectionStart : 0` Nivel 2: `entryData.secureConnectionStart > 0 ? entryData.connectEnd - entryData.secureConnectionStart : 0`

Nombre	Tipo	Descripción	Notas
requestStart	Número	Tiempo inmediatamente antes de que el agente de usuario comience a solicitar el recurso del servidor, de las memorias caché de aplicaciones pertinentes o de recursos locales.	Nivel 1: `: entryData.requestStart > 0 ? entryData.requestStart - entryData.navigationStart : 0` Nivel 2: entryData.requestStart
timeToFirstByte	Número	Tiempo que tarda en recibir el primer byte de información después de haber realizado una solicitud. Esta vez es relativa a la `startTime` .	Nivel 1: entryData.responseStart - entryData.requestStart Nivel 2: entryData.responseStart - entryData.requestStart

Nombre	Tipo	Descripción	Notas
responseStart	Número	Hora inmediatamente después de que el analizador HTTP del agente de usuario recibe el primer byte de la respuesta de las memorias caché de aplicaciones pertinentes, de recursos locales o del servidor.	Nivel 1: `entryData.responseStart > 0 ? entryData.responseStart - entryData.navigationStart : 0` Nivel 2: entryData.responseStart

Nombre	Tipo	Descripción	Notas
responseTime	Cadena	Tiempo que tarda en recibir una respuesta completa en forma de bytes de las memorias caché de aplicaciones pertinentes, de recursos locales o del servidor.	Nivel 1: `entryData.responseStart > 0 ? entryData.responseEnd - entryData.responseStart : 0` Nivel 2: `entryData.responseStart > 0 ? entryData.responseEnd - entryData.responseStart : 0`

Nombre	Tipo	Descripción	Notas
domInteractive	Número	El momento en el que el analizador finaliza su trabajo en el documento principal y se construye el DOM en HTML. En este momento, el `Document.readyState` cambia a "interactivo" y se genera el evento `readystatechange` correspondiente.	Nivel 1: `entryData.domInteractive > 0 ? entryData.domInteractive - entryData.navigationStart : 0` Nivel 2: entryData.domInteractive

Nombre	Tipo	Descripción	Notas
domContentLoadedEventStart	Número	Representa el valor de tiempo igual al tiempo inmediatamente anterior a que el agente de usuario active el evento DOMContentLoaded en el documento actual. El evento DOMContentLoaded se activará cuando el documento HTML inicial se haya cargado y analizado por completo. En ese momento, el documento HTML principal habrá terminado de analizar, el navegador comenzará a construir el render tree y los subrecursos aún deberán cargarse. Esto no esperará a que finalicen de cargar las hojas de estilo, las imágenes y los submarcos.	Nivel 1: `entryData.domContentLoadedEventStart > 0 ? entryData.domContentLoadedEventStart - entryData.navigationStart : 0` Nivel 2: entryData.domContentLoadedEventStart

Nombre	Tipo	Descripción	Notas
domContentLoaded	Número	Esta hora de inicio y fin de la construcción del render tree está marcada por `domContentLoadedEventStart` y `domContentLoadedEventEnd`. Permite que CloudWatch RUM realice un seguimiento de la ejecución. Esta propiedad es la diferencia entre `domContentLoadedStart` y `domContentLoadedEnd`. Durante este tiempo, DOM y CSSOM están listos. Esta propiedad espera la ejecución del script, excepto los scripts asíncronos y creados dinámicamente. Si los scripts dependen de hojas de estilo, también `domContentLoaded` espera a las hojas de estilo. No espera a las imágenes. **Note** Los valores reales de `domContentLoadedStart` y `domContentLoadedEnd` se aproximan a `domContentLoaded` en el panel Red de Google Chrome. Indica el tiempo de construcción del render tree HTML DOM + CSSOM desde el principio del proceso de carga de la página. En el caso de las métricas de navegación, el valor `domContentLoaded` represent a la diferencia entre los valores inicial y final, que es el tiempo necesario para descargar los subrecursos y la construcción del render tree únicament e.	Nivel 2: entryData.domContentLoadedEventEnd - entryData.domContentLoadedEventStart Nivel 2: entryData.domContentLoadedEventEnd - entryData.domContentLoadedEventStart

Nombre	Tipo	Descripción	Notas
domComplete	Número	La hora inmediatamente anterior a que el navegador establezca la disponibilidad del documento actual para que se complete. En este momento, se ha completado la carga de subrecursos, como las imágenes. Esto incluye el tiempo que tarda en descargar contenido bloqueado como CSS y JavaScript sincrónico. Esto se aproxima a `loadTime` en el panel de Red de Google Chrome.	Nivel 1: `entryData.domComplete > 0 ? entryData.domComplete - entryData.navigationStart : 0` Nivel 2: entryData.domComplete
domProcessingTime	Número	Tiempo total entre la respuesta y el inicio del evento de carga.	Nivel 1: entryData.loadEventStart - entryData.responseEnd Nivel 2: entryData.loadEventStart - entryData.responseEnd

Nombre	Tipo	Descripción	Notas
loadEvent Start	Número	La hora inmediatamente anterior a que se active el evento de `load` (carga) del documento actual.	Nivel 1: `entryData .loadEvent tStart > 0 ? entryData .loadEvent tStart - entryData .navigati onStart : 0` Nivel 2: entryData .loadEvent tStart
loadEvent Time	Número	La diferencia entre `loadEventStart` y `loadEventEnd` . Durante este tiempo se activarán funciones o lógica adicionales que aguardan este evento de carga.	Nivel 1: entryData .loadEvent tEnd - entryData .loadEvent tStart Nivel 2: entryData .loadEvent tEnd - entryData .loadEvent tStart

Nombre	Tipo	Descripción	Notas
Duración	Cadena	La duración es el tiempo total de carga de la página. Registra el tiempo de descarga de la página principal y todos sus subrecurs os sincrónicos, así como el tiempo para renderizar la página. Los recursos asincróni cos, como los scripts, continúan descargán dose más adelante. Esta es la diferencia entre el `loadEventEnd` y `startTime` propiedad es.	Nivel 1: entryData .loadEven tEnd - entryData .navigati onStart Nivel 2: entryData .duration
headerSize	Número	Devuelve la diferencia entre `transferSize` y `encodedBodySize` . Este atributo no es obligatorio.	Nivel 1: no disponible Nivel 2: entryData .transferSize - entryData .encodedB odySize Nivel 2: entryData .transferSize - entryData .encodedB odySize

Nombre	Tipo	Descripción	Notas
compressionRatio	Número	La proporción de `encodedBodySize` y `decodedBodySize`. El valor de `encodedBodySize` es el tamaño comprimido del recurso, sin incluir los encabezados HTTP. El valor de `decodedBodySize` es el tamaño descomprimido del recurso, sin incluir los encabezados HTTP. Este atributo no es obligatorio.	Nivel 1:No disponible. Nivel 2: `entryData.encodedBodySize > 0 ? entryData.decodedBodySize / entryData.encodedBodySize : 0`
navigationTimingLevel	Número	La versión de la API de temporización de navegación.	Valor: 1 o 2

Esquema de eventos de recursos

Los eventos de recursos se recopilan solo si el monitor de aplicaciones tiene activada la telemetría de rendimiento.

Las métricas de marca temporal se basan en The DOMHighResTimeStamp typedef. Con las API de nivel 2, todas las temporizaciones son relativas a la `startTime` de forma predeterminada. Sin embargo, para las API de nivel 1, la métrica de `navigationStart` se resta de las métricas de marca temporal para obtener valores relativos. Todos los valores de marca temporal se expresan en milisegundos.

Los eventos de recursos que genera el agente contienen las siguientes propiedades.

Nombre	Tipo	Descripción	Notas
targetUrl	Cadena	Devuelve la URL del recurso.	Fórmula: entryData .name
initiatorType	Cadena	Representa el tipo de recurso que inició el evento de recurso de rendimiento.	Valor: “recurso” Fórmula: entryData .initiatorType
Duración	Cadena	Devuelve la diferencia entre las propiedades de `responseEnd` y `startTime` . Este atributo no es obligatorio.	Fórmula: entryData .duration
transferSize	Número	Devuelve el tamaño (en octetos) del recurso recuperado, incluidos los campos de encabezado de respuesta y el cuerpo de la carga de la respuesta. Este atributo no es obligatorio.	Fórmula: entryData .transferSize
fileType	Cadena	Extensiones derivadas del patrón de URL de destino.	

Esquema del evento Largest Contentful Paint

Los eventos Largest Contentful Paint incluyen las siguientes propiedades.

Estos eventos se recopilan solo si el monitor de aplicaciones tiene activada la telemetría de rendimiento.

Nombre	Descripción		
Valor	Para obtener más información,		

Nombre	Descripción		
	consulte Web vitals.		

Primer evento de retardo de entrada

Los primeros eventos de retardo de entrada contienen las siguientes propiedades.

Estos eventos se recopilan solo si el monitor de aplicaciones tiene activada la telemetría de rendimiento.

Nombre	Descripción		
Valor	Para obtener más información, consulte Web vitals.		

Evento de cambio de diseño acumulativo

Los eventos de cambio de diseño acumulativos contienen las siguientes propiedades.

Estos eventos se recopilan solo si el monitor de aplicaciones tiene activada la telemetría de rendimiento.

Nombre	Descripción		
Valor	Para obtener más información, consulte Web Vitals.		

Evento HTTP

Los eventos HTTP pueden contener las siguientes propiedades. Contendrá un campo `Response` (respuesta) o `Error` (error), pero no ambos.

Estos eventos se recopilan solo si el monitor de aplicaciones tiene activada la telemetría HTTP.

Nombre	Descripción
Solicitud	La solicitud incluye lo siguiente: • El campo `Method`, que puede tener valores como GET, POST, etc. • La URL
Respuesta	La respuesta incluye lo que se detalla a continuación: • Estado, como 2xx, 4xx o 5xx • Texto de estado
Error	El campo de error incluye lo que se detalla a continuación: • Tipo • Mensaje • Nombre de archivo • Número de fila • Número de columna • Seguimiento de pila

Esquema de eventos de seguimiento de X-Ray

Estos eventos se recopilan solo si el monitor de la aplicación tiene activado el seguimiento de X-Ray.

Para obtener información sobre esquemas de eventos de seguimiento de X-Ray, consulte Documentos de segmentos de AWS X-Ray.

Tiempo de cambio de ruta para aplicaciones de una sola página

En una aplicación tradicional de varias páginas, cuando un usuario solicita que se cargue contenido nuevo, en realidad está solicitando una nueva página HTML al servidor. Como resultado, el cliente

web CloudWatch RUM captura los tiempos de carga mediante las métricas de API de desempeño habituales.

Sin embargo, las aplicaciones web de una sola página utilizan JavaScript y Ajax para actualizar la interfaz sin cargar una página nueva desde el servidor. La API de tiempo del navegador no registra las actualizaciones de una sola página, sino que utilizan el tiempo de cambio de ruta.

CloudWatch RUM admite la supervisión de las cargas de páginas completas desde el servidor y de las actualizaciones de una sola página con las siguientes diferencias:

- En lo que respecta al tiempo de cambio de ruta, no hay métricas proporcionadas por el navegador, como `tlsTime`, `timeToFirstByte` y así sucesivamente.
- En lo relativo al tiempo de cambio de ruta, el campo `initiatorType` será `route_change`.

El cliente web de CloudWatch RUM escucha las interacciones de los usuarios que pueden provocar un cambio de ruta y, cuando se registra una interacción así, el cliente web registra una marca de tiempo. El tiempo de cambio de ruta se iniciará entonces si se cumplen las siguientes condiciones:

- Se ha utilizado una API de historial del navegador (excepto los botones de avance y retroceso del navegador) para realizar el cambio de ruta.
- La diferencia entre la hora de detección del cambio de ruta y la última marca de tiempo de interacción del usuario es inferior a 1000 ms. Esto evita el sesgo de datos.

Luego, una vez da comienzo el tiempo de cambio de ruta, ese tiempo se completa si no hay solicitudes AJAX ni mutaciones DOM en curso. Seguidamente, se utilizará la marca de tiempo de la última actividad completada como marca de tiempo de finalización.

El tiempo de cambio de ruta expirará si hay solicitudes AJAX en curso o mutaciones de DOM durante más de 10 segundos (de forma predeterminada). En este caso, el cliente web de CloudWatch RUM ya no registrará el tiempo de este cambio de ruta.

Como resultado, la duración de un evento de cambio de ruta se calcula de la siguiente manera:

```
(time of latest completed activity) - (latest user interaction timestamp)
```

Administre las aplicaciones que utilizan CloudWatch RUM

Siga los pasos de estas secciones para administrar el uso de CloudWatch RUM por parte de sus aplicaciones.

¿Cómo encuentro un fragmento de código que ya he generado?

Para encontrar un fragmento de código de CloudWatch RUM que ya ha generado para una aplicación, siga estos pasos.

Para encontrar un fragmento de código que ya ha generado

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Señales de aplicación, RUM.
3. Elija List view (Vista de lista).
4. Elija View JavaScript (Ver JavaScript) junto al nombre del monitor de aplicaciones.
5. En el panel JavaScript Snippet (Fragmento de JavaScript), elija Copy to clipboard (Copiar al portapapeles).

Edición de la aplicación

Para cambiar la configuración de un monitor de aplicaciones, siga estos pasos. Puede cambiar cualquier configuración excepto el nombre del monitor de la aplicación.

Para editar cómo la aplicación usa CloudWatch RUM

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Señales de aplicación, RUM.
3. Elija List view (Vista de lista).
4. Elija el botón situado junto al nombre de la aplicación y luego, Actions (Acciones), Edit (Editar).
5. Cambie cualquier configuración excepto el nombre de la aplicación. Para obtener más información acerca de la configuración, consulte Paso 2: Cree un monitor de aplicaciones.
6. Cuando termine, elija Guardar.

   Al cambiar la configuración, se cambia el fragmento de código. Ahora debe pegar el fragmento de código actualizado en la aplicación.

7. Una vez que se haya creado el fragmento de código JavaScript, elija Copy to clipboard (Copiar al portapapeles) o Download (Descargar) y luego, Done (Listo).

   Para empezar a supervisar con la nueva configuración, inserte el fragmento de código en la aplicación. Inserte el fragmento de código dentro del elemento `<head>` de la aplicación, antes del elemento `<body>` o cualquier otra etiqueta `<script>`.

Dejar de utilizar CloudWatch RUM o eliminar un monitor de aplicaciones

Para dejar de utilizar CloudWatch RUM con una aplicación, elimine el fragmento de código que RUM generó a partir del código de la aplicación.

Si desea eliminar un monitor de aplicación RUM, siga estos pasos.

Para eliminar un monitor de aplicaciones

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Señales de aplicación, RUM.
3. Elija List view (Vista de lista).
4. Elija el botón situado junto al nombre de la aplicación y luego, Actions (Acciones), Delete (Eliminar).
5. En el casillero de confirmación, ingrese **`Delete`** (Eliminar) y luego, elija Delete (Eliminar).
6. Si aún no lo ha hecho, elimine el fragmento de código de CloudWatch RUM del código de la aplicación.

Cuotas de CloudWatch RUM

CloudWatch RUM tiene las siguientes cuotas.

Recurso	Cuota predeterminada
Monitores de aplicaciones	20 por cuenta Puede solicitar un aumento de cuota.
Tasa de ingesta de RUM	50 solicitudes PutRumEvents por segundo (TPS). Puede solicitar un aumento de cuota.

Recurso	Cuota predeterminada

Solución de problemas de CloudWatch RUM

En esta sección se incluyen sugerencias para solucionar problemas de CloudWatch RUM.

No hay datos para mi aplicación

En primer lugar, asegúrese de que el fragmento de código se ha insertado en la aplicación de forma correcta. Para obtener más información, consulte Paso 4: Inserte el fragmento de código en la aplicación.

Si ese no es el problema, tal vez aún no hay tráfico a su aplicación. Genere algo de tráfico al acceder a su aplicación de la misma manera que lo haría un usuario.

Los datos han dejado de registrarse para mi aplicación

Es posible que la aplicación se haya actualizado y que ya no contenga un fragmento de código de CloudWatch RUM. Verifique el código de la aplicación.

Otra posibilidad es que alguien haya actualizado el fragmento de código pero no haya insertado el fragmento actualizado en la aplicación. Siga las instrucciones que aparecen en ¿Cómo encuentro un fragmento de código que ya he generado? y busque el fragmento de código correcto actual y compárelo con el fragmento de código que se pega en la aplicación.

Monitoreo de la red

En los temas de esta sección se describen las capacidades de supervisión de redes e Internet de CloudWatch que ofrecen Amazon CloudWatch Internet Monitor y Amazon CloudWatch Network Monitor. Estos servicios lo ayudan a obtener una visibilidad operativa del rendimiento y la disponibilidad de las aplicaciones alojadas en AWS.

- Internet Monitor utiliza los datos de conectividad que AWS recopila de su huella en la red global para calcular una línea base de rendimiento y disponibilidad del tráfico de Internet. Puede obtener una vista global de los patrones de tráfico y los eventos de estado y analizar fácilmente la información sobre los eventos. También puede recibir alertas sobre eventos del estado de Internet que afecten a los clientes de sus aplicaciones. Además, puede utilizar la información que proporciona Internet Monitor para explorar posibles mejoras en la experiencia de sus clientes, ya sea mediante Amazon CloudFront o el enrutamiento por diferentes Regiones de AWS.
- Network Monitor utiliza un enfoque de agente totalmente administrado que le permite rastrear y visualizar la latencia y la pérdida de paquetes en las conexiones de red híbridas. Para recopilar mediciones y permitir que Network Monitor cree alertas de eventos de estado para su aplicación, debe crear sondeos que se envían desde sus recursos alojados en AWS a direcciones IP de destino en las instalaciones. No necesitará instalar agentes adicionales para supervisar el rendimiento de la red. Al igual que con Internet Monitor, puede establecer alertas y umbrales, obtener información que lo ayude a solucionar rápidamente los problemas y, a continuación, tomar medidas para mejorar la experiencia de usuario final.

Temas

Uso de Amazon CloudWatch Internet Monitor

Amazon CloudWatch Internet Monitor proporciona visibilidad sobre cómo los problemas de Internet afectan al rendimiento y la disponibilidad entre las aplicaciones alojadas en AWS y los usuarios finales. Puede reducir el tiempo que tarda en diagnosticar problemas de Internet de días a minutos. Internet Monitor utiliza los datos de conectividad que AWS recopila de su huella en la red global para calcular una línea base de rendimiento y disponibilidad del tráfico de Internet. Estos son los mismos datos que utilizamos en AWS para supervisar el tiempo de actividad y la disponibilidad de Internet.

Con esas mediciones como punto de referencia, Internet Monitor le avisa cuando hay problemas importantes para los usuarios finales (clientes) en las diferentes ubicaciones geográficas en las que se ejecuta la aplicación.

En la consola de Amazon CloudWatch, puede ver una vista global de los patrones de tráfico y los eventos de estado y analizar fácilmente la información sobre los eventos, con diferentes granularidades geográficas (ubicaciones). Puede visualizar el impacto de forma clara y determinar con precisión las ubicaciones y redes de clientes (ASN, por lo general, proveedores de servicios de Internet y proveedores de servicios de Internet) que se vean afectadas. Si Internet Monitor determina que un problema de disponibilidad o rendimiento de Internet se debe a un ASN específico o a la red de AWS, proporciona esa información.

Características principales de Internet Monitor

- Internet Monitor también sugiere ideas y recomendaciones que pueden servirle para mejorar la experiencia de los usuarios finales. Puede explorar, casi en tiempo real, cómo mejorar la latencia prevista de su aplicación pasando a utilizar otros servicios, o redirigiendo el tráfico a su carga de trabajo a través de diferentes Regiones de AWS.
- Con Internet Monitor, puede identificar rápidamente lo que afecta al rendimiento y la disponibilidad de su aplicación, de modo que pueda localizar y solucionar los problemas.
- Internet Monitor publica mediciones de Internet en registros de CloudWatch y CloudWatch Metrics, para apoyar el uso de herramientas CloudWatch con información de estado para ubicaciones y ASNs (proveedores de servicios de Internet) específicos de su aplicación. De forma opcional, también puede publicar las mediciones de Internet en Amazon S3.
- Internet Monitor envía eventos de estado a Amazon EventBridge para que pueda configurar notificaciones. Si la red de AWS provoca un problema, también recibirá automáticamente una notificación de AWS Health Dashboard que le indicará las medidas que AWS está tomando para mitigar el problema.

Cómo usar Internet Monitor

Para usar Internet Monitor, cree un monitor y asocie recursos de sus aplicaciones a él, ya sean nubes privadas virtuales (VPC), equilibradores de carga de red, distribuciones de CloudFront o directorios de WorkSpaces. De esa forma habilitará a Internet Monitor para que muestre dónde está el tráfico de Internet de su aplicación. A continuación, Internet Monitor publica las mediciones de Internet de AWS específicas de las redes urbanas, es decir, las ubicaciones de los clientes y las ASN (normalmente proveedores de servicios de Internet o ISP), desde donde los clientes acceden a su

aplicación. Para obtener más información, consulte Funcionamiento de Amazon CloudWatch Internet Monitor. Para empezar a trabajar con Internet Monitor, consulte Introducción a Amazon CloudWatch Internet Monitor con la consola.

Contenido

Amazon CloudWatch Internet Monitor es compatible con Regiones de AWS

En esta sección se describe la compatibilidad con Amazon CloudWatch Internet Monitor en Regiones de AWS. Si desea conocer la lista actual de las regiones compatibles con Internet Monitor, incluidas las regiones opcionales, consulte Amazon CloudWatch Internet Monitor endpoints and quotas (Puntos de conexión y cuotas de Amazon CloudWatch Internet Monitor) en la Referencia general de Amazon Web Services.

Tenga en cuenta que Internet Monitor almacena los datos de un monitor únicamente en la Región de AWS en que se creó, aunque un monitor puede incluir recursos de varias regiones.

Nombre de región (compatibilidad opcional)	Región
África (Ciudad del Cabo)	`af-south-1`
Asia-Pacífico (Hong Kong)	`ap-east-1`
Asia-Pacífico (Hyderabad)	`ap-south-2`
Asia-Pacífico (Yakarta)	`ap-southeast-3`
Asia-Pacífico (Melbourne)	`ap-southeast-4`
Europa (Milán)	`eu-south-1`
Europa (España)	`eu-south-2`
Europa (Zúrich)	`eu-central-2`
Medio Oriente (Baréin)	`me-south-1`
Medio Oriente (EAU)	`me-central-1`

Nombre de región (compatibilidad predeterminada)	Región
US East (Ohio)	`us-east-2`
Este de EE. UU. (Norte de Virginia)	`us-east-1`
Oeste de EE. UU. (Norte de California)	`us-west-1`
Oeste de EE. UU. (Oregón)	`us-west-2`
Asia-Pacífico (Bombay)	`ap-south-1`
Asia-Pacífico (Osaka)	`ap-northeast-3`
Asia-Pacífico (Seúl)	`ap-northeast-2`
Asia-Pacífico (Singapur)	`ap-southeast-1`

Nombre de región (compatibilidad predeterminada)	Región
Asia-Pacífico (Sídney)	ap-southeast-2
Asia-Pacífico (Tokio)	ap-northeast-1
Canadá (centro)	ca-central-1
Europa (Fráncfort)	eu-central-1
Europa (Irlanda)	eu-west-1
Europa (Londres)	eu-west-2
Europa (París)	eu-west-3
Europa (Estocolmo)	eu-north-1
América del Sur (São Paulo)	sa-east-1

Precios de Amazon CloudWatch Internet Monitor

Con Amazon CloudWatch Internet Monitor, no hay costes iniciales ni compromisos a largo plazo. El precio de Internet Monitor tiene dos componentes: una tarifa por recurso supervisado y una tarifa por red urbana. Una red urbana es la ubicación desde la que los clientes acceden a los recursos de la aplicación y la red (ASN, como un proveedor de servicios de Internet o un ISP) desde la que los clientes acceden a los recursos. Tenga en cuenta que también se le cobrarán los precios estándar de CloudWatch por los registros y cualquier métrica, panel, alarma o información adicional que cree.

Al crear un monitor, se elige el porcentaje del tráfico que se va a supervisar. Para ayudar a controlar su factura, también puede establecer un límite para la cantidad máxima de redes de la ciudad que se supervisan. Puede actualizar el porcentaje de tráfico que desea supervisar o el límite máximo de redes urbanas en cualquier momento editando su monitor. Se incluyen las primeras 100 redes urbanas (en todos los monitores de cada cuenta). Después, solo se paga por el número adicional real de redes urbanas que supervise, hasta el número máximo.

Solo pagará el número adicional real de redes urbanas que supervise, hasta el número máximo, sin coste alguno para las primeras 100 redes urbanas (en todos los monitores de cada cuenta). De la factura mensual se deduce un importe fijo equivalente al coste de 100 redes urbanas.

Por ejemplo, una gran empresa internacional podría optar por supervisar el 100 % de su tráfico conectado a Internet y establecer un máximo de 50 000 redes urbanas para un monitor con un recurso. Suponiendo que el tráfico llegara a 50 000 redes urbanas, esa parte de su factura rondaría los 2700 USD al mes. Para otra empresa, en menos áreas geográficas, con un monitor con un recurso y 200 redes urbanas, esta parte de la factura rondaría los 13 USD al mes. Para obtener más información, consulte Elegir un límite máximo entre ciudades y redes.

Puede probar diferentes opciones con la calculadora de precios. Para explorar las opciones de precios, en la página Calculadora de precios para CloudWatch, desplácese hacia abajo hasta Internet Monitor.

Para obtener más información acerca de los precios de Internet Monitor y CloudWatch, consulte la página Precios de Amazon CloudWatch.

Componentes y definiciones de Amazon CloudWatch Internet Monitor

Amazon CloudWatch Internet Monitor utiliza o hace referencia a lo siguiente.

Supervisar

Un monitor incluye los recursos de una sola aplicación para la que desea ver las mediciones de rendimiento y disponibilidad de Internet y sobre la que desea recibir alertas sobre eventos de estado. Al crear un monitor para una aplicación, se agregan recursos a la aplicación para definir las ciudades (ubicaciones) que Internet Monitor debe supervisar. Internet Monitor utiliza los patrones de tráfico de los recursos de la aplicación que agrega para publicar las mediciones del rendimiento y la disponibilidad de Internet específicas de las ubicaciones y redes ASN (por lo general, proveedores de servicios de Internet o ISP) que se comunican con la aplicación. En otras palabras, los recursos que añada crean un conjunto de redes urbanas que desea que Internet Monitor supervise y cuyas mediciones desea que publique.

Recurso agregado para supervisar («recurso supervisado»)

Un recurso que se agrega a un monitor es un «recurso supervisado» en Internet Monitor. Es decir:

- Cada VPC que añada en una región es un recurso supervisado. Al añadir una VPC, Internet Monitor supervisa el tráfico de cualquier aplicación con acceso a Internet de la VPC, por

ejemplo, una aplicación alojada en una instancia de Amazon EC2, detrás de un Equilibrador de carga de red o un contenedor de AWS Fargate.

- Cada equilibrador de carga de red que agregue en una región es un recurso supervisado.
- Cada directorio WorkSpaces que añada en una Región es un recurso supervisado.
- Cada distribución de CloudFront que añada es un recurso supervisado.

Número de sistema autónomo (ASN)

En Internet Monitor, un ASN normalmente hace referencia a un proveedor de red o proveedor de servicios de Internet (ISP), como Verizon o Comcast. Un ASN es un proveedor de red que un cliente utiliza para acceder a su aplicación de Internet. Un sistema autónomo (AS) es un conjunto de prefijos de protocolo de Internet (IP) enrutables de Internet que pertenecen a una red o a un conjunto de redes gestionadas, controladas y supervisadas por una organización.

Red urbana (ubicación y ASN)

Una red urbana es la ubicación (como una ciudad) desde la que los clientes acceden a los recursos de la aplicación y al ASN, normalmente un proveedor de servicios de Internet (ISP), a través del cual los clientes acceden a los recursos. Para ayudarle a controlar su factura, también puede establecer un límite para el número máximo de redes urbanas que Internet Monitor puede supervisar para cada monitor. Solo se paga por el número real de redes urbanas que supervise, hasta el número máximo. Para obtener más información, consulte Elección del tipo de red.

Mediciones de Internet

Internet Monitor publica las mediciones de Internet en archivos de registro de registros de CloudWatch cada cinco minutos para las 500 principales redes urbanas (ubicaciones de clientes y ASN, normalmente proveedores de servicios de Internet o ISP) de su cuenta. Estas mediciones cuantifican el puntaje de rendimiento, el puntaje de disponibilidad, los bytes transferidos (bytes de entrada y de salida) y el tiempo de ida y vuelta de las redes de ciudad de su aplicación. Son medidas de las redes urbanas específicas de sus VPC, equilibradores de carga de red, distribuciones de CloudFront o directorios de WorkSpaces. Si lo desea, puede optar por publicar las mediciones y los eventos de Internet de todas las redes urbanas supervisadas (hasta el límite de servicio de 500 000 redes urbanas) en un bucket de Amazon S3.

Métricas

Internet Monitor genera métricas agregadas de CloudWatch para el tráfico global a su aplicación y el tráfico global a cada Región de AWS. Para obtener más información, consulte Uso de las métricas de CloudWatch con Amazon CloudWatch Internet Monitor.

Evento de estado

Internet Monitor crea automáticamente eventos de estado para enviarle alertas sobre los problemas específicos que afectan a su aplicación. Internet Monitor detecta problemas de Internet, como el aumento de la latencia de la red, en todo el mundo. Luego, utiliza sus mediciones históricas de Internet de toda la infraestructura global de AWS para calcular el impacto de los problemas actuales en su aplicación y crea eventos de estado. Internet Monitor, de forma predeterminada, crea eventos de estado en función de los umbrales de impacto global y local. Para obtener más información sobre la configuración de los umbrales, consulte Cambiar los umbrales de los eventos de estado.

Cada evento de estado incluye información sobre las redes urbanas afectada. Puede ver los eventos de estado en la consola de CloudWatch o mediante el SDK de AWS o AWS CLI con las operaciones de la API de Internet Monitor. Internet Monitor también envía notificaciones de Amazon EventBridge para los eventos de estado. Para obtener más información, consulte Cuándo Internet Monitor crea y resuelve eventos de estado.

Evento de Internet

Internet Monitor muestra información sobre los recientes eventos de estado mundiales, denominados eventos de Internet, en un mapa meteorológico de Internet que está disponible para todos los clientes de AWS. No es necesario crear un monitor en Internet Monitor para ver el mapa meteorológico de Internet. A diferencia de los eventos de estado, los eventos de Internet no están asociados a clientes específicos ni al tráfico de sus aplicaciones. Para obtener más información, consulte Mapa meteorológico mundial de Internet en Amazon CloudWatch Internet Monitor.

Umbrales

Internet Monitor crea eventos de estado en función de los umbrales globales y locales. Puede cambiar los umbrales predeterminados y configurar otras opciones, como desactivar los umbrales locales. Para obtener más información sobre la configuración de los umbrales, consulte Cambiar los umbrales de los eventos de estado.

Puntuaciones de rendimiento y disponibilidad

Al analizar los datos que AWS recopila, Internet Monitor puede detectar cuándo el rendimiento y la disponibilidad de su aplicación han disminuido en comparación con las bases de referencia estimadas que calcula. Para que le resulte más fácil ver esas caídas, Internet Monitor le informa en forma de puntuaciones. Una puntuación de rendimiento representa el porcentaje estimado de tráfico que no experimenta una caída en el rendimiento. Del mismo modo, una puntuación de disponibilidad representa el porcentaje estimado de tráfico que no sufre una caída en la

disponibilidad. Para obtener más información, consulte Cómo AWS calcula las puntuaciones de rendimiento y disponibilidad.

Bytes transferidos y bytes supervisados transferidos

Bytes transferidos es el número total de bytes de tráfico de entrada y salida entre una aplicación en AWS y la red urbana (es decir, la ubicación y el ASN, normalmente el proveedor de servicios de Internet) desde la que los clientes acceden a una aplicación. Los bytes supervisados transferidos son una métrica similar, pero solo incluyen los bytes del tráfico supervisado.

Tiempo de ida y vuelta

El tiempo de ida y vuelta (RTT) es el tiempo que tarda una solicitud de un usuario cliente en devolver una respuesta al usuario. Cuando el RTT se agrega a todas las ubicaciones de los clientes (ciudades u otros lugares geográficos), el valor se pondera en función de la cantidad de tráfico de aplicaciones que proviene de cada ubicación del cliente.

Mapa meteorológico mundial de Internet en Amazon CloudWatch Internet Monitor

Amazon CloudWatch Internet Monitor muestra un mapa meteorológico mundial de Internet que está disponible para todos los clientes de AWS. Para ver el mapa, en la consola de Amazon CloudWatch, vaya a Internet Monitor.

El mapa destaca los eventos de Internet (“interrupciones”) en todo el mundo que afectan a los clientes de AWS, así como las ciudades y redes específicas (ASN, por lo general, proveedores de servicios de Internet) en las que hay problemas de rendimiento o disponibilidad. El mapa meteorológico de Internet incluye los eventos de Internet de las últimas 24 horas.

No es necesario crear un monitor en Internet Monitor para ver el mapa meteorológico de Internet. A diferencia de los eventos de estado en Internet Monitor, los eventos de Internet no están asociados a clientes específicos ni al tráfico de sus aplicaciones.

En el mapa meteorológico de Internet, puede elegir un evento de Internet para obtener más información sobre él. Puede ver la hora de inicio, la hora de finalización (si el evento ha terminado), el estado actual (activo o resuelto) y el tipo de problema de interrupción (disponibilidad o rendimiento). Para obtener más información sobre cómo se crea el mapa meteorológico mundial de Internet y qué incluye, consulte las preguntas frecuentes sobre el mapa meteorológico mundial de Internet.

Para ver y trabajar con información detallada específica del tráfico de las aplicaciones y las ubicaciones de los clientes, puede configurar fácilmente un monitor para su aplicación en Internet Monitor. De esta forma, verá patrones y eventos de rendimiento y disponibilidad, actuales e históricos, además de recibir alertas de eventos de estado, adaptadas solo a su huella de aplicaciones y a sus clientes. El mapa meteorológico de Internet le ofrece una visión general, mientras que un monitor específico filtra la información según las mediciones y los detalles relevantes para su aplicación. Con un monitor, también puede explorar las métricas históricas y obtener recomendaciones para mejorar la experiencia del cliente en su aplicación. Para obtener más información, consulte Introducción a Amazon CloudWatch Internet Monitor con la consola.

Funcionamiento de Amazon CloudWatch Internet Monitor

En esta sección, se proporciona información sobre el funcionamiento de Amazon CloudWatch Internet Monitor. Esto incluye descripciones de cómo AWS recopila los datos que utilice para ayudar a detectar problemas de conectividad en Internet y cómo se calculan las puntuaciones de rendimiento y disponibilidad.

Contenido

Cómo Internet Monitor se centra únicamente en la huella de tráfico de sus aplicaciones

Internet Monitor centra la supervisión en solo el subconjunto de Internet al que acceden los usuarios de sus recursos de AWS, en lugar de supervisar ampliamente su sitio web desde todas las regiones del mundo, como hacen otras herramientas. También es una solución rentable y asequible para grandes y pequeñas empresas.

Internet Monitor utiliza los mismos potentes sondeos y algoritmos de detección de problemas que AWS aprovecha internamente para enviarle alertas de los problemas de conectividad que afecten

a su aplicación mediante la creación de eventos de estado en Internet Monitor. A continuación, Internet Monitor le da acceso al mapa de rendimiento y disponibilidad resultante, al superponer el perfil de tráfico que cree a partir de sus espectadores activos, en función de los recursos de la aplicación.

Con esta información, Internet Monitor le muestra solo los eventos relevantes (es decir, los eventos de los lugares donde tiene espectadores activos) y únicamente el impacto que esos eventos tienen en su volumen total de espectadores. Por lo tanto, el impacto que tiene un evento, en términos de porcentajes, se basa en el tráfico total en todo el mundo.

Internet Monitor publica las mediciones de Internet en registros de CloudWatch cada cinco minutos para las 500 principales redes urbanas (ubicaciones de clientes y ASN, normalmente proveedores de servicios de Internet o ISP) que envían tráfico a cada monitor. Si lo desea, puede optar por publicar las mediciones de Internet de todas las redes urbanas supervisadas (hasta el límite de servicio de 500 000 redes urbanas) en un bucket de Amazon S3. Para obtener más información, consulte Publicar mediciones de Internet en Amazon S3 en Amazon CloudWatch Internet Monitor.

Algunas de las ventajas de Internet Monitor son las siguientes:

- El uso de Internet Monitor no supone una carga ni un costo adicionales para la aplicación alojada en AWS.
- No tiene que incluir código de medición del rendimiento en los recursos del lado del cliente ni en la aplicación.
- Puede ver el rendimiento y la disponibilidad en Internet a la que esté conectada su aplicación, incluida la información sobre los usuarios finales.

Tenga en cuenta que, dado que Internet Monitor crea mediciones en función de sus recursos de AWS, Internet Monitor solo crea eventos que sean específicos del tráfico de su aplicación. No se informa sobre problemas globales de Internet en general. Además, cuando la ubicación del servicio es una región de Región de AWS, las mediciones y los eventos emitidos están diseñados para representar la conectividad en un ámbito regional y no representan con precisión la conectividad entre una ubicación de usuario final y una zona de disponibilidad.

Cómo AWS mide los problemas de conectividad y calcula las medidas

Amazon CloudWatch Internet Monitor utiliza los datos de conectividad a Internet entre diferentes Regiones de AWS y puntos de presencia (POP) de Amazon CloudFront con diferentes ubicaciones de clientes a través de números de sistema autónomos (ASN), normalmente proveedores de servicios de Internet (ISP). Estos son los datos de conectividad que los

operadores de AWS utilizan internamente cada día para detectar de forma proactiva problemas de conectividad en Internet en todo el mundo.

Para cada región de Región de AWS, sabemos qué partes de Internet se comunican con la región y hacemos lo siguiente:

- Supervisamos activamente esas partes de Internet, con un período continuo de 30 días.
- Utilizamos sondeos de red y de protocolo de nivel superior, así como sondeos entrantes y salientes.

AWS cuenta con sondas activas y pasivas que miden la latencia (rendimiento) en el percentil 90 y la accesibilidad (disponibilidad) desde todas las regiones de Región de AWS y desde el servicio de CloudFront hacia todo Internet. Los patrones anormales de conectividad entre un servicio y la ubicación de un cliente se supervisan y, luego, se notifican al cliente en forma de alertas.

Cómo calcular la disponibilidad y el RTT

El tiempo de ida y vuelta (RTT) es el tiempo que tarda una solicitud del usuario en devolverle una respuesta. Cuando se agrega el tiempo de ida y vuelta entre las ubicaciones de los usuarios finales, el valor se pondera en función de la cantidad de tráfico generado por cada ubicación de usuario final.

Por ejemplo, con dos ubicaciones de usuarios finales, una que atiende el 90 % del tráfico con un RTT de 5 ms y la otra que atiende el 10 % del tráfico con un RTT de 10 ms, el resultado es un RTT agregado de 5,5 ms (que proviene de 5 ms * 0,9 + 10 ms * 0,1).

Tenga en cuenta que existen diferencias en los recursos sobre la medición de la latencia de última milla. Para las mediciones de latencia de Internet Monitor, los directorios de VPC, Equilibradores de carga de red y WorkSpaces no incluyen la latencia de última milla.

Cómo calcular el rendimiento y la disponibilidad

AWS tiene datos históricos sustanciales sobre el rendimiento y la disponibilidad de Internet entre los servicios de AWS y las distintas redes urbanas (ubicaciones y ASN). Al aplicar análisis estadísticos a los datos, podemos detectar cuándo el rendimiento y la disponibilidad de su aplicación han disminuido en comparación con una base de referencia estimada que se ha calculado. Para que le resulte más fácil ver esas caídas, esa información se le presenta en forma de puntuaciones de estado: una puntuación de rendimiento y una puntuación de disponibilidad.

Las puntuaciones de estado se calculan con diferentes niveles de detalle. Con el máximo detalle, calculamos la puntuación de estado de una región geográfica, como una ciudad o un

área metropolitana, y de un ASN (una ciudad urbana). También agrupamos las puntuaciones de estado individuales en números de puntuación de estado globales para aplicarlas en un monitor. Si consulta las puntuaciones de rendimiento o disponibilidad sin filtrar por una ubicación geográfica o un proveedor de servicios específicos, Internet Monitor proporciona puntuaciones de estado globales.

Las puntuaciones de estado globales abarcan toda la aplicación durante el período de tiempo especificado. Cuando la puntuación de rendimiento o disponibilidad de los pares de ciudades y redes de la aplicación en toda la aplicación alcanza o cae por debajo del umbral de eventos de estado correspondiente al rendimiento o la disponibilidad, Internet Monitor desencadena un evento de estado. De forma predeterminada, el umbral es del 95 % tanto para el rendimiento general como para la disponibilidad. Internet Monitor también crea eventos de estado en función de los umbrales locales (si la opción está habilitada, como lo está de forma predeterminada) en función de los valores que se configuren. Para obtener más información sobre la configuración de los umbrales, consulte Cambiar los umbrales de los eventos de estado.

Cuando explore la información del monitor y los archivos de registro para investigar los problemas y obtener más información, puede filtrar por ciudad (ubicación) y red (ASN o proveedor de servicios de Internet), o por ambos. Por lo tanto, puede usar filtros para ver las puntuaciones de estado de diferentes ciudades, ASN o pares de ciudades y redes, según los filtros que elija.

- Una puntuación de disponibilidad representa el porcentaje estimado de tráfico que no sufre una caída en la disponibilidad. Internet Monitor calcula el porcentaje de tráfico que experimenta una caída con respecto al tráfico total visto y a las mediciones de las métricas de disponibilidad. Por ejemplo, una puntuación de disponibilidad del 99 % para un par de usuarios finales y ubicaciones de servicios equivale a un 1 % de tráfico que experimenta una caída de disponibilidad para ese par.
- Una puntuación de rendimiento representa el porcentaje de tráfico que no experimenta una caída en el rendimiento. Por ejemplo, una puntuación de rendimiento del 99 % para un par de usuarios finales y ubicaciones de servicios equivale a un 1 % de tráfico que experimenta una caída de rendimiento para ese par.

Cómo calcular el TTFB y el RTT (latencia)

El tiempo hasta el primer byte (TTFB) se refiere al tiempo que transcurre entre el momento en el que el cliente realiza una solicitud y el momento en el que se recibe el primer byte de información del servidor. Los cálculos de TTFB de AWS miden el tiempo transcurrido desde

Amazon EC2 o Amazon CloudFront hasta el nodo de medición de Internet Monitor (incluida la última milla del nodo). Es decir, Internet Monitor mide el tiempo que pasa desde el usuario hasta la región Amazon EC2 para TTFB para EC2 y desde el usuario hasta CloudFront para TTFB para CloudFront.

En el caso del tiempo de ida y vuelta (RTT), Internet Monitor incluye el tiempo transcurrido desde la red urbana (es decir, la ubicación del cliente y el ASN, normalmente un proveedor de servicios de Internet), mapeado por la dirección IP pública, hasta la Región de AWS. Esto significa que Internet Monitor no tiene visibilidad de última milla para los usuarios que acceden a Internet desde una puerta de enlace o una VPN.

Tenga en cuenta que existen diferencias en los recursos sobre la medición de la latencia de última milla. Para las mediciones de latencia de Internet Monitor, los directorios de VPC, Equilibradores de carga de red y WorkSpaces no incluyen la latencia de última milla.

Internet Monitor incluye información promedio de TTFB en la sección de sugerencias de optimización del tráfico de la pestaña Información sobre el tráfico del panel de CloudWatch, para ayudarle a evaluar las opciones para las diferentes configuraciones de su aplicación que pueden mejorar el rendimiento.

Mediciones y agregaciones regionales y de zonas de disponibilidad

Si bien Internet Monitor agrega las mediciones y comparte el impacto a nivel regional, calcula el impacto a nivel de zona de disponibilidad (AZ). Esto significa que, si en el caso de un evento, solo se ve afectada una zona de disponibilidad y la mayor parte del tráfico fluye a través de esa zona, verá el impacto en su tráfico. Sin embargo, en el mismo caso, si el tráfico de su aplicación no fluye a través de una zona de disponibilidad afectada, no verá ningún impacto.

Tenga en cuenta que esto solo se aplica a los recursos que no son directorios de WorkSpaces. Los directorios de WorkSpaces se miden solo a nivel regional.

Precisión de la geolocalización en Internet Monitor

Para obtener información sobre la ubicación, Internet Monitor utiliza los datos de geolocalización IP proporcionados por MaxMind. La precisión de la información de ubicación en las mediciones del Monitor de Internet depende de la precisión de los datos de MaxMind.

Tenga en cuenta que las mediciones de nivel de `Metro` pueden no ser precisas en ubicaciones fuera de los Estados Unidos.

Cuándo Internet Monitor crea y resuelve eventos de estado

Internet Monitor crea y cierra los eventos de estado del tráfico de las aplicaciones que usted supervisa en función de los umbrales actuales establecidos. Internet Monitor tiene una configuración de umbrales predeterminada y también puede establecer su propia configuración para los umbrales. Internet Monitor determina el impacto general que los problemas de conectividad están teniendo en la aplicación y el impacto en las áreas locales donde la aplicación tiene clientes, y crea eventos de estado cuando se superan los umbrales.

Internet Monitor calcula el impacto de los problemas de conectividad en la ubicación de un cliente a partir de los datos históricos sobre el rendimiento de Internet y la disponibilidad del tráfico de red que esté disponible para el servicio mediante AWS. Aplica la información relevante para su aplicación en función de las ubicaciones geográficas de los ASN y los servicios en los que los clientes utilizan su aplicación: los pares de ciudades y redes que se ven afectados. Las ubicaciones se determinan a partir de los recursos que se agreguen al monitor. Internet Monitor utiliza análisis estadístico para detectar el descenso del rendimiento y la disponibilidad, lo que afecta a la experiencia del cliente de la aplicación.

Las puntuaciones de rendimiento y disponibilidad que calcula Internet Monitor se representan como el porcentaje del tráfico que no experimenta ninguna caída. El impacto es lo contrario de esto: es una representación de la gravedad de un problema para los usuarios finales de un cliente. Por lo tanto, si se produce una caída de la disponibilidad global del 93 %, por ejemplo, el impacto correspondiente sería del 7 %.

Cuando la puntuación de rendimiento o disponibilidad de los pares de ciudades y redes de la aplicación alcanza o cae globalmente por debajo del umbral de eventos de estado correspondiente al rendimiento o la disponibilidad, Internet Monitor desencadena un evento de estado. De forma predeterminada, el umbral es del 95 % tanto para el rendimiento como para la disponibilidad. Los valores para alcanzar el umbral o caer por debajo de él son acumulativos, por lo que podría significar que varios eventos más pequeños se combinan para alcanzar el porcentaje del umbral o que un solo evento alcanza o cae por debajo del nivel umbral.

Mientras las puntuaciones de rendimiento o disponibilidad que desencadenaron el evento sean iguales o inferiores al porcentaje del umbral de impacto global correspondiente, el evento de estado permanecerá activo. Cuando la puntuación o las puntuaciones combinadas que desencadenaron el evento superan el umbral, Internet Monitor resuelve el problema de estado.

Internet Monitor también crea eventos de estado en función de los umbrales locales y del porcentaje del tráfico total al que afecta un problema. Puede configurar opciones para los umbrales locales o desactivarlos por completo.

Para obtener más información sobre la configuración de los umbrales, consulte Cambiar los umbrales de los eventos de estado.

Calendario de informes de eventos de estado

Internet Monitor utiliza un agregador para recopilar todas las señales sobre problemas de Internet y crear eventos de estado en los monitores en cuestión de minutos.

Cuando es posible, Internet Monitor analiza el origen de un evento de estado para determinar si fue causado por AWS o un ASN. El análisis de los eventos de estado continúa después de que se resuelva un evento. Internet Monitor puede actualizar los eventos con nueva información durante un máximo de una hora.

Cómo funciona Internet Monitor con el tráfico de IPv4 e IPv6

Internet Monitor mide el estado de una red únicamente a través de IPv4 y le muestra los eventos de estado y las métricas de disponibilidad y rendimiento si envía tráfico a esa red a través de cualquier familia de IP (IPv4 o IPv6). Si atiende el tráfico desde un recurso de doble pila, como una distribución de CloudFront de doble pila, Internet Monitor genera un problema de estado y muestra una caída en la puntuación de rendimiento o disponibilidad solo si el tráfico IPv4 presenta los mismos problemas para el recurso que el tráfico IPv6.

Tenga en cuenta que las métricas de Internet Monitor para el total de bytes de entrada y salida reflejan con precisión todo el tráfico de Internet (IPv4 e IPv6).

Cómo Internet Monitor selecciona el subconjunto de redes urbanas que desea incluir

Al establecer un límite máximo para el número de redes urbanas supervisadas por el monitor o al elegir un porcentaje del tráfico a supervisar, Internet Monitor selecciona las redes urbanas que se incluirán (supervisarán) según el volumen de tráfico reciente más alto.

Por ejemplo, si establece un límite máximo de 100 redes urbanas, Internet Monitor supervisará hasta 100 redes urbanas según el tráfico de sus aplicaciones durante un período reciente de una hora. Concretamente, Internet Monitor supervisa las 100 principales redes urbanas que han tenido más tráfico en el último período de una hora antes del último período de una hora.

Para ilustrarlo, supongamos que la hora actual son las 14:30 h. En este escenario, el tráfico que ve en el monitor se capturó entre las 13:00 h y las 14:00 h, y la medición del volumen de tráfico

que Internet Monitor utiliza para determinar las 100 principales redes urbanas se capturó entre las 12:00 h y las 13:00 h.

Cómo se crea el mapa meteorológico mundial de Internet (preguntas frecuentes)

El mapa meteorológico de Internet de Amazon CloudWatch Internet Monitor está disponible en la consola de Internet Monitor para todos los clientes autenticados de AWS. En esta sección se incluyen detalles sobre cómo se crea el mapa meteorológico de Internet y cómo usarlo.

¿Qué es el mapa meteorológico de Internet de Internet Monitor?

El mapa meteorológico de Internet proporciona una representación visual e interactiva de los problemas de Internet en todo el mundo. Destaca las ubicaciones de los clientes afectados, es decir, las ciudades y las ASN (normalmente proveedores de servicios de Internet). El mapa muestra una combinación de problemas de disponibilidad y rendimiento que han afectado recientemente a la experiencia de Internet de los clientes en sus principales ubicaciones y los servicios de AWS de todo el mundo.

¿De dónde provienen los datos del mapa?

Los datos se basan en una combinación de exploraciones activas y pasivas de Internet. Para obtener más información sobre cómo Internet Monitor mide los datos, consulte la sección Cómo AWS mide los problemas de conectividad.

¿Con qué frecuencia se actualiza el mapa?

El mapa meteorológico de Internet se actualiza cada 15 minutos.

¿De qué redes se realiza un seguimiento de las interrupciones?

AWS rastrea redes en todo el mundo que representan prefijos IP importantes utilizados por los clientes para realizar conexiones de Internet a AWS. Abarcamos las interrupciones en las ubicaciones de los clientes que son las más importantes en cuanto al volumen de tráfico enviado y recibido desde la red de AWS.

¿Qué determina si un evento de Internet se incluye en el mapa?

Estos son algunos criterios generales que utilizamos para determinar si un evento de Internet se incluye en el mapa meteorológico de Internet:

- AWS detecta que hay un evento de disponibilidad o rendimiento.
- Si el evento es de corta duración, por ejemplo, dura menos de 5 minutos, lo ignoramos.
- Entonces, si el evento se produce en la ubicación de un cliente que está clasificada como una de las más visitadas, se considera una interrupción.

¿Qué umbrales se utilizan para el mapa meteorológico de Internet?

Los umbrales para determinar las interrupciones no son estáticos en el mapa meteorológico de Internet. Internet Monitor determina lo que constituye un evento basándose en la detección de una desviación de los valores esperados. Para obtener más información sobre cómo funciona esto, consulte cómo Internet Monitor determina cuándo se deben crear eventos de estado para los monitores que se crean con el servicio. Al crear un monitor, Internet Monitor genera mediciones del estado del tráfico de Internet que son específicas del tráfico de su propia aplicación. Internet Monitor también le avisa de problemas de estado que afectan al tráfico de Internet de su aplicación.

¿Qué puedo hacer con estos datos?

El mapa meteorológico de Internet proporciona un resumen rápido de los principales eventos de Internet que se han producido en todo el mundo en las últimas 24 horas. Le ayuda a hacerse una idea de la experiencia de supervisión de Internet, sin necesidad de incorporar su propio tráfico de Internet a Internet Monitor. Para aprovechar todo el potencial de las capacidades de supervisión de Internet de AWS y personalizarlas para sus aplicaciones y servicios alojados en AWS, puede crear un monitor en Internet Monitor.

Al crear un monitor, permite que Internet Monitor identifique las rutas de Internet específicas que afectan a los clientes de sus aplicaciones y obtiene acceso a características y capacidades que pueden ayudarle a mejorar la experiencia de sus clientes. También recibirá notificaciones proactivas de los nuevos problemas de Internet que afecten específicamente al tráfico y a los clientes de sus aplicaciones.

¿Cómo puedo obtener más información sobre los eventos?

Haga clic en una interrupción en el mapa para ver detalles como cuándo comenzó y terminó el evento, la ciudad y los ASN afectados, y el tipo de problema (es decir, un problema de rendimiento o un problema de disponibilidad).

Para obtener información más detallada sobre los eventos y mediciones personalizadas del tráfico de sus aplicaciones, cree un monitor en Internet Monitor.

Ejemplos de casos de uso de Amazon CloudWatch Internet Monitor

En esta sección, describimos varios ejemplos específicos, con enlaces a publicaciones de blog con más detalles. Estos ejemplos muestran cómo puede utilizar las capacidades de Amazon CloudWatch Internet Monitor para supervisar su aplicación y mejorar la experiencia de los usuarios.

Configure alertas y decida qué acciones tomar

Puede utilizar Internet Monitor para obtener información sobre las métricas de rendimiento promedio de Internet a lo largo del tiempo y sobre los eventos de estado por ciudad y red (ubicación del cliente y ASN, normalmente un proveedor de servicios de Internet). Con Internet Monitor, puede identificar los eventos que afectan a la experiencia del usuario final en el caso de las aplicaciones alojadas en nubes privadas virtuales (VPC) de Amazon, Equilibradores de carga de red, Amazon WorkSpaces o Amazon CloudFront.

Después de crear un monitor, tiene varias opciones para recibir alertas sobre los eventos de estado de Internet Monitor. Estas incluyen notificaciones basadas en alarmas de CloudWatch que utilizan métricas de eventos o reglas de Amazon EventBridge para filtrar los eventos de estado. Puede elegir diferentes opciones para las notificaciones o acciones en función de las alarmas, incluidas, por ejemplo, notificaciones AWS SMS o actualizaciones de un grupo de registro de CloudWatch.

Para ver un ejemplo con instrucciones detalladas, consulte la siguiente entrada del blog: Presentación de Amazon CloudWatch Internet Monitor.

Identifique los problemas de latencia y mejore el TTFB para mejorar la experiencia de juego multijugador

Use Internet Monitor para ayudarle a identificar rápidamente los problemas de latencia que experimentan los jugadores en las aplicaciones de juegos en la nube de todo el mundo y proporcionar información sobre cómo mejorar el rendimiento. Al identificar los lugares en los que el mayor número de jugadores tienen actualmente el menor tiempo hasta el primer byte (TTFB), sabrá cómo mejorar la latencia para que su base de jugadores más numerosa esté más satisfecha.

Cuando esté listo para implementar el siguiente servidor EC2 para su juego, elija la Región de AWS que Internet Monitor sugiera que reducirá el TTFB en la zona con alta latencia y gran cantidad de jugadores.

Para obtener más información sobre la configuración y el uso de Internet Monitor para este caso, consulte la siguiente entrada del blog: Uso de Amazon CloudWatch Internet Monitor para una mejor experiencia de juego.

Observabilidad entre cuentas de Internet Monitor

Gracias a la observabilidad entre cuentas de Internet Monitor, puede supervisar las aplicaciones que abarcan varias cuentas de AWS de una sola Región de AWS.

Puede usar el Administrador de acceso a la observabilidad de Amazon CloudWatch para configurar una o más de sus cuentas AWS como una cuenta de supervisión. Para proporcionar a la cuenta de supervisión la capacidad de ver los datos de su cuenta de origen, deberá crear un depósito en su cuenta de supervisión. Un receptor es un recurso que representa un punto de enlace en una cuenta de supervisión. En el caso de Internet Monitor, el punto de conexión del recurso es un monitor. Use el receptor para crear un enlace desde su cuenta de origen a su cuenta de supervisión. Para obtener más información, consulte Observabilidad entre cuentas de CloudWatch.

Recursos necesarios de

Para que la observabilidad multicuenta de Información de aplicaciones de CloudWatch funcione correctamente, asegúrese de que los siguientes tipos de telemetría se compartan a través del administrador de acceso a la observabilidad de CloudWatch.

- Monitores en Internet Monitor
- Métricas en Amazon CloudWatch
- Grupos de registro de Amazon CloudWatch Logs

Introducción a Amazon CloudWatch Internet Monitor con la consola

Para empezar a utilizar Amazon CloudWatch Internet Monitor, debe crear un monitor en Internet Monitor para su aplicación añadiendo los recursos de AWS que utilice. En este capítulo se describe el procedimiento para añadir un monitor a la consola. También incluye una sección con más información sobre los recursos de Internet Monitor y, a continuación, secciones adicionales con descripciones y limitaciones de las distintas opciones que puede o debe configurar para el monitor.

Contenido

- Uso de un monitor de Internet Monitor
- Edición o eliminación de un monitor de Internet Monitor
- Agregar o crear un monitor Amazon CloudWatch Internet Monitor mediante Amazon VPC
- Agregación o creación de un monitor de Amazon CloudWatch Internet Monitor con CloudFront

Cree un monitor en Amazon CloudWatch Internet Monitor mediante la consola

Cree un monitor en Amazon CloudWatch Internet Monitor para su aplicación añadiendo los recursos de AWS que utilice y estableciendo a continuación varias opciones de configuración. Los recursos que se agregan, como nubes privadas virtuales (VPC) de Amazon, equilibradores de carga de red (NLB), distribuciones de CloudFront o directorios de WorkSpaces, proporcionan la información necesaria para que Internet Monitor visualice los datos de tráfico de Internet de la aplicación. Tras crear el monitor, espere entre quince a treinta minutos para generar el perfil de tráfico específico del lugar en el que se utiliza la aplicación. Luego podrá utilizar el monitor de Internet Monitor u otras herramientas para visualizar y explorar el rendimiento y la disponibilidad en relación con el uso que hace de sus clientes. Estas herramientas le proporcionan información mediante las mediciones del tráfico de su aplicación, recopiladas y publicadas por el monitor, por ejemplo, en registros de CloudWatch.

Por lo general, lo más sencillo es crear un monitor en Internet Monitor para una aplicación. En el mismo monitor, puede buscar y ordenar las medidas y métricas de los archivos de registro de Internet Monitor por diferentes ubicaciones y ASN (normalmente proveedores de servicios de Internet) u otra información. No es necesario crear monitores separados para aplicaciones en áreas diferentes, por ejemplo.

En los pasos de esta sección se explica cómo configurar el monitor con Internet Monitor mediante la consola. Para ver ejemplos del uso de AWS Command Line Interface con las acciones de la API de Internet Monitor, para crear un monitor, ver eventos, etc., consulte Ejemplos de uso de la CLI con Amazon CloudWatch Internet Monitor.

Crear un monitor con la consola

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación izquierdo, en Supervisión de redes, elija Monitor de Internet.
3. Elija Crear monitor.
4. En Monitor name (Nombre del monitor), ingrese el nombre que desee utilizar para este monitor en Internet Monitor.

5. Elija Add resources (Agregar recursos) y, a continuación, seleccione los recursos para establecer los límites de supervisión que Internet Monitor utilizará para este monitor.

> **Note**
>
> Tenga en cuenta lo siguiente:
>
> - Para generar resultados significativos con Internet Monitor, las VPC que agregue deben estar conectadas a Internet mediante una puerta de enlace de Internet configurada.
> - Puede añadir una combinación de VPC y distribuciones de CloudFront, o bien puede añadir directorios de WorkSpaces o Equilibradores de carga de red. No puede agregar directorios de Equilibradores de carga de red ni WorkSpaces junto con otros tipos de recursos.

6. Elija el porcentaje del tráfico de Internet que desee supervisar.
7. Si lo desea, especifique opciones adicionales en Configuración avanzada.

   - Para el Máximo de redes urbanas, puede seleccionar un límite para el número de redes urbanas (ubicaciones y redes ASN o proveedores de servicios de Internet) en las que Internet Monitor supervisará el tráfico. Puede cambiar esto en cualquier momento editando el monitor. Consulte Elegir un límite máximo entre ciudades y redes.

     Para restablecer los valores predeterminados, introduzca `500000`.

     Si establece un límite máximo para las redes urbanas, se fija un límite para el número de redes urbanas que Internet Monitor supervisa para su aplicación, independientemente del porcentaje de tráfico que decida supervisar.
   - Si lo desea, puede especificar un nombre de bucket de Amazon S3 y un prefijo personalizado para publicar las mediciones de Internet en Amazon S3 para todas las redes urbanas supervisadas.

     Internet Monitor publica las 500 principales mediciones de Internet (por volumen de tráfico) de su aplicación en registros de CloudWatch cada cinco minutos. Si decide publicar las mediciones en S3, las mediciones seguirán publicándose en registros de CloudWatch. Para obtener más información, consulte Publicar mediciones de Internet en Amazon S3 en Amazon CloudWatch Internet Monitor.
   - Si lo desea, puede añadir una etiqueta para el monitor.

8. Elija Crear monitor.

Tras crear un monitor, puede editarlo en cualquier momento, por ejemplo, para cambiar el porcentaje de tráfico de la aplicación, actualizar el límite máximo de redes urbanas o añadir o eliminar recursos. También puede eliminar el monitor. Para realizar estas tareas, en la consola de Internet Monitor, seleccione un monitor y, a continuación, elija una opción en el menú Acción. Tenga en cuenta que no puede cambiar el nombre del monitor.

Visualización del panel de Internet Monitor

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Supervisión de redes y, a continuación, Monitor de Internet.

   En la pestaña Monitors (Monitores) se observa una lista de los monitores que ha creado.

Para ver más información sobre un monitor concreto, selecciónelo.

Agregar recursos a su monitor

Cuando crea un monitor, lo asocia a los recursos de la aplicación: nubes privadas virtuales (VPC) de Amazon, equilibradores de carga de red, distribuciones de Amazon CloudFront, equilibradores de carga de red (NLB) o directorios de Amazon WorkSpaces. De este modo, Internet Monitor puede identificar la ubicación del tráfico y los clientes de la aplicación con acceso a Internet, y así crear y mantener un perfil de tráfico que determine las medidas pertinentes para monitorear su publicación.

Puede añadir los siguientes recursos a un monitor en Internet Monitor como «recursos supervisados». Tenga en cuenta que Internet Monitor no admite la adición de distintos tipos de recursos en un monitor.

- VPCs: Cada VPC que añada en una región es un recurso supervisado. Al añadir una VPC, Internet Monitor supervisa el tráfico de cualquier aplicación con acceso a Internet de la VPC, por ejemplo, una aplicación alojada en una instancia de Amazon EC2, detrás de un Equilibrador de carga de red o en un contenedor de AWS Fargate.
- Equilibradores de carga de red: cada equilibrador de carga que añada es un recurso supervisado.
- Distribuciones de CloudFront: Cada distribución de CloudFront que añada es un recurso supervisado.
- Directorios de WorkSpaces: Cada directorio de WorkSpaces que añada en una región es un recurso supervisado.

Al supervisar el tráfico de las VPC, se supervisa el tráfico de las aplicaciones que están alojadas en los equilibradores de carga detrás de la VPC. Puede optar por supervisar el tráfico de los Equilibradores de carga de red individuales en lugar de supervisar una VPC con varios equilibradores de carga. Esto puede resultar útil, por ejemplo, si necesita entender y configurar las características para mejorar el rendimiento o la eficiencia a nivel del equilibrador de carga. O puede que necesite información sobre el cumplimiento a nivel del Equilibrador de carga de red.

Cuando añada recursos a un monitor en Internet Monitor, tenga en cuenta lo siguiente:

- Para generar resultados significativos con Internet Monitor, las VPC que agregue deben estar conectadas a Internet mediante una puerta de enlace de Internet configurada.
- Internet Monitor no admite la adición de distintos tipos de recursos en un monitor.

A la hora de añadir VPC y equilibradores de carga de red (NLB) como recursos, se deben tener en cuenta las diferencias regionales entre las regiones en las que se ha optado por participar. Para obtener más información, consulte Amazon CloudWatch Internet Monitor es compatible con Regiones de AWS.

Asimismo, tenga en cuenta que existen diferencias en los recursos sobre la medición de la latencia de última milla. Para las mediciones de latencia de Internet Monitor, los directorios de VPC, Equilibradores de carga de red y WorkSpaces no incluyen la latencia de última milla.

Elegir el porcentaje de tráfico de aplicaciones que se va a supervisar

La cobertura que elija para el porcentaje del tráfico de aplicaciones que se va a supervisar determina el número de redes urbanas (ubicaciones de clientes y ASN, normalmente proveedores de servicios de Internet) que se supervisarán para su aplicación, hasta un límite máximo opcional de redes urbanas que también puede establecer.

Si decide supervisar menos del 100 % del tráfico de sus aplicaciones, es posible que haya una brecha de observabilidad en el monitor. Esto se debe a que si Amazon CloudWatch Internet Monitor crea eventos de estado en los que no supervisa el tráfico, no se dará cuenta de esos problemas. También es posible que tenga menos cobertura de la información sobre las puntuaciones de rendimiento y disponibilidad sobre el acceso de los clientes a su aplicación.

En las siguientes secciones se describen las opciones para explorar la configuración del porcentaje de tráfico y la cobertura, y para brindar una noción del impacto de aumentar o disminuir la cobertura.

- Explore cómo cambiar el porcentaje de tráfico de sus aplicaciones

- Vea la cantidad de redes urbanas supervisadas con diferentes configuraciones de porcentaje de tráfico

Explore cómo cambiar el porcentaje de tráfico de sus aplicaciones

Puede explorar los valores a los que quizás desee cambiar el porcentaje de tráfico de aplicaciones consultando la cantidad de redes urbanas supervisadas al cambiar el porcentaje. El procedimiento de esta sección proporciona información paso a paso.

En la consola de Internet Monitor, puede intentar aumentar o reducir el porcentaje de tráfico de aplicaciones del monitor y ver el número estimado de redes urbanas que se cubrirían como resultado. Con esta opción, puede comprobar rápidamente cómo el cambio en el porcentaje de tráfico afecta al número de monitores urbanos que se supervisan. Esto puede ayudarle a hacerse una idea de cuál podría ser el mejor porcentaje de tráfico de aplicaciones para su aplicación.

Explorar la cobertura de supervisión aumentando y disminuyendo el porcentaje de tráfico de aplicaciones

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación izquierdo, en Supervisión de redes, elija Monitor de Internet.
3. En su lista de monitores, elija uno.
4. En la pestaña Descripción general, en la sección Tráfico supervisado, elija el gráfico de porcentaje y, a continuación, elija Actualizar la cobertura de supervisión.
5. En el cuadro de diálogo Explore y establezca la cobertura de supervisión del tráfico, haga clic en las flechas para aumentar o disminuir el porcentaje de tráfico que se debe supervisar. Al elegir el 100 % del tráfico, puede ver cuántas redes urbanas se supervisan con una cobertura total para supervisar su aplicación.
6. Para obtener más información sobre cómo la cantidad de redes urbanas supervisadas (estimadas aquí) podría afectar a sus costes, seleccione el enlace a la calculadora de precios de CloudWatch y, a continuación, desplácese hacia abajo hasta Internet Monitor.
7. Para establecer el nuevo porcentaje de tráfico que se va a supervisar, seleccione Actualizar la cobertura del monitor. O bien, para mantener el nivel de cobertura actual, seleccione Cancelar.

Ver la cantidad de redes urbanas supervisadas con diferentes configuraciones de porcentaje de tráfico

Puede ver la cantidad de redes urbanas que se supervisarían para su aplicación en diferentes porcentajes de tráfico de aplicaciones. El procedimiento de esta sección proporciona información paso a paso.

En la consola de Internet Monitor, puede ver gráficos que muestran cómo cambiaría la cobertura de las redes de las ciudades en función de los distintos porcentajes de tráfico de las aplicaciones, durante un intervalo de tiempo que especifique. Es una forma rápida de visualizar y comparar la cobertura de supervisión de su aplicación en porcentajes de tráfico específicos, todo en un solo gráfico.

Ver los gráficos del porcentaje de tráfico de las aplicaciones y la cobertura correspondiente de las redes urbanas

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación izquierdo, en Supervisión de redes, elija Monitor de Internet.
3. En su lista de monitores, elija uno.
4. Seleccione la pestaña Estadísticas de tráfico y desplácese hacia abajo hasta los gráficos de tráfico de Internet.
5. En Comparar opciones de cobertura de tráfico, en la lista desplegable, seleccione uno o más porcentajes. Puede elegir uno o varios porcentajes de tráfico de aplicaciones y el gráfico del total de redes urbanas supervisadas se actualiza para mostrar la cobertura de supervisión que Internet Monitor proporciona para ese porcentaje de tráfico. Al elegir Redes urbanas al 100 % del tráfico, puede ver cuántas redes urbanas se supervisan con una cobertura total para supervisar su aplicación.

Tenga en cuenta lo siguiente:

- La cobertura del tráfico se calcula en función de la cantidad de redes urbanas durante la hora anterior al tráfico de su aplicación. Esto significa que, después de elegir un porcentaje específico de tráfico al cual supervisar, es posible que se supervisen menos redes urbanas para su aplicación de las que se muestran aquí en un gráfico comparativo de cobertura de tráfico.
- Para asegurarse de que se supervise todo el tráfico de su aplicación, establezca `TrafficPercentageToMonitor` en 100 y no establezca `MaxCityNetworksToMonitor`.

Como alternativa, puede establecer `MaxCityNetworksToMonitor` en 500 000, el límite superior de Internet Monitor.

- Si estableció un límite máximo para las redes urbanas, la cantidad total de redes urbanas supervisadas no excederá ese límite, independientemente de las opciones de porcentaje de tráfico de aplicaciones que seleccione.
- Puede obtener más información sobre cómo la cantidad de redes urbanas supervisadas podría afectar a sus costes. En la página Calculadora de precios de CloudWatch, desplácese hacia abajo hasta Internet Monitor.

Para establecer un nuevo porcentaje de tráfico que supervisar, en Explorar otras opciones de cobertura de tráfico, seleccione Actualizar la cobertura de supervisión. En el cuadro de diálogo, seleccione un porcentaje de tráfico y, a continuación, seleccione Actualizar la cobertura del monitor.

Elegir un límite máximo entre ciudades y redes

Amazon CloudWatch Internet Monitor puede supervisar el tráfico de las aplicaciones en algunas o todas las ubicaciones en las que los clientes acceden a los recursos de la aplicación y en todos los ASN (normalmente proveedores de servicios de Internet) a través de los que acceden a la aplicación, es decir, las redes urbanas para el tráfico de Internet de la aplicación. Al crear el monitor, usted elige el porcentaje del tráfico de aplicaciones que desea supervisar, y puede actualizarlo en cualquier momento editando el monitor.

Además de establecer un porcentaje de tráfico, también puede establecer un límite máximo para la cantidad de redes urbanas supervisadas. En esta sección se describe cómo el límite de las redes urbanas puede ayudarle a administrar los costos de facturación y se proporciona información y un ejemplo para ayudarle a determinar el límite que debe establecer.

El límite máximo que fije para la cantidad de redes de ciudades ayuda a garantizar que su factura sea previsible. Para obtener más información, consulte Precios de Amazon CloudWatch. También puede obtener información sobre cómo los diferentes valores del número de redes urbanas realmente supervisadas pueden afectar a su factura mediante la calculadora de precios de CloudWatch. Para explorar las opciones, en la página Calculadora de precios para CloudWatch, desplácese hacia abajo hasta Internet Monitor.

Para actualizar el monitor y cambiar el límite máximo de redes urbanas, consulte Edición o eliminación de un monitor de Internet Monitor.

Cómo funciona la facturación con los límites máximos de las redes urbanas

Establecer un límite máximo para la cantidad de redes urbanas supervisadas puede ayudar a evitar costos inesperados en la factura. Esto resulta útil, por ejemplo, si los patrones de tráfico varían mucho. Los costes de facturación aumentan por cada red urbana que se supervisa después de incluir las 100 primeras redes urbanas (en todos los monitores de cada cuenta). Si establece un límite máximo para las redes urbanas, se fija un límite para el número de redes urbanas que Internet Monitor supervisa para su aplicación, independientemente del porcentaje de tráfico que decida supervisar.

Solo pagará por la cantidad de redes urbanas que realmente estén supervisadas. El límite máximo de redes urbanas que elija le permite establecer un límite al total que se puede incluir cuando Internet Monitor supervise el tráfico con su monitor. Puede cambiar el límite máximo en cualquier momento editando su monitor.

Para explorar las opciones, en la página Calculadora de precios para CloudWatch, desplácese hacia abajo hasta Internet Monitor. Para obtener más información sobre los precios de Internet Monitor, consulte la sección Internet Monitor en la página de precios de Amazon CloudWatch.

Cómo elegir un límite máximo entre ciudades y redes

Para ayudarle a decidir el límite máximo que desea seleccionar en las redes urbanas, considere la cantidad de tráfico que desea supervisar para su aplicación. Las siguientes métricas de Internet Monitor pueden ayudarle a analizar el uso y la cobertura del tráfico después de crear el monitor: `CityNetworksMonitored`, `TrafficMonitoredPercent`, y una o más de las métricas `CityNetworksFor`*`NN`*`PercentTraffic`, donde *`NN`* es un valor porcentual que es uno de los siguientes: 25, 50, 90, 95, 99 o 100. Para revisar las definiciones de estas métricas y de todas las demás métricas de Internet Monitor, consulte Uso de las métricas de CloudWatch con Amazon CloudWatch Internet Monitor.

Para ver un gráfico general de la cobertura del tráfico de Internet, vaya a la pestaña Estadísticas del tráfico del panel de CloudWatch y, en la sección Gráficos de tráfico de Internet, elija una opción para Comparar opciones de cobertura de tráfico. El gráfico que se muestra en la sección muestra el número real de redes urbanas que se están supervisando para su aplicación, así como líneas gráficas para los diferentes porcentajes de tráfico de la aplicación que seleccione en la lista desplegable. Para obtener más información, consulte Configurar el porcentaje de tráfico de aplicaciones.

Para explorar sus opciones con más detalle, puede usar las métricas de Internet Monitor, tal y como se describe en los siguientes ejemplos. Estos ejemplos muestran cómo seleccionar el límite máximo

de redes urbanas que mejor se adapte a sus necesidades, en función de la amplitud de cobertura de tráfico de Internet de la aplicación que desee. El uso de las consultas para las métricas de Internet Monitor en CloudWatch Metrics puede ayudarle a comprender mejor la cobertura del tráfico de Internet de su aplicación.

Ejemplo de cómo determinar el límite máximo de una red urbana

Como ejemplo, supongamos que ha establecido un límite máximo de supervisión de 100 redes urbanas y que los clientes de 2637 redes urbanas acceden a su aplicación. En CloudWatch Metrics, verá que se devuelven las siguientes métricas de Internet Monitor:

```
CityNetworksMonitored 100
TrafficMonitoredPercent  12.5
CityNetworksFor90PercentTraffic  2143
CityNetworksFor100PercentTraffic  2637
```

En este ejemplo, puede ver que actualmente está supervisando el 12,5 % de su tráfico de Internet, con el límite máximo establecido en 100 redes urbanas. Si quiere supervisar el 90 % de su tráfico, la siguiente métrica proporciona información al respecto: `CityNetworksFor90PercentTraffic` indica que tendría que supervisar 2143 redes urbanas para obtener una cobertura del 90 %. Para ello, debe actualizar el monitor y establecer el límite máximo de redes urbanas en 2.143.

Del mismo modo, supongamos que desea que su aplicación controle el tráfico de Internet al 100 %. La siguiente métrica, `CityNetworksFor100PercentTraffic`, indica que, para ello, debe actualizar el monitor para establecer el límite máximo de redes urbanas en 2637.

Si ahora establece el máximo en 5000 redes urbanas, como es superior a 2637, verá las siguientes métricas:

```
CityNetworksMonitored 2637
TrafficMonitoredPercent  100
CityNetworksFor90PercentTraffic  2143
CityNetworksFor100PercentTraffic  2637
```

A partir de estas métricas, puede ver que, con el límite más alto, se supervisan las 2637 redes urbanas, es decir, el 100 % de su tráfico de Internet.

Publicar mediciones de Internet en Amazon S3 en Amazon CloudWatch Internet Monitor

Puede elegir que Amazon CloudWatch Internet Monitor publique en Amazon S3 las mediciones de Internet del tráfico dirigido a Internet en las redes urbanas supervisadas (ubicaciones de clientes y ASN, normalmente proveedores de servicios de Internet) de su monitor, hasta el límite de servicio de 500 000 redes urbanas. Internet Monitor publica automáticamente en registros de CloudWatch cada cinco minutos las mediciones de Internet de las 500 principales ciudades urbanas (por volumen de tráfico) de cada monitor. Las mediciones que publica en S3 incluyen las 500 principales que se publican en registros de CloudWatch.

Al crear o actualizar el monitor, puede elegir la opción de publicar en S3 y especificar el bucket en el que desea publicar las mediciones. El bucket ya debe estar creado en S3 para poder especificarlo en Internet Monitor. Hay un límite de servicio de 500 000 redes urbanas para las mediciones de Internet publicadas en S3. Internet Monitor publica las mediciones de Internet en S3 como eventos, una serie de objetos de archivo de registro comprimidos que se almacenan en el bucket.

Al crear el bucket de S3 para que Internet Monitor publique las mediciones, asegúrese de seguir las instrucciones de permisos proporcionadas por registros de CloudWatch. De este modo, se garantiza que Internet Monitor pueda publicar los registros directamente en S3 y que, de ser necesario, AWS pueda crear y cambiar las políticas de recursos asociadas al grupo de registro que los reciba. Para obtener más información, consulte Registros enviados a registros CloudWatch en la Guía del usuario de Registros de Amazon CloudWatch.

Los archivos de registro publicados están comprimidos. Si abre los archivos de registro con la consola de Amazon S3, se descomprimirán y se mostrarán los eventos de medición de Internet. Si descarga los archivos, deberá descomprimirlos para ver los eventos.

También puede consultar las mediciones de Internet en los archivos de registro mediante Amazon Athena. Amazon Athena es un servicio de consultas interactivo que facilita el análisis de datos en Amazon S3 con SQL estándar. Para obtener más información, consulte Usar Amazon Athena para consultar las mediciones de Internet en los archivos de registro de Amazon S3.

Uso de un monitor de Internet Monitor

Hay varias formas de utilizar un monitor de Amazon CloudWatch Internet Monitor después de crearlo: por ejemplo, puede ver información en el panel de control de CloudWatch, obtener información mediante AWS Command Line Interface y configurar alertas de estado.

El monitor proporciona información sobre la aplicación y las preferencias de configuración para que Internet Monitor pueda personalizar las mediciones y las métricas y publicarlas en eventos por usted. Internet Monitor recopila las mediciones de la infraestructura global de AWS. Estas mediciones representan una enorme cantidad de información de rendimiento y disponibilidad de red de todo el mundo. Al utilizar la información de los recursos que se añaden a la aplicación, Internet Monitor publica las medidas de rendimiento y disponibilidad que se refieren a las redes urbanas (es decir, las ubicaciones de los clientes y las ASN, normalmente proveedores de servicios de Internet o ISP) en las que está activa la aplicación. Por lo tanto, las medidas y métricas del panel de control de Internet Monitor y de registros de CloudWatch (sobre la disponibilidad, el rendimiento, los bytes supervisados transferidos y el tiempo de ida y vuelta) son específicas de las ubicaciones de sus clientes y los ASN.

Internet Monitor también determina cuándo hay anomalías en el rendimiento y la disponibilidad. De forma predeterminada, Internet Monitor superpone el tráfico con las mediciones de disponibilidad y rendimiento que AWS ha recopilado para cada par de origen y destino en las ubicaciones de sus clientes, a fin de determinar cuándo se producen caídas notables en el rendimiento o la disponibilidad. Cuando se produce una degradación significativa en las ubicaciones y el alcance de la aplicación, Internet Monitor genera un evento de estado y publica información sobre el problema en el monitor.

Después de crear un monitor, puede utilizarlo para acceder a la información que proporcione Internet Monitor o recibir alertas sobre ella, de las siguientes maneras:

- Utilice el panel de CloudWatch para ver y explorar los eventos de rendimiento, disponibilidad y estado; explorar los datos históricos de su aplicación y obtener información sobre nuevas formas de configurar su aplicación para obtener un mejor rendimiento. Para obtener más información, consulte lo siguiente:
  - Seguimiento del rendimiento y la disponibilidad en tiempo real en Amazon CloudWatch Internet Monitor (pestaña Overview [Descripción general])
  - Filtrado y visualización de datos en Amazon CloudWatch Internet Monitor (pestaña Explorador histórico)
  - Obtener información para mejorar el rendimiento de las aplicaciones en Amazon CloudWatch Internet Monitor (pestaña Estadísticas del tráfico)
- Configure los umbrales de eventos de estado para cambiar lo que desencadena que Internet Monitor cree un evento de estado para su aplicación. Puede configurar los umbrales generales y los umbrales locales (red urbana). Para obtener más información, consulte Cambiar los umbrales de eventos de estado.

- Utilice comandos de AWS CLI con las operaciones de la API de Internet Monitor para ver la información del perfil de tráfico, ver las mediciones, enumerar los eventos de estado, etc. Para obtener más información, consulte Ejemplos de uso de la CLI con Amazon CloudWatch Internet Monitor.
- Utilice las herramientas estándar de CloudWatch, como CloudWatch Contributor Insights, el explorador de CloudWatch Metrics y CloudWatch Logs Insights, para visualizar los datos en CloudWatch. Para obtener más información, consulte Exploración de datos mediante las herramientas de CloudWatch y la interfaz de consulta de Internet Monitor.
- Utilice Athena con los registros de S3 para acceder y analizar las mediciones de Internet de Internet Monitor para su aplicación, si ha activado la publicación de las mediciones en S3.
- Cree notificaciones de Amazon EventBridge para avisarle cuando Internet Monitor emita eventos de estado. Para obtener más información, consulte Uso de Amazon CloudWatch Internet Monitor con Amazon EventBridge.
- Reciba una notificación de AWS Health Dashboard automáticamente cuando Internet Monitor determine que un problema está causado por la red AWS. La notificación incluye las medidas que AWS está tomando para mitigar el problema.

Edición o eliminación de un monitor de Internet Monitor

Con el menú Acción, puede editar o eliminar un monitor en Amazon CloudWatch Internet Monitor después de crearlo. Por ejemplo, puede editar un monitor para hacer lo siguiente:

- Cambiar el porcentaje de tráfico de aplicaciones que se va a supervisar
- Establecer o actualizar el límite máximo de las redes urbanas
- Cambiar los umbrales de eventos de estado para comprobar la disponibilidad o las puntuaciones de rendimiento
- Añadir o eliminar tipos de recursos
- Habilitar o actualizar eventos de publicación en Amazon S3

También puede eliminar un monitor. Tenga en cuenta que no puede cambiar el nombre de un monitor después de crearlo.

Para realizar cambios o eliminar un monitor, utilice alguno de los siguientes procedimientos.

Editar un monitor

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación izquierdo, en Supervisión de redes, elija Monitor de Internet.
3. Elija su monitor y, a continuación, elija el menú Acción.
4. Seleccione Actualizar monitor.
5. Realice las actualizaciones que desee. Por ejemplo, para cambiar el porcentaje de tráfico que se va a supervisar, en Tráfico de aplicaciones que se va a supervisar, seleccione o introduzca un porcentaje.
6. Elija Actualizar.

Para eliminar un monitor

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación izquierdo, en Supervisión de redes, elija Monitor de Internet.
3. Elija su monitor y, a continuación, elija el menú Acción.
4. Elija Deshabilitar.
5. Vuelva a seleccionar el menú Acción y, a continuación, elija Eliminar.

Para obtener más información sobre las opciones que se puedan actualizar, consulte lo siguiente:

- Para obtener más información sobre los recursos que se agregan a Internet Monitor, consulte Agregar recursos a su monitor.
- Para saber más sobre el porcentaje de tráfico de aplicaciones, consulte Elegir el porcentaje de tráfico de aplicaciones que se va a supervisar.
- Para obtener más información sobre cómo cambiar el umbral de los eventos de estado, consulte Cambiar los umbrales de los eventos de estado.
- Para obtener más información sobre el límite máximo de redes urbanas, consulte Elegir un límite máximo entre ciudades y redes.
- Para obtener más información sobre cómo optar por publicar eventos en S3, consulte Publicar mediciones de Internet en Amazon S3 en Amazon CloudWatch Internet Monitor.

Agregar o crear un monitor Amazon CloudWatch Internet Monitor mediante Amazon VPC

Cuando cree una nube privada virtual VPC de Amazon en la AWS Management Console, si lo desea, puede optar por configurar también su supervisión en Amazon CloudWatch Internet Monitor. Puede agregar la VPC a un monitor existente o puede optar por crear un nuevo monitor para la VPC en la consola de Amazon VPC.

Cuando usa Internet Monitor con su VPC, puede ver y evaluar las medidas y las métricas sobre la disponibilidad, el rendimiento, los bytes supervisados transferidos y los tiempos de ida y vuelta específicos de las ubicaciones de los clientes y los ASN de su aplicación (normalmente, proveedores de servicios de Internet). Internet Monitor también determina cuándo hay anomalías en el rendimiento y la disponibilidad y crea eventos de estado en el monitor, de los que puede elegir recibir notificaciones. Para obtener más información sobre cómo puede utilizar un monitor para administrar y mejorar la experiencia de sus clientes con su aplicación, consulte Uso de un monitor de Internet Monitor.

> **Important**
>
> Para crear un monitor o agregar una VPC a un monitor existente, debe disponer de los permisos correctos. Para obtener más información, consulte Identity and Access Management para Amazon CloudWatch Internet Monitor.

Agregar una VPC a un monitor existente

Puede elegir que Amazon CloudWatch Internet Monitor agregue automáticamente una nueva VPC a un monitor existente cuando crea la VPC en la AWS Management Console. Después de agregar la VPC, espere unos minutos y, a continuación, las métricas de la VPC comenzarán a mostrarse en la consola de Internet Monitor.

Puede editar el monitor en cualquier momento para eliminar la VPC o agregar otra VPC u otros recursos. También puede modificar el porcentaje de tráfico que está supervisando o hacer otros cambios. Si decide eliminar la VPC del monitor, Internet Monitor dejará de supervisar el tráfico de los clientes a esa VPC.

Para obtener más información sobre la actualización de un monitor, consulte Edición o eliminación de un monitor de Internet Monitor.

Crear un monitor para una VPC

Si opta por crear un monitor para una VPC, el asistente Crear monitor indicará los pasos a seguir. Agregue la VPC como recurso supervisado al crear el monitor. Si lo desea, también puede elegir el porcentaje de tráfico de clientes que desee supervisar para su aplicación (el valor predeterminado es el 100 %).

Para obtener más información, consulte los detalles en Cree un monitor en Amazon CloudWatch Internet Monitor mediante la consola.

Precios

Con Amazon CloudWatch Internet Monitor solo paga por lo que usa. El precio de Internet Monitor tiene dos componentes: una tarifa por recurso supervisado y una tarifa por red urbana. Una red urbana es la ubicación desde la que los clientes acceden a los recursos de la aplicación y la red (un ASN, como un proveedor de servicios de Internet o un ISP) desde la que los clientes acceden a los recursos.

Para obtener más información, incluidos algunos ejemplos de precios, consulte Precios de Amazon CloudWatch Internet Monitor.

Dejar de supervisar una VPC

Si desea dejar de supervisar su recurso de VPC con Internet Monitor, haga lo siguiente en la consola de Internet Monitor:

Para eliminar un recurso de un monitor

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación izquierdo, en Supervisión de redes, elija Monitor de Internet.
3. Elija su monitor y, a continuación, elija el menú Acción.
4. Seleccione Actualizar monitor.
5. En Recursos agregados, seleccione Eliminar recursos.
6. Elija la VPC que desea eliminar y, a continuación, elija Eliminar.
7. Elija Actualizar.

Agregación o creación de un monitor de Amazon CloudWatch Internet Monitor con CloudFront

En el panel de métricas de una distribución en la consola de Amazon CloudFront, puede configurar una supervisión adicional para una distribución en Amazon CloudWatch Internet Monitor. Puede agregar la distribución a un monitor existente o puede crear un nuevo monitor para la distribución.

Cuando usa Internet Monitor con su distribución de CloudFront, puede ver y evaluar las medidas y las métricas sobre la disponibilidad, el rendimiento, los bytes supervisados transferidos y los tiempos de ida y vuelta específicos de las ubicaciones de los clientes y los ASN de su aplicación (normalmente, proveedores de servicios de Internet). Internet Monitor también determina cuándo hay anomalías en el rendimiento y la disponibilidad y crea eventos de estado en el monitor, de los que puede elegir recibir notificaciones. Para obtener más información sobre cómo puede utilizar un monitor para administrar y mejorar la experiencia de sus clientes con su aplicación, consulte Uso de un monitor de Internet Monitor.

> **Important**
>
> Para crear un monitor o agregar una distribución a un monitor existente, debe disponer de los permisos correctos. Para obtener más información, consulte Identity and Access Management para Amazon CloudWatch Internet Monitor.

Agregación de una distribución a un monitor existente

Puede elegir que Internet Monitor agregue una distribución a un monitor existente directamente desde el panel de control de métricas de CloudFront de la AWS Management Console. Después de agregar la distribución, espere unos minutos y, a continuación, las métricas de la distribución comenzarán a mostrarse en la consola de Internet Monitor.

Puede editar el monitor en cualquier momento para eliminar la distribución o agregar otra distribución u otros recursos. También puede modificar el porcentaje de tráfico que está supervisando o hacer otros cambios. Si decide eliminar la distribución del monitor, Internet Monitor dejará de supervisar el tráfico de los clientes a esa distribución.

Para obtener más información sobre la actualización de un monitor, consulte Edición o eliminación de un monitor de Internet Monitor.

Creación de un monitor para una distribución

Si opta por crear un monitor para una distribución, el asistente Crear monitor indicará los pasos que debe seguir. Agregue la distribución como recurso supervisado al crear el monitor. Si lo desea, también puede elegir el porcentaje de tráfico de clientes que desee supervisar para su aplicación (el valor predeterminado es el 100 %).

Para obtener más información, consulte los detalles en Cree un monitor en Amazon CloudWatch Internet Monitor mediante la consola.

Precios

Con Amazon CloudWatch Internet Monitor solo paga por lo que usa. El precio de Internet Monitor tiene dos componentes: una tarifa por recurso supervisado y una tarifa por red urbana. Una red urbana es la ubicación desde la que los clientes acceden a los recursos de la aplicación y la red (un ASN, como un proveedor de servicios de Internet o un ISP) desde la que los clientes acceden a los recursos.

Para obtener más información, incluidos algunos ejemplos de precios, consulte Precios de Amazon CloudWatch Internet Monitor.

Desactivación de la supervisión de una distribución

Si desea dejar de supervisar su recurso de distribución con Internet Monitor, haga lo siguiente en la consola de Internet Monitor:

Para eliminar un recurso de un monitor

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación izquierdo, en Supervisión de redes, elija Monitor de Internet.
3. Elija su monitor y, a continuación, elija el menú Acción.
4. Seleccione Actualizar monitor.
5. En Recursos agregados, seleccione Eliminar recursos.
6. Elija la distribución que desea eliminar y, a continuación, elija Eliminar.
7. Elija Actualizar.

Ejemplos de uso de la CLI con Amazon CloudWatch Internet Monitor

En esta sección se incluyen ejemplos de uso de AWS Command Line Interface con las operaciones de Amazon CloudWatch Internet Monitor.

Antes de empezar, asegúrese de iniciar sesión para utilizar la AWS CLI con la misma cuenta de AWS que tiene las nubes privadas virtuales (VPC) de Amazon, los Equilibradores de carga de red, las distribuciones de Amazon CloudFront o los directorios de Amazon WorkSpaces que desea supervisar. Internet Monitor no admite el acceso a los recursos entre cuentas. Para obtener más información sobre el uso de la AWS CLI, consulte Referencia de comandos de la AWS CLI. Para obtener más información sobre el uso de las acciones de la API con Amazon CloudWatch Internet Monitor, consulte la Amazon CloudWatch Internet Monitor API Reference Guide (Guía de referencia de la API de Amazon CloudWatch Internet Monitor).

Temas

Creación de un monitor

Al crear un monitor en Internet Monitor, proporciona un nombre y asocie recursos al monitor para mostrar dónde está el tráfico de Internet de la aplicación. Usted especifica un porcentaje de tráfico que define qué parte del tráfico de su aplicación se supervisa. Esto también determina la cantidad de redes urbanas, es decir, las ubicaciones de los clientes y los ASN, normalmente proveedores de servicios de Internet o ISP, que se supervisan. También puede optar por establecer un límite para el número máximo de ciudades urbanas que supervisar para los recursos de su aplicación a fin de ayudarle a controlar su factura. Para obtener más información, consulte Elegir un límite máximo entre ciudades y redes.

Por último, puede elegir si quiere publicar todas las mediciones de Internet de su aplicación en Amazon S3. Las mediciones de Internet de las 500 principales redes urbanas (por volumen de

tráfico) se publican automáticamente en registros de CloudWatch de Internet Monitor, pero también puede optar por publicar todas las mediciones en S3.

Para crear un alojamiento con la AWS CLI, utilice el comando `create-monitor`. El siguiente comando crea un monitor que supervisa el 100 % del tráfico, pero establece un límite máximo de 10 000 redes urbanas, añade un recurso de VPC y opta por publicar las mediciones de Internet en Amazon S3.

> **Note**
>
> Internet Monitor publica las mediciones de Internet en registros de CloudWatch cada cinco minutos para las 500 principales redes urbanas (ubicaciones de clientes y ASN, normalmente proveedores de servicios de Internet o ISP) que envían tráfico a cada monitor. Si lo desea, puede optar por publicar las mediciones de Internet de todas las redes urbanas supervisadas (hasta el límite de servicio de 500 000 redes urbanas) en un bucket de Amazon S3. Para obtener más información, consulte Publicar mediciones de Internet en Amazon S3 en Amazon CloudWatch Internet Monitor.

```
aws internetmonitor --create-monitor monitor-name "TestMonitor" \
    --traffic-percentage-to-monitor 100 \
    --max-city-networks-to-monitor 10000 \
    --resources "arn:aws:ec2:us-east-1:111122223333:vpc/vpc-11223344556677889" \
    --internet-measurements-log-delivery
 S3Config="{BucketName=MyS3Bucket,LogDeliveryStatus=ENABLED}"
```

```
{
"Arn": "arn:aws:internetmonitor:us-east-1:111122223333:monitor/TestMonitor",
 "Status": "ACTIVE"
}
```

> **Note**
>
> No se puede cambiar el nombre del monitor.

Consultar detalles de supervisión

Para ver información sobre un monitor con la AWS CLI, utilice el comando `get-monitor`.

```
aws internetmonitor get-monitor --monitor-name "TestMonitor"
```

```
{
    "ClientLocationType": "city",
    "CreatedAt": "2022-09-22T19:27:47Z",
    "ModifiedAt": "2022-09-22T19:28:30Z",
    "MonitorArn": "arn:aws:internetmonitor:us-east-1:111122223333:monitor/TestMonitor",
    "MonitorName": "TestMonitor",
    "ProcessingStatus": "OK",
    "ProcessingStatusInfo": "The monitor is actively processing data",
    "Resources": [
        "arn:aws:ec2:us-east-1:111122223333:vpc/vpc-11223344556677889"
    ],
    "MaxCityNetworksToMonitor": 10000,
    "Status": "ACTIVE"
}
```

Enumeración de eventos de estado

Cuando el rendimiento del tráfico de Internet de la aplicación disminuye, Internet Monitor crea eventos de estado en el monitor. Para ver una lista de los eventos de estado actuales con la AWS CLI, utilice el comando `list-health-events`

```
aws internetmonitor list-health-events --monitor-name "TestMonitor"
```

```
{
    "HealthEvents": [
        {
            "EventId": "2022-06-20T01-05-05Z/latency",
            "Status": "RESOLVED",
            "EndedAt": "2022-06-20T01:15:14Z",
            "ServiceLocations": [
                {
                    "Name": "us-east-1"
                }
            ],
            "PercentOfTotalTrafficImpacted": 1.21,
            "ClientLocations": [
                {
                    "City": "Lockport",
                    "PercentOfClientLocationImpacted": 60.370000000000005,
```

```
                    "PercentOfTotalTraffic": 2.01,
                    "Country": "United States",
                    "Longitude": -78.6913,
                    "AutonomousSystemNumber": 26101,
                    "Latitude": 43.1721,
                    "Subdivision": "New York",
                    "NetworkName": "YAHOO-BF1"
                }
            ],
            "StartedAt": "2022-06-20T01:05:05Z",
            "ImpactType": "PERFORMANCE",
            "EventArn": "arn:aws:internetmonitor:us-east-1:111122223333:monitor/
TestMonitor/health-event/2022-06-20T01-05-05Z/latency"
        },
        {
            "EventId": "2022-06-20T01-17-56Z/latency",
            "Status": "RESOLVED",
            "EndedAt": "2022-06-20T01:30:23Z",
            "ServiceLocations": [
                {
                    "Name": "us-east-1"
                }
            ],
            "PercentOfTotalTrafficImpacted": 1.29,
            "ClientLocations": [
                {
                    "City": "Toronto",
                    "PercentOfClientLocationImpacted": 75.32,
                    "PercentOfTotalTraffic": 1.05,
                    "Country": "Canada",
                    "Longitude": -79.3623,
                    "AutonomousSystemNumber": 14061,
                    "Latitude": 43.6547,
                    "Subdivision": "Ontario",
                    "CausedBy": {
                        "Status": "ACTIVE",
                        "Networks": [
                            {
                                "AutonomousSystemNumber": 16509,
                                "NetworkName": "Amazon.com"
                            }
                        ],
                        "NetworkEventType": "AWS"
                    },
```

```
                    "NetworkName": "DIGITALOCEAN-ASN"
                },
                {
                    "City": "Lockport",
                    "PercentOfClientLocationImpacted": 22.91,
                    "PercentOfTotalTraffic": 2.01,
                    "Country": "United States",
                    "Longitude": -78.6913,
                    "AutonomousSystemNumber": 26101,
                    "Latitude": 43.1721,
                    "Subdivision": "New York",
                    "NetworkName": "YAHOO-BF1"
                },
                {
                    "City": "Hangzhou",
                    "PercentOfClientLocationImpacted": 2.88,
                    "PercentOfTotalTraffic": 0.7799999999999999,
                    "Country": "China",
                    "Longitude": 120.1612,
                    "AutonomousSystemNumber": 37963,
                    "Latitude": 30.2994,
                    "Subdivision": "Zhejiang",
                    "NetworkName": "Hangzhou Alibaba Advertising Co.,Ltd."
                }
            ],
            "StartedAt": "2022-06-20T01:17:56Z",
            "ImpactType": "PERFORMANCE",
            "EventArn": "arn:aws:internetmonitor:us-east-1:111122223333:monitor/
TestMonitor/health-event/2022-06-20T01-17-56Z/latency"
        },
        {
            "EventId": "2022-06-20T01-34-20Z/latency",
            "Status": "RESOLVED",
            "EndedAt": "2022-06-20T01:35:04Z",
            "ServiceLocations": [
                {
                    "Name": "us-east-1"
                }
            ],
            "PercentOfTotalTrafficImpacted": 1.15,
            "ClientLocations": [
                {
                    "City": "Lockport",
                    "PercentOfClientLocationImpacted": 39.45,
```

```
                    "PercentOfTotalTraffic": 2.01,
                    "Country": "United States",
                    "Longitude": -78.6913,
                    "AutonomousSystemNumber": 26101,
                    "Latitude": 43.1721,
                    "Subdivision": "New York",
                    "NetworkName": "YAHOO-BF1"
                },
                {
                    "City": "Toronto",
                    "PercentOfClientLocationImpacted": 29.770000000000003,
                    "PercentOfTotalTraffic": 1.05,
                    "Country": "Canada",
                    "Longitude": -79.3623,
                    "AutonomousSystemNumber": 14061,
                    "Latitude": 43.6547,
                    "Subdivision": "Ontario",
                    "CausedBy": {
                        "Status": "ACTIVE",
                        "Networks": [
                            {
                                "AutonomousSystemNumber": 16509,
                                "NetworkName": "Amazon.com"
                            }
                        ],
                        "NetworkEventType": "AWS"
                    },
                    "NetworkName": "DIGITALOCEAN-ASN"
                },
                {
                    "City": "Hangzhou",
                    "PercentOfClientLocationImpacted": 2.88,
                    "PercentOfTotalTraffic": 0.7799999999999999,
                    "Country": "China",
                    "Longitude": 120.1612,
                    "AutonomousSystemNumber": 37963,
                    "Latitude": 30.2994,
                    "Subdivision": "Zhejiang",
                    "NetworkName": "Hangzhou Alibaba Advertising Co.,Ltd."
                }
            ],
            "StartedAt": "2022-06-20T01:34:20Z",
            "ImpactType": "PERFORMANCE",
```

```
            "EventArn": "arn:aws:internetmonitor:us-east-1:111122223333:monitor/
TestMonitor/health-event/2022-06-20T01-34-20Z/latency"
        }
    ]
}
```

Consulta de un evento de estado específico

Para ver información más detallada sobre un evento de estado específico con la CLI, ejecute el comando `get-health-event` con el nombre del monitor y el ID del evento de estado.

```
aws internetmonitor get-monitor --monitor-name "TestMonitor" --event-id "health-event/
TestMonitor/2021-06-03T01:02:03Z/latency"
```

```
{
    "EventId": "2022-06-20T01-34-20Z/latency",
    "Status": "RESOLVED",
    "EndedAt": "2022-06-20T01:35:04Z",
    "ServiceLocations": [
        {
            "Name": "us-east-1"
        }
    ],
    "EventArn": "arn:aws:internetmonitor:us-east-1:111122223333:monitor/TestMonitor/
health-event/2022-06-20T01-34-20Z/latency",
    "LastUpdatedAt": "2022-06-20T01:35:04Z",
    "ClientLocations": [
        {
            "City": "Lockport",
            "PercentOfClientLocationImpacted": 39.45,
            "PercentOfTotalTraffic": 2.01,
            "Country": "United States",
            "Longitude": -78.6913,
            "AutonomousSystemNumber": 26101,
            "Latitude": 43.1721,
            "Subdivision": "New York",
            "NetworkName": "YAHOO-BF1"
        },
        {
            "City": "Toronto",
            "PercentOfClientLocationImpacted": 29.770000000000003,
            "PercentOfTotalTraffic": 1.05,
            "Country": "Canada",
```

```
            "Longitude": -79.3623,
            "AutonomousSystemNumber": 14061,
            "Latitude": 43.6547,
            "Subdivision": "Ontario",
            "CausedBy": {
                "Status": "ACTIVE",
                "Networks": [
                    {
                        "AutonomousSystemNumber": 16509,
                        "NetworkName": "Amazon.com"
                    }
                ],
                "NetworkEventType": "AWS"
            },
            "NetworkName": "DIGITALOCEAN-ASN"
        },
        {
            "City": "Shenzhen",
            "PercentOfClientLocationImpacted": 4.07,
            "PercentOfTotalTraffic": 0.61,
            "Country": "China",
            "Longitude": 114.0683,
            "AutonomousSystemNumber": 37963,
            "Latitude": 22.5455,
            "Subdivision": "Guangdong",
            "NetworkName": "Hangzhou Alibaba Advertising Co.,Ltd."
        },
        {
            "City": "Hangzhou",
            "PercentOfClientLocationImpacted": 2.88,
            "PercentOfTotalTraffic": 0.7799999999999999,
            "Country": "China",
            "Longitude": 120.1612,
            "AutonomousSystemNumber": 37963,
            "Latitude": 30.2994,
            "Subdivision": "Zhejiang",
            "NetworkName": "Hangzhou Alibaba Advertising Co.,Ltd."
        }
    ],
    "StartedAt": "2022-06-20T01:34:20Z",
    "ImpactType": "PERFORMANCE",
    "PercentOfTotalTrafficImpacted": 1.15
}
```

Consulta de la lista de monitores

Para ver una lista de todos los monitores de su cuenta con la CLI, ejecute el comando `list-monitors`.

```
aws internetmonitor list-monitors
```

```
{
    "Monitors": [
        {
            "MonitorName": "TestMonitor",
            "ProcessingStatus": "OK",
            "Status": "ACTIVE"
        }
    ],
    "NextToken": " zase12"
}
```

Edición de un monitor

Para actualizar la información sobre el monitor mediante la CLI, utilice el comando `update-monitor` y especifique el nombre del monitor que se va a actualizar. Puede actualizar el porcentaje de tráfico que se debe supervisar, el límite del número máximo de redes urbanas que se deben supervisar, añadir o eliminar los recursos que Internet Monitor utiliza para supervisar el tráfico y cambiar el estado del monitor de `ACTIVE` a `INACTIVE` o viceversa. Tenga en cuenta que no puede cambiar el nombre del monitor.

La respuesta de una llamada `update-monitor` devuelve solo el `MonitorArn` y el `Status`.

En el siguiente ejemplo, se muestra cómo se usa el comando `update-monitor` para cambiar el estado del monitor a `50000`:

```
aws internetmonitor update-monitor --monitor-name "TestMonitor" --max-city-networks-to-
monitor 50000
```

```
{
    "MonitorArn": "arn:aws:internetmonitor:us-east-1:111122223333:monitor/TestMonitor",
    "Status": " ACTIVE "
}
```

En el siguiente ejemplo, se muestra cómo agregar y eliminar recursos:

```
aws internetmonitor update-monitor --monitor-name "TestMonitor" \
    --resources-to-add "arn:aws:ec2:us-east-1:111122223333:vpc/vpc-11223344556677889" \
    --resources-to-remove "arn:aws:ec2:us-east-1:111122223333:vpc/vpc-2222444455556666"
```

```
{
    "MonitorArn": "arn:aws:internetmonitor:us-east-1:111122223333:monitor/TestMonitor",
    "Status": "ACTIVE"
}
```

En el siguiente ejemplo, se muestra cómo se usa el comando `update-monitor` para cambiar el estado del monitor a `INACTIVE`:

```
aws internetmonitor update-monitor --monitor-name "TestMonitor" --status "INACTIVE"
```

```
{
    "MonitorArn": "arn:aws:internetmonitor:us-east-1:111122223333:monitor/TestMonitor",
    "Status": "INACTIVE"
}
```

Eliminación de un monitor

Puede eliminar un monitor con la CLI mediante el comando `delete-monitor`. En primer lugar, debe configurar el monitor para que esté inactivo. Para ello, utilice el comando `update-monitor` para cambiar el estado a `INACTIVE`. Confirme que el monitor está inactivo mediante el comando `get-monitor` y comprobando el estado.

Cuando el estado del monitor sea `INACTIVE`, puede usar la CLI para ejecutar el comando `delete-monitor` para eliminar el monitor. La respuesta de una llamada `delete-monitor` exitosa está vacía.

```
aws internetmonitor delete-monitor --monitor-name "TestMonitor"
```

```
{}
```

Supervisión y optimización con el panel de control de Internet Monitor

La información de esta sección describe cómo filtrar y ver datos en el panel de Amazon CloudWatch Internet Monitor para visualizar y obtener información sobre el tráfico y la configuración de Internet de su aplicación de AWS.

Tras crear un monitor para supervisar el rendimiento y la disponibilidad de Internet de la aplicación, Amazon CloudWatch Internet Monitor publica los registros de CloudWatch que contienen las mediciones de Internet de los pares ubicación-red del cliente (ciudad-red), y publica métricas agregadas de CloudWatch sobre el tráfico a su aplicación y a cada región de Región de AWS y ubicación periférica. Puede filtrar, explorar y obtener sugerencias orientadas a la acción a partir de esta información de Internet Monitor de varias maneras diferentes.

Para empezar, en la consola de CloudWatch, en Supervisión de redes, seleccione Monitor de Internet.

En esta sección se describe principalmente cómo filtrar y ver las métricas de Internet Monitor mediante AWS Management Console. Como alternativa, puede utilizar las operaciones de la API de Internet Monitor con la AWS CLI o un SDK para trabajar directamente con los eventos de Internet Monitor almacenados en los archivos de registros de CloudWatch. Para obtener más información, consulte Uso del monitor y la información de las mediciones. Para obtener más información sobre el uso de las operaciones de la API, consulte Ejemplos de uso de la CLI con Amazon CloudWatch Internet Monitor y la Referencia de la API de Amazon CloudWatch Internet Monitor.

Hay tres pestañas en el panel de control de Internet Monitor:

- En la pestaña Overview (Descripción general), puede ver la información actual e histórica sobre el rendimiento y la disponibilidad de la aplicación y los eventos de estado que afectan a las ubicaciones de los clientes.
- En la siguiente pestaña, Explorador histórico, puede filtrar por ubicación, proveedor de red, fecha, etc., y visualizar las métricas del tráfico de Internet a lo largo del tiempo mediante los gráficos.
- En la pestaña Estadísticas del tráfico, además de ver la información sobre el tráfico principal supervisado resumida de varias formas personalizables, puede obtener sugerencias de configuraciones optimizadas a fin de mejorar el rendimiento de los diferentes pares de proveedores de redes y ubicaciones. Internet Monitor predice la mejora del rendimiento de la aplicación en función de los patrones de tráfico y el rendimiento anterior, cuando cambia la forma en que dirige el tráfico o los recursos de AWS que utilice. También puede ver un gráfico para comparar cuántas

redes urbanas están incluidas en su cobertura de supervisión, en función del porcentaje de tráfico de aplicaciones que elija para su monitor.

Además, dado que Internet Monitor genera y publica archivos de registro con las mediciones del tráfico, puede utilizar otras herramientas de CloudWatch en la consola para visualizar mejor los datos que ha publicado Internet Monitor, como Información de colaboradores de Amazon CloudWatch, las métricas de CloudWatch e Información de registros de Amazon CloudWatch. Para obtener más información, consulte Exploración de datos mediante las herramientas de CloudWatch y la interfaz de consulta de Internet Monitor.

Obtenga información sobre el uso de Internet Monitor para explorar sus medidas de rendimiento y disponibilidad en las siguientes secciones.

Temas

- Seguimiento del rendimiento y la disponibilidad en tiempo real en Amazon CloudWatch Internet Monitor (pestaña Overview [Descripción general])
- Filtrado y visualización de datos en Amazon CloudWatch Internet Monitor (pestaña Explorador histórico)
- Obtener información para mejorar el rendimiento de las aplicaciones en Amazon CloudWatch Internet Monitor (pestaña Estadísticas del tráfico)

Seguimiento del rendimiento y la disponibilidad en tiempo real en Amazon CloudWatch Internet Monitor (pestaña Overview [Descripción general])

En la consola de CloudWatch, en Internet Monitor, en la pestaña Descripción general, puede obtener una vista general del rendimiento y la disponibilidad del tráfico que rastrea su monitor. También se muestra un mapa general del tráfico de Internet, con clústeres de tráfico que pueden serle de ayuda para visualizar el tráfico global de su aplicación y la ubicación y el impacto de los eventos de estado.

Puntuaciones de estado

El gráfico Puntuaciones de estado muestra información sobre el rendimiento y la disponibilidad del tráfico global. AWS tiene datos históricos sustanciales sobre el rendimiento de Internet y la disponibilidad del tráfico de red entre ubicaciones geográficas para diferentes proveedores y servicios de AWS. Internet Monitor utiliza estos datos de conectividad que AWS ha recopilado de su huella en la red global para calcular una línea base de rendimiento y disponibilidad del tráfico

de Internet. Estos son los mismos datos que utilizamos en AWS para supervisar nuestro propio tiempo de actividad y la disponibilidad de Internet.

Con esas mediciones como punto de referencia, Internet Monitor puede detectar cuándo el rendimiento y la disponibilidad de la aplicación han disminuido, en comparación con el punto de referencia. Para que le resulte más fácil ver esas caídas, le informamos en forma de puntuaciones de rendimiento y puntuaciones de disponibilidad. Para obtener más información, consulte Exploración de datos mediante las herramientas de CloudWatch y la interfaz de consulta de Internet Monitor.

El gráfico de Puntuaciones de estado incluye los eventos de estado que ocurrieron durante el período de tiempo que elija. Cuando se produce un evento de estado, se observa una caída en la línea de rendimiento o disponibilidad del gráfico. Si selecciona el evento, verá más detalles y aparecerán bandas en el gráfico, con información de fecha y hora que muestra cuánto duró el evento.

También puede consultar estas métricas accediendo directamente a los archivos de registro de cada punto de datos. En el menú Actions (Acciones), elija View CloudWatch Logs (Ver registros de CloudWatch).

Descripción general del tráfico de Internet

El mapa de la descripción general del tráfico de Internet muestra el tráfico de Internet y los eventos de estado específicos de las ubicaciones y ASN desde las que sus usuarios acceden a su aplicación. Los países que aparecen en gris en el mapa son los que incluyen el tráfico de su aplicación.

Cada círculo del mapa indica un evento de estado en un área, en un período de tiempo que seleccione. Internet Monitor crea eventos de estado cuando detecta un problema, en un umbral específico, de conectividad entre uno de sus recursos alojados en AWS y el de la red de ciudad en el que el usuario acceda a la aplicación. Al elegir un círculo en el mapa, se muestran más detalles sobre el evento de estado de esa ubicación. Además, en el caso de los clústeres que tengan eventos de estado, puede ver información detallada en la tabla de eventos de estado situada debajo del mapa.

Internet Monitor crea y resuelve eventos de estado en un monitor cuando determina que un evento tiene un impacto global significativo en su aplicación. Si no hay ningún evento de estado que supere el umbral de impacto en el tráfico para las ubicaciones de los clientes en el periodo de tiempo que haya seleccionado, el mapa estará en blanco. Para obtener más información, consulte Cuándo Internet Monitor crea y resuelve eventos de estado.

Cambiar umbrales de eventos de estado

Puede configurar varias opciones sobre cómo y cuándo Internet Monitor crea eventos de estado para su aplicación. Seleccione Actualizar umbrales para realizar cambios.

Puede cambiar el umbral general que hace que Internet Monitor cree un problema de estado. De forma predeterminada, el umbral de eventos de estado es del 95 % tanto para las puntuaciones de rendimiento como para las de disponibilidad. Es decir, cuando la puntuación general de rendimiento o disponibilidad de la aplicación cae al 95 % o menos, Internet Monitor crea un evento de estado. En el caso del umbral general, el problema de estado puede estar provocado por un único problema importante o por la combinación de varios problemas más leves.

También puede cambiar el umbral local (es decir, el de la red urbana) y añadirlo a un porcentaje del nivel general de impacto, ya que, en conjunto, provocará un problema estado. Al establecer un umbral que genere un problema de estado cuando una puntuación caiga por debajo del umbral en una o más redes urbanas (ubicaciones y ASN, por lo general, proveedores de servicios de Internet), puede obtener información sobre cuándo se producen problemas en ubicaciones con menos tráfico, por ejemplo.

Una opción de umbral local adicional funciona junto con el umbral local para las puntuaciones de disponibilidad o rendimiento. El segundo factor es el porcentaje del tráfico total que debe verse afectado antes de que Internet Monitor genere un problema de estado basado en el umbral local.

Al configurar las opciones de umbral para el tráfico general y el tráfico local, puede ajustar la frecuencia con la que se crean eventos de estado para adaptarlos al uso de las aplicaciones y a sus necesidades. Tenga en cuenta que cuando establece que el umbral local sea más bajo, normalmente se crean más eventos de estado, en función de la aplicación y de los demás valores de configuración del umbral que establezca.

En resumen, puede configurar los umbrales de eventos de estado (para las puntuaciones de rendimiento, las puntuaciones de disponibilidad o ambas) de las siguientes maneras:

- Elija diferentes umbrales globales para desencadenar un evento de estado.
- Elija diferentes umbrales locales para desencadenar un evento de estado. Con esta opción, también puede cambiar el porcentaje de impacto en toda la aplicación que debe superarse antes de que Internet Monitor cree un evento.
- Puede deshabilitar la activación de un evento de estado en función de los umbrales locales o activar las opciones de umbrales locales.

También puede configurar opciones para las puntuaciones de rendimiento, las puntuaciones de disponibilidad o ambas. Puede configurar una combinación de las opciones o solo una de ellas.

Para actualizar los umbrales y otras opciones de configuración para las puntuaciones de rendimiento, las puntuaciones de disponibilidad o ambas, haga lo siguiente:

Para cambiar las opciones de configuración de los umbrales

1. En la AWS Management Console, vaya a CloudWatch y, a continuación, en el panel de navegación de la izquierda, seleccione Internet Monitor.
2. En la pestaña Descripción general, en la sección Cronología de eventos de estado, seleccione Actualizar umbrales.
3. En la página de diálogo que se abre, elija los nuevos valores y opciones que desee para los umbrales y otras opciones que hagan que Internet Monitor cree un evento de estado. Puede elegir cualquiera de las opciones siguientes:
   - Elija un nuevo valor para el umbral de puntuación de disponibilidad, el umbral de puntuación de rendimiento o ambos.

     Los gráficos de las secciones de cada configuración muestran la configuración del umbral actual y las puntuaciones reales de los eventos de estado recientes en función de la disponibilidad o el rendimiento de la aplicación. Al ver los valores típicos, puede hacerse una idea de los valores por los que podría querer cambiar un umbral.

     Consejo: para ver un gráfico más grande y cambiar el período de tiempo, elija el expansor en la esquina superior derecha del gráfico.
   - Elija activar o desactivar un umbral local de disponibilidad o rendimiento, o ambos. Cuando una opción está habilitada, puede establecer el umbral y el nivel de impacto para cuando desee que Internet Monitor cree un evento de estado.
4. Tras configurar las opciones de umbral, guarde las actualizaciones seleccionando Actualizar los umbrales de los eventos de estado.

Para obtener más información sobre cómo funcionan los eventos de estado, consulte Cuándo Internet Monitor crea y resuelve eventos de estado.

Tabla de eventos de estado

La Tabla de eventos de estado muestra las ubicaciones de los clientes que se han visto afectadas por eventos de estado e información sobre los eventos. En la tabla se incluyen las siguientes columnas.

	Descripción
Ubicación del cliente	La ubicación de los usuarios finales que se vieron afectados por el evento, que experimentaron un aumento de la latencia o una disponibilidad reducida. Para obtener más información sobre la precisión de la ubicación del cliente en Internet Monitor, consulte Información y precisión de la geolocalización en Internet Monitor.
Impacto en el tráfico	Qué impacto causó el evento, en términos de latencia aumentada o disponibilidad reducida. En cuanto a la latencia, se trata del porcentaj e de latencia que aumentó durante el evento en comparación con el rendimiento típico del tráfico, desde esta ubicación del cliente a esta ubicación de AWS mediante esta red de cliente.
Red de clientes	La red por la que pasó el tráfico. Por lo general, es el proveedor de servicios de Internet (ISP) o el número de sistema autónomo (ASN) del tráfico de red.
Ubicación de AWS	La ubicación de AWS del tráfico de red, que puede ser una región de Región de AWS o una ubicación periférica de Internet.
Tipo de impacto	El tipo de impacto del evento de estado. Los eventos de estado suelen deberse

	Descripción
	a aumentos de latencia (problemas de rendimiento) o de accesibilidad (problemas de disponibilidad). También puede hacer clic en el tipo de impacto para ver la causa del deterioro. Cuando es posible, Internet Monitor analiza el origen de un evento de estado para determinar si fue causado por AWS o un ASN (proveedor de servicios de Internet). Tenga en cuenta que este análisis continúa después de que se resuelva el evento. Internet Monitor puede actualizar los eventos con nueva información durante un máximo de una hora.

Si elige una de las ubicaciones de los clientes en la Tabla de eventos de estado, podrá ver más detalles sobre el evento de estado en esa ubicación. Por ejemplo, puede ver cuándo comenzó el evento, cuándo finalizó y el impacto en el tráfico local.

Visualización de rutas de red

El análisis de deterioro completo incluye una ruta de red completa en la sección Visualización de rutas de red. La ruta completa le muestra cada nodo a lo largo de la ruta de red de la aplicación para el evento de estado, entre la ubicación de AWS y el cliente, para un par cliente-ubicación.

Si Internet Monitor determina la causa de una deficiencia, se marca con un círculo rojo discontinuo. Las deficiencias pueden ser causadas por los ASN, normalmente proveedores de servicios de Internet (ISP), o la causa puede ser AWS. Si el deterioro se debe a varias causas, se marcan con un círculo múltiples nodos.

Filtrado y visualización de datos en Amazon CloudWatch Internet Monitor (pestaña Explorador histórico)

Utilice la pestaña Explorador histórico de la consola de CloudWatch, en Internet Monitor, para filtrar y ver los datos de la aplicación que están en registros de CloudWatch. Internet Monitor publica en registros de CloudWatch las mediciones específicas de la aplicación en cuanto a la disponibilidad, el rendimiento, los bytes supervisados transferidos (o el recuento de conexiones de los clientes, solo en los directorios de WorkSpaces) y el tiempo de ida y vuelta de las redes urbanas supervisadas en Regiones de AWS.

> **Note**
>
> Internet Monitor publica las mediciones de Internet en registros de CloudWatch cada cinco minutos (por volumen de tráfico) para las 500 principales redes urbanas (ubicaciones de clientes y ASN, normalmente proveedores de servicios de Internet o ISP) que envían tráfico a cada monitor. Si lo desea, puede optar por publicar las mediciones de Internet de todas las redes urbanas supervisadas (hasta el límite de servicio de 500 000 redes urbanas) en un bucket de Amazon S3. Para obtener más información, consulte Publicar mediciones de Internet en Amazon S3 en Amazon CloudWatch Internet Monitor.

Para empezar a explorar los datos de su aplicación, seleccione un período de tiempo. A continuación, elija una ubicación geográfica específica, como una ciudad, y, opcionalmente, otros filtros. Internet Monitor aplica los filtros a los datos de los registros de mediciones de Internet que ha publicado para las redes de ciudad para el tráfico de sus aplicaciones. A continuación, muestra gráficos de los datos con la puntuación de rendimiento, la puntuación de disponibilidad, los bytes transferidos (para las VPC y los Equilibradores de carga de red) o el recuento de conexiones de los clientes (para los directorios de WorkSpaces) y el tiempo de ida y vuelta (RTT) de la aplicación a lo largo del tiempo.

La tabla All events (Todos los eventos) que aparece debajo de los gráficos muestra los eventos de estado que el filtro devuelve para el tráfico de la aplicación, con información sobre cada evento. Incluye las siguientes columnas.

	Descripción
Inicio del evento	La hora en la que se inició el evento de estado.

	Descripción
Status	Si el evento sigue activo o se ha resuelto.
Ubicación del cliente	La ubicación de los usuarios finales que se vieron afectados por el evento, que experimen taron un aumento de la latencia o un rendimien to reducido. Para obtener más información sobre la precisión de la ubicación del cliente en Internet Monitor, consulte Información y precisión de la geolocalización en Internet Monitor.
Impacto en el tráfico	El impacto ponderado del evento en la ubicación del evento de estado. Es decir, por ejemplo, el impacto en la latencia, en comparación con el rendimiento típico para el tráfico desde la ubicación de un cliente a la de AWS a través del ASN del cliente, normalmen te un proveedor de servicios de Internet (ISP). Del mismo modo, en el caso de un evento que afecte a la disponibilidad, verá cuál es el impacto en la disponibilidad en comparación con la disponibilidad típica de la ubicación del cliente para la ubicación de AWS a través del ASN del cliente.
Duración del evento	Cuánto duró el evento. Internet Monitor cancela los eventos de estado cuando ya no afectan a más del 5 % (en total) de las ubicaciones de los clientes de la aplicación.
ISP del cliente	El ASN, normalmente el proveedor de servicios de Internet (ISP), que era el operador del tráfico de red.

	Descripción
Ubicación del servicio	La ubicación del servicio desde la que se originó el tráfico de red, que puede ser una región de Región de AWS o una ubicación periférica.

Como alternativa, puede ver las mediciones de su aplicación accediendo a los registros directamente de cada punto de datos. En el menú Actions (Acciones), elija View CloudWatch Logs (Ver registros de CloudWatch). Tenga en cuenta que, dado que estas métricas se registran en su cuenta cuando se crean, también puede crear otros paneles o alarmas de CloudWatch en función de las métricas. Para obtener más información, consulte Obtener información para mejorar el rendimiento de las aplicaciones en Amazon CloudWatch Internet Monitor (pestaña Estadísticas del tráfico) y Crear alarmas con Amazon CloudWatch Internet Monitor.

Además de explorar y analizar las métricas de Internet Monitor y, posiblemente, crear paneles y alarmas en función de ellas, puede utilizar Internet Monitor para comprender las formas en que podría mejorar el rendimiento de la aplicación. La pestaña Traffic insights (Estadísticas del tráfico) tiene varias formas de permitirle explorar las opciones. Para obtener más información, consulte las Sugerencias de optimización del tráfico en la pestaña Información sobre el tráfico. Además, puede ver los ejemplos específicos en el capítulo de casos de uso de Internet Monitor.

Obtener información para mejorar el rendimiento de las aplicaciones en Amazon CloudWatch Internet Monitor (pestaña Estadísticas del tráfico)

Utilice la pestaña Información sobre el tráfico de la consola de CloudWatch, en Internet Monitor, para consultar la información resumida del tráfico principal (por volumen) de su aplicación. Puede filtrar y ordenar el tráfico de su aplicación de varias maneras. A continuación, desplácese hacia abajo y seleccione diferentes combinaciones de configuraciones para su aplicación para ver qué sugiere Internet Monitor como alternativas óptimas a fin de obtener el mejor rendimiento de tiempo hasta el primer byte (TTFB).

Internet Monitor publica las mediciones de Internet en registros de CloudWatch cada cinco minutos (por volumen de tráfico) para las 500 principales redes urbanas (ubicaciones de clientes y ASN, normalmente proveedores de servicios de Internet o ISP) que envían tráfico a cada monitor. Si lo desea, puede optar por publicar las mediciones de Internet de todas las redes urbanas supervisadas (hasta el límite de servicio de 500 000 redes urbanas) en un bucket de Amazon S3. Para obtener

más información, consulte Publicar mediciones de Internet en Amazon S3 en Amazon CloudWatch Internet Monitor.

Resúmenes de tráfico principal

Puede empezar por ver resúmenes generales del tráfico y el rendimiento generales de su aplicación, durante un período de tiempo específico, filtrados por ubicación del cliente o proveedor de red. También puede analizar el rendimiento de su aplicación para las ubicaciones de clientes superiores (o inferiores) por volumen de tráfico, filtrarlas y ordenarlas de varias formas. Por ejemplo, puede ordenar por grado de detalle (es decir, ciudad, subdivisión, país o área metropolitana), por el tráfico total, el tiempo medio hasta el primer byte (TTFB) y otros factores.

Para obtener más información sobre la precisión de la ubicación del cliente en Internet Monitor, consulte Información y precisión de la geolocalización en Internet Monitor.

> **Note**
>
> Los filtros que utilice se aplican a toda la página, por lo que afectan a las redes de ciudades que se incluyen en los gráficos resumidos y la información del tráfico total, y también a las redes de ciudades que se incluyen en la sección de sugerencias de optimización del tráfico que aparece a continuación.

Sugerencias de optimización del tráfico

La sección de sugerencias de optimización del tráfico muestra un conjunto filtrado de redes urbanas supervisadas (ubicaciones y ASN, proveedores de servicios de Internet) para su tráfico, junto con el tráfico total de clientes de cada una de ellas. Las entradas de la tabla se basan en los filtros elegidos para el tráfico de la aplicación en Estadísticas del tráfico, que se encuentra en la parte superior de la página. El valor predeterminado son las 10 principales ciudades por volumen de tráfico. Por lo general, aparecen más de 10 filas en la tabla, porque hay una entrada para cada par único de ciudades y redes. Es decir, hay una fila para cada combinación de ubicación (ciudad) y ASN (proveedor de red) a través de la cual los clientes acceden a la aplicación, como Dallas, Texas (EE. UU.) y Comcast, por ejemplo.

> **Note**
>
> Para ver las sugerencias de optimización del tráfico para todas las redes urbanas supervisadas, puede ejecutar una consulta directamente en CloudWatch Insights. Para ver un ejemplo de consulta que no incluye el filtro de granularidad geográfica que limita la lista de redes urbanas de esta página, consulte Uso de Información de registros de Amazon CloudWatch con Amazon CloudWatch Internet Monitor.

En esta sección, seleccione diferentes opciones: Amazon EC2, CloudFront o ambas. Esto le permite ver cuáles son los valores del tiempo medio previsto hasta el primer byte (TTFB) para los clientes que utilizan su aplicación con esos servicios en distintas regiones de AWS, en comparación con el TTFB actual. Para obtener más información sobre los cálculos de TTFB, consulte Cálculos de TTFB y latencia de AWS.

Al seleccionar diferentes opciones y, a continuación, ver los resultados en la tabla, puede empezar a planificar las configuraciones y las implementaciones que pueden mejorar el rendimiento de sus clientes. Tenga en cuenta que es posible que vea un guion (-) en lugar de un valor en la columna cuando los datos no estén disponibles para su visualización. Para revisar un ejemplo específico de cómo mejorar el rendimiento, consulte Uso de Amazon CloudWatch Internet Monitor para una mejor experiencia de juego.

Por ejemplo, para empezar, para una red urbana específica (ubicación del cliente y par de ASN), pruebe con la opción EC2 o CloudFront, o ambas. Para cada red urbana que aparece en la tabla, Internet Monitor muestra las posibles mejoras del rendimiento del TTFB, en función de la elección del enrutamiento del tráfico (a través de una Región de AWS específica) con esa opción, en comparación con la configuración actual. (Tenga en cuenta que, para mayor exhaustividad, la tabla también incluye las rutas que ya están optimizadas). Por ejemplo, es posible que vea un TTFB promedio previsto de 50 ms para usar EC2 y el enrutamiento mediante `us-east-1` en comparación con su configuración actual con un TTFB de 100 ms, en el que utiliza enrutamiento EC2 mediante `us-west-2`. Por lo tanto, podría considerar el enrutamiento mediante `us-west-2`.

Como otro ejemplo, puede seleccionar EC2 y, a continuación, comprobar que no supone una diferencia apreciable en el rendimiento de una ubicación de cliente y un ASN, pero, a continuación, tener en cuenta que, al seleccionar CloudFront con la misma región, reduce un poco el TTFB. Esto sugiere que agregar una distribución de CloudFront delante de la aplicación

podría mejorar el rendimiento y valdría la pena intentarlo para este proveedor de red y ubicación de cliente.

Exploración de datos mediante las herramientas de CloudWatch y la interfaz de consulta de Internet Monitor

Además de visualizar el rendimiento y la disponibilidad de su aplicación mediante el panel de Amazon CloudWatch Internet Monitor, existen varios métodos que puede utilizar para profundizar más en los datos que Internet Monitor genera para usted. Estos métodos incluyen el uso de las herramientas de CloudWatch con los datos de Internet Monitor almacenados en los archivos de registro de CloudWatch y el uso de la interfaz de consulta de Internet Monitor. Algunas de las herramientas que puede utilizar son Información de registros de CloudWatch, las métricas de CloudWatch, Información de colaboradores de CloudWatch y Amazon Athena. Puede utilizar algunas o todas estas herramientas, así como el panel de control, para explorar los datos de Internet Monitor, según sus necesidades.

Internet Monitor agrega las métricas de CloudWatch sobre el tráfico a su aplicación y a cada Región de AWS, e incluye datos como el impacto total en el tráfico, la disponibilidad y el tiempo de ida y vuelta. Estos datos se publican en los registros de CloudWatch y también están disponibles para su uso con la interfaz de consulta de Internet Monitor. Los detalles sobre la granularidad geográfica y otros aspectos de la información disponible para cada uno de ellos varían.

Amazon CloudWatch Internet Monitor publica los datos del monitor en intervalos de 5 minutos y, a continuación, los pone a disposición de varias formas. En la siguiente tabla se enumeran los escenarios para acceder a los datos de Internet Monitor y se describen las características de los datos que se recopilan para cada uno de ellos.

Característica	Registros de CloudWatch	Exportar a S3	Interfaz de consulta	Panel de CloudWatch
Habilitado de forma predeterminada	Sí	No	Sí	Sí
Cantidad de redes urbanas	Las 500 principales (ver la nota a continuación)	Todos	Todos	Todos

Característica	Registros de CloudWatch	Exportar a S3	Interfaz de consulta	Panel de CloudWatch
para las que se recopilan datos				
Retención de datos	Controlados por el usuario	Controlados por el usuario	30 días	30 días
Granularidades geográficas para las que se recopilan los datos	Todos (red urbana, área metropolitana + red, subdivisión + red, país + red)	Red urbana	Todos (red urbana, área metropolitana + red, subdivisión + red, país + red)	Todos (red urbana, área metropolitana + red, subdivisión + red, país + red)
Cómo consultar y filtrar datos	Uso de Información de registros de Amazon CloudWatch con Amazon CloudWatch Internet Monitor	Usar Amazon Athena para consultar las mediciones de Internet en los archivos de registro de Amazon S3	Uso de la interfaz de consulta de Amazon CloudWatch Internet Monitor	Supervisión y optimización con el panel de control de Internet Monitor

Nota: Las 500 mediciones principales se recopilan para redes urbanas. Las 250 principales para área metropolitana + redes, las 100 principales para subdivisión + redes, y las 50 principales para país + redes.

En este capítulo se describe cómo consultar y explorar los datos mediante las herramientas de CloudWatch o la interfaz de consulta de Internet Monitor, junto con ejemplos de cada método.

Contenido

- Usar Amazon Athena para consultar las mediciones de Internet en los archivos de registro de Amazon S3
- Uso de la interfaz de consulta de Amazon CloudWatch Internet Monitor

Uso de Información de registros de Amazon CloudWatch con Amazon CloudWatch Internet Monitor

Amazon CloudWatch Internet Monitor publica mediciones detalladas de la disponibilidad y el tiempo de ida y vuelta en Registros de CloudWatch, y puede utilizar las consultas de Información de registros de Amazon CloudWatch para filtrar un subconjunto de registros para una ciudad o zona geográfica específica (ubicación del cliente), ASN del cliente (ISP) y origen de AWS concretos.

Para obtener más información sobre la precisión de la ubicación del cliente en Internet Monitor, consulte Información y precisión de la geolocalización en Internet Monitor.

Los ejemplos de esta sección pueden ayudarle a crear consultas de Información de registros de CloudWatch para obtener más información sobre las medidas y métricas del tráfico de sus propias aplicaciones. Si utiliza estos ejemplos en Información de registros de CloudWatch, *`sustituya MonitorName`* por el nombre de su monitor.

Ver sugerencias de optimización del tráfico

En la pestaña Estadísticas del tráfico de Internet Monitor, puede ver las sugerencias de optimización del tráfico, filtradas por ubicación. Para ver la misma información que se muestra en la sección de Sugerencias de optimización del tráfico de esa pestaña, pero sin el filtro de granularidad de ubicación, puede utilizar la siguiente consulta de Información de registros de CloudWatch.

1. En la AWS Management Console, vaya a Información de registros de Amazon CloudWatch.
2. En Log Group (Grupo de registro), seleccione `/aws/internet-monitor/`*`monitorName`*`/byCity` y `/aws/internet-monitor/`*`monitorName`*`/byCountry` y, a continuación, especifique un intervalo de tiempo.
3. Agregue la siguiente consulta y, a continuación, ejecútela.

```
fields @timestamp,
clientLocation.city as @city, clientLocation.subdivision as @subdivision,
 clientLocation.country as @country,
`trafficInsights.timeToFirstByte.currentExperience.serviceName` as @serviceNameField,
concat(@serviceNameField, ` (`, `serviceLocation`, `)`) as @currentExperienceField,
```

```
concat(`trafficInsights.timeToFirstByte.ec2.serviceName`, ` (`,
 `trafficInsights.timeToFirstByte.ec2.serviceLocation`, `)`) as @ec2Field,
`trafficInsights.timeToFirstByte.cloudfront.serviceName` as @cloudfrontField,
concat(`clientLocation.networkName`, ` (AS`, `clientLocation.asn`, `)`) as @networkName
| filter ispresent(`trafficInsights.timeToFirstByte.currentExperience.value`)
| stats avg(`trafficInsights.timeToFirstByte.currentExperience.value`) as @averageTTFB,
avg(`trafficInsights.timeToFirstByte.ec2.value`) as @ec2TTFB,
avg(`trafficInsights.timeToFirstByte.cloudfront.value`) as @cloudfrontTTFB,
sum(`bytesIn` + `bytesOut`) as @totalBytes,
latest(@ec2Field) as @ec2,
latest(@currentExperienceField) as @currentExperience,
latest(@cloudfrontField) as @cloudfront,
count(*) by @networkName, @city, @subdivision, @country
| display @city, @subdivision, @country, @networkName, @totalBytes, @currentExperience,
 @averageTTFB, @ec2, @ec2TTFB, @cloudfront, @cloudfrontTTFB
| sort @totalBytes desc
```

Ver la disponibilidad de Internet y el RTT (p50, p90 y p95)

Para ver la disponibilidad de Internet y el tiempo de ida y vuelta (p50, p90 y p95) del tráfico, puede utilizar la siguiente consulta en Información de registros de CloudWatch.

Zona geográfica del usuario final: Chicago, IL, Estados Unidos

Red de usuario final (ASN): AS7018

Ubicación del servicio de AWS: región de Este de EE. UU. (Norte de Virginia)

Para consultar los registros, haga lo siguiente:

1. En la AWS Management Console, vaya a Información de registros de Amazon CloudWatch.
2. En Log Group (Grupo de registro), seleccione `/aws/internet-monitor/`*`monitorName`*`/byCity` y `/aws/internet-monitor/`*`monitorName`*`/byCountry` y, a continuación, especifique un intervalo de tiempo.
3. Agregue la siguiente consulta y, a continuación, ejecútela.

La consulta devuelve todos los datos de rendimiento de los usuarios que se conecten desde AS7018 en Chicago, Illinois, hacia la región de Este de EE. UU. (Norte de Virginia) durante el período de tiempo seleccionado.

```
fields @timestamp,
```

```
internetHealth.availability.experienceScore as availabilityExperienceScore,
internetHealth.availability.percentageOfTotalTrafficImpacted as
 percentageOfTotalTrafficImpacted,
internetHealth.performance.experienceScore as performanceExperienceScore,
internetHealth.performance.roundTripTime.p50 as roundTripTimep50,
internetHealth.performance.roundTripTime.p90 as roundTripTimep90,
internetHealth.performance.roundTripTime.p95 as roundTripTimep95
 | filter clientLocation.country == `United States`
 and clientLocation.city == `Chicago`
 and serviceLocation == `us-east-1`
 and clientLocation.asn == 7018
```

Para obtener más información, consulte Análisis de los datos de registros con Información de registros de Amazon CloudWatch.

Utilizar Contributor Insights con Amazon CloudWatch Internet Monitor

Información de colaboradores de Amazon CloudWatch puede ayudarle a identificar las principales ubicaciones de clientes y redes (ASN o proveedores de servicios de Internet) de su aplicación. Utilice los siguientes ejemplos de reglas de Información de colaboradores para comenzar con las reglas que sean útiles con Amazon CloudWatch Internet Monitor. Para obtener más información, consulte Creación de una regla de Información de colaboradores.

Para obtener más información sobre la precisión de la ubicación del cliente en Internet Monitor, consulte Información y precisión de la geolocalización en Internet Monitor.

> **Note**
>
> Internet Monitor publica datos cada cinco minutos, por lo que, después de configurar una regla de Información de colaboradores, debe ajustar el período a cinco minutos para ver un gráfico.

Ver las principales ubicaciones y ASN afectados por un impacto de disponibilidad

Para ver las principales ubicaciones de los clientes y los ASN afectados por una caída de la disponibilidad, puede utilizar la siguiente regla de Información de colaboradores en el editor de Syntax. Reemplace *monitor-name* por el nombre de su monitor.

```
{
    "Schema": {
```

```
        "Name": "CloudWatchLogRule",
        "Version": 1
    },
    "AggregateOn": "Sum",
    "Contribution": {
        "Filters": [
            {
                "Match": "$.clientLocation.city",
                "IsPresent": true
            }
        ],
        "Keys": [
            "$.clientLocation.city",
            "$.clientLocation.networkName"
        ],
        "ValueOf": "$.awsInternetHealth.availability.percentageOfTotalTrafficImpacted"
    },
    "LogFormat": "JSON",
    "LogGroupNames": [
        "/aws/internet-monitor/monitor-name/byCity"
    ]
}
```

Ver las principales ubicaciones de clientes y ASN afectados por un impacto de latencia

Para ver las principales ubicaciones de los clientes y los ASN afectados por un aumento del tiempo de ida y vuelta (latencia), puede utilizar la siguiente regla de Información de colaboradores en el editor de Syntax. Reemplace *monitor-name* por el nombre de su monitor.

```
{
    "Schema": {
        "Name": "CloudWatchLogRule",
        "Version": 1
    },
    "AggregateOn": "Sum",
    "Contribution": {
        "Filters": [            {
                "Match": "$.clientLocation.city",
                "IsPresent": true
            }
        ],
        "Keys": [
            "$.clientLocation.city",
```

```
            "$.clientLocation.networkName"
        ],
        "ValueOf": "$.awsInternetHealth.performance.percentageOfTotalTrafficImpacted"
    },
    "LogFormat": "JSON",
    "LogGroupNames": [
        "/aws/internet-monitor/monitor-name/byCity"
    ]
}
```

Ver las principales ubicaciones de clientes y los ASN afectados por el porcentaje total de tráfico

Para ver las principales ubicaciones de clientes y los ASN afectados por el porcentaje total de tráfico, puede usar la siguiente regla de Información de colaboradores en el editor de Syntax. Reemplace *monitor-name* por el nombre de su monitor.

```
{
    "Schema": {
        "Name": "CloudWatchLogRule",
        "Version": 1
    },
    "AggregateOn": "Sum",
    "Contribution": {
        "Filters": [
            {
                "Match": "$.clientLocation.city",
                "IsPresent": true
            }
        ],
        "Keys": [
            "$.clientLocation.city",
            "$.clientLocation.networkName"
        ],
        "ValueOf": "$.percentageOfTotalTraffic"
    },
    "LogFormat": "JSON",
    "LogGroupNames": [
        "/aws/internet-monitor/monitor-name/byCity"
    ]
}
```

Uso de las métricas de CloudWatch con Amazon CloudWatch Internet Monitor

Amazon CloudWatch Internet Monitor publica las métricas de la cuenta, incluidas las métricas del rendimiento, la disponibilidad, el tiempo de ida y vuelta y el rendimiento (bytes por segundo), que puede ver en las métricas de CloudWatch de la consola de CloudWatch. Para encontrar todas las métricas del monitor, en el panel de métricas de CloudWatch, consulte el espacio de nombres personalizado `AWS/InternetMonitor`.

Estas métricas se suman a todo el tráfico de Internet a las VPC, los Equilibradores de carga de red, las distribuciones de CloudFront o los directorios de WorkSpaces del monitor, y a todo el tráfico de cada Región de AWS y a la ubicación periférica de Internet que se supervise. Las regiones se definen por la ubicación del servicio, que puede ser todas las ubicaciones o una región específica, por ejemplo `us-east-1`.

Nota: las redes urbanas son ubicaciones de clientes y ASN (normalmente proveedores de servicios de Internet o ISP).

Internet Monitor proporciona las siguientes métricas.

Métrica	Descripción
Puntuación de rendimiento	Una puntuación de rendimiento represent a el porcentaje estimado de tráfico que no experimenta una caída en el rendimiento.
Puntuación de disponibilidad	Una puntuación de disponibilidad representa el porcentaje estimado de tráfico que no sufre una caída en la disponibilidad.
BytesIn	Bytes de entrada transferidos para el tráfico de Internet de su aplicación en todas las redes urbanas de aplicaciones.
BytesOut	Bytes de salida transferidos para el tráfico de Internet de su aplicación en todas las redes urbanas de aplicaciones.

Métrica	Descripción
BytesInMonitored	Bytes de entrada transferidos para el tráfico de Internet de su aplicación en todas las redes urbanas supervisadas.
BytesOutMonitored	Bytes de salida transferidos para el tráfico de Internet de su aplicación en las redes urbanas supervisadas.
Tiempo de ida y vuelta (RTT)	Tiempo de ida y vuelta entre las Regiones de AWS, ASN (normalmente proveedores de servicios de Internet o ISP) y ubicaciones (como ciudades) específicas de sus directori os de VPC, Equilibradores de carga de red, distribuciones de CloudFront o WorkSpaces.
Redes urbanas supervisadas	El número de redes urbanas que Internet Monitor supervisa para detectar el tráfico de Internet de su aplicación. Nunca supera el límite máximo establecido como máximo de redes urbanas para el monitor.
Porcentaje de tráfico supervisado	El porcentaje del tráfico total de Internet de las aplicaciones de este monitor representado (incluido) por las redes urbanas que Internet Monitor supervisa. Es inferior a 100 (es decir, inferior al 100 %) si los clientes acceden a la aplicación desde más redes urbanas que el límite máximo de redes urbanas establecido para el monitor.
Redes urbanas para un tráfico del 100 por ciento	El número en el que debe establecer el límite máximo de las redes de su ciudad si desea supervisar el 100 % del tráfico de Internet de sus aplicaciones en Internet Monitor.

Métrica	Descripción
Redes urbanas para un tráfico del 99 por ciento	El número en el que debe establecer el límite máximo de las redes de su ciudad si desea supervisar el 99 % del tráfico de Internet de sus aplicaciones en Internet Monitor.
Redes urbanas para un tráfico del 95 por ciento	El número en el que debe establecer el límite máximo de las redes de su ciudad si desea supervisar el 95 % del tráfico de Internet de sus aplicaciones en Internet Monitor.
Redes urbanas para un tráfico del 90 por ciento	El número en el que debe establecer el límite máximo de las redes de su ciudad si desea supervisar el 90 % del tráfico de Internet de sus aplicaciones en Internet Monitor.
Redes urbanas para un tráfico del 75 por ciento	El número en el que debe establecer el límite máximo de las redes de su ciudad si desea supervisar el 75 % del tráfico de Internet de sus aplicaciones en Internet Monitor.
Redes urbanas para un tráfico del 50 por ciento	El número en el que debe establecer el límite máximo de las redes de su ciudad si desea supervisar el 50 % del tráfico de Internet de sus aplicaciones en Internet Monitor.
Redes urbanas para un tráfico del 25 por ciento	El número en el que debe establecer el límite máximo de las redes de su ciudad si desea supervisar el 25 % del tráfico de Internet de sus aplicaciones en Internet Monitor.

> **Note**
>
> Para ver ejemplos del uso de varias de estas métricas para ayudar a determinar los valores que se deben elegir para el máximo de redes urbanas para su monitor, consulte Elegir un valor máximo de red urbana.

Para obtener más información, consulte Uso de métricas de Amazon CloudWatch.

Usar Amazon Athena para consultar las mediciones de Internet en los archivos de registro de Amazon S3

Puede utilizar Amazon Athena para consultar y ver las mediciones de Internet que Amazon CloudWatch Internet Monitor publica en un bucket de Amazon S3. Internet Monitor ofrece la opción de publicar las mediciones de Internet de su aplicación en un bucket de S3 para el tráfico dirigido a Internet de las redes urbanas supervisadas (ubicaciones de clientes y ASN, normalmente proveedores de servicios de Internet o ISP). Independientemente de si elige publicar las mediciones en S3, Internet Monitor publica automáticamente en Registros de CloudWatch cada cinco minutos las mediciones de Internet de las 500 principales ciudades urbanas (por volumen de tráfico) de cada monitor.

Este capítulo incluye los pasos para crear una tabla en Athena para las mediciones de Internet ubicadas en un archivo de registro de S3 y, a continuación, proporciona consultas de ejemplo para ver diferentes vistas de las mediciones. Por ejemplo, puede consultar las 10 redes urbanas más afectadas según el impacto en la latencia.

Usar Amazon Athena para crear una tabla de mediciones de Internet en Internet Monitor

Para empezar a utilizar Athena con los archivos de registro de Internet Monitor S3, primero debe crear una tabla para las mediciones de Internet.

Siga los pasos de este procedimiento para crear una tabla en Athena basada en los archivos de registro de S3. A continuación, puede ejecutar consultas de Athena en la tabla, como estas consultas de medición de Internet de ejemplo, para obtener información sobre sus medidas.

Crear una tabla de Athena

1. Abra la consola Athena en https://console.aws.amazon.com/athena/.
2. En el editor de consultas de Athena, introduzca una declaración de consulta para generar una tabla con las mediciones de Internet de Internet Monitor. Sustituya el valor del parámetro LOCATION por la ubicación del bucket S3 en el que se almacenan las mediciones de Internet de Internet Monitor.

```
CREATE EXTERNAL TABLE internet_measurements (
    version INT,
    timestamp INT,
    clientlocation STRING,
```

```
    servicelocation STRING,
    percentageoftotaltraffic DOUBLE,
    bytesin INT,
    bytesout INT,
    clientconnectioncount INT,
    internethealth STRING,
    trafficinsights STRING
)
PARTITIONED BY (year STRING, month STRING, day STRING)
ROW FORMAT SERDE 'org.openx.data.jsonserde.JsonSerDe'
LOCATION
's3://bucket_name/bucket_prefix/AWSLogs/account_id/internetmonitor/AWS_Region/'
TBLPROPERTIES ('skip.header.line.count' = '1');
```

3. Introduzca una sentencia para crear una partición que lea los datos. La consulta de ejemplo siguiente crea una sola partición para la fecha y ubicación especificadas:

```
ALTER TABLE internet_measurements
ADD PARTITION (year = 'YYYY', month = 'MM', day = 'dd')
LOCATION
's3://bucket_name/bucket_prefix/AWSLogs/account_id/internetmonitor/AWS_Region/YYYY/
MM/DD';
```

4. Elija Ejecutar.

Ejemplos de declaraciones de Athena para mediciones de Internet

El siguiente es un ejemplo de una sentencia para generar una tabla:

```
CREATE EXTERNAL TABLE internet_measurements (
    version INT,
    timestamp INT,
    clientlocation STRING,
    servicelocation STRING,
    percentageoftotaltraffic DOUBLE,
    bytesin INT,
    bytesout INT,
    clientconnectioncount INT,
    internethealth STRING,
    trafficinsights STRING
)
PARTITIONED BY (year STRING, month STRING, day STRING)
ROW FORMAT SERDE 'org.openx.data.jsonserde.JsonSerDe'
```

```
LOCATION 's3://internet-measurements/TestMonitor/AWSLogs/1111222233332/internetmonitor/
us-east-2/'
TBLPROPERTIES ('skip.header.line.count' = '1');
```

El siguiente es un ejemplo de una sentencia para crear una partición para leer los datos:

```
ALTER TABLE internet_measurements
ADD PARTITION (year = '2023', month = '04', day = '07')
LOCATION 's3://internet-measurements/TestMonitor/AWSLogs/1111222233332/internetmonitor/
us-east-2/2023/04/07/'
```

Ejemplos de consultas de Amazon Athena para usarlas con mediciones de Internet en Internet Monitor

En esta sección se incluyen ejemplos de consultas que puede utilizar con Amazon Athena para obtener información sobre las mediciones de Internet de su aplicación publicadas en Amazon S3.

Consultar las 10 ubicaciones de clientes y ASN más afectados (por porcentaje total de tráfico)

Ejecute esta consulta de Athena para obtener las 10 redes urbanas más afectadas (por porcentaje total del tráfico), es decir, las ubicaciones de los clientes y los ASN, normalmente proveedores de servicios de Internet.

```
SELECT json_extract_scalar(clientLocation, '$.city') as city,
    json_extract_scalar(clientLocation, '$.networkname') as networkName,
    sum(percentageoftotaltraffic) as percentageoftotaltraffic
FROM internet_measurements
GROUP BY json_extract_scalar(clientLocation, '$.city'),
    json_extract_scalar(clientLocation, '$.networkname')
ORDER BY percentageoftotaltraffic desc
limit 10
```

Consultar las 10 ubicaciones de clientes y ASN más afectados (por disponibilidad)

Ejecute esta consulta de Athena para obtener las 10 redes urbanas más afectadas (por porcentaje total del tráfico), es decir, las ubicaciones de los clientes y los ASN, normalmente proveedores de servicios de Internet.

```
SELECT json_extract_scalar(clientLocation, '$.city') as city,
    json_extract_scalar(clientLocation, '$.networkname') as networkName,
    sum(
        cast(
```

```
            json_extract_scalar(
                internetHealth,
                '$.availability.percentageoftotaltrafficimpacted'
            )
        as double )
    ) as percentageOfTotalTrafficImpacted
FROM internet_measurements
GROUP BY json_extract_scalar(clientLocation, '$.city'),
    json_extract_scalar(clientLocation, '$.networkname')
ORDER BY percentageOfTotalTrafficImpacted desc
limit 10
```

Consultar las 10 ubicaciones de clientes y ASN más afectados (por latencia)

Ejecute esta consulta de Athena para obtener las 10 redes urbanas más afectadas (por impacto de la latencia), es decir, las ubicaciones de los clientes y los ASN, normalmente proveedores de servicios de Internet.

```
SELECT json_extract_scalar(clientLocation, '$.city') as city,
    json_extract_scalar(clientLocation, '$.networkname') as networkName,
    sum(
        cast(
            json_extract_scalar(
                internetHealth,
                '$.performance.percentageoftotaltrafficimpacted'
            )
        as double )
    ) as percentageOfTotalTrafficImpacted
FROM internet_measurements
GROUP BY json_extract_scalar(clientLocation, '$.city'),
    json_extract_scalar(clientLocation, '$.networkname')
ORDER BY percentageOfTotalTrafficImpacted desc
limit 10
```

Consultar los puntos destacados del tráfico para las ubicaciones de sus clientes y los ASN

Ejecute esta consulta de Athena para devolver los aspectos más destacados del tráfico, como la puntuación de disponibilidad, la puntuación de rendimiento y el tiempo transcurrido hasta el primer byte para las redes de sus ciudades, es decir, las ubicaciones de los clientes y los ASN, normalmente proveedores de servicios de Internet.

```
SELECT json_extract_scalar(clientLocation, '$.city') as city,
    json_extract_scalar(clientLocation, '$.subdivision') as subdivision,
    json_extract_scalar(clientLocation, '$.country') as country,
    avg(cast(json_extract_scalar(internetHealth, '$.availability.experiencescore') as
 double)) as availabilityScore,
    avg(cast(json_extract_scalar(internetHealth, '$.performance.experiencescore') as
 double)) performanceScore,
    avg(cast(json_extract_scalar(trafficinsights,
 '$.timetofirstbyte.currentexperience.value') as double)) as averageTTFB,
    sum(bytesIn) as bytesIn,
    sum(bytesOut) as bytesOut,
    sum(bytesIn + bytesOut) as totalBytes
FROM internet_measurements
where json_extract_scalar(clientLocation, '$.city') != 'N/A'
GROUP BY
json_extract_scalar(clientLocation, '$.city'),
    json_extract_scalar(clientLocation, '$.subdivision'),
    json_extract_scalar(clientLocation, '$.country')
ORDER BY totalBytes desc
limit 100
```

Para obtener más información sobre el uso de Athena, consulte la Guía del usuario de Amazon Athena.

Uso de la interfaz de consulta de Amazon CloudWatch Internet Monitor

Una opción para obtener más información sobre el tráfico de Internet de su aplicación de AWS es utilizar la interfaz de consulta de Amazon CloudWatch Internet Monitor. Para utilizar la interfaz de consulta, debe crear una consulta con los filtros de datos que elija y, a continuación, ejecutar la consulta para obtener un subconjunto de los datos de Internet Monitor. Explorar los datos que devuelve la consulta puede proporcionar información sobre el rendimiento de su aplicación en Internet.

Puede consultar y explorar todas las métricas que Internet Monitor captura mediante el monitor, incluidas las puntuaciones de disponibilidad y rendimiento, los bytes transferidos, los tiempos de ida y vuelta y el tiempo hasta el primer byte (TTFB).

Internet Monitor utiliza la interfaz de consulta para proporcionar los datos que puede explorar en el panel de la consola de Internet Monitor. Cuando consulta las opciones de búsqueda del panel, en la pestaña Explorador histórico o en la pestaña Información sobre el tráfico, puede consultar y filtrar los datos de Internet de su aplicación.

Si desea tener una flexibilidad para explorar y filtrar sus datos mayor que la que ofrece el panel, puede usar la interfaz de consulta mediante las operaciones de la API de Internet Monitor mediante la AWS Command Line Interface o un AWS SDK. En esta sección, se presentan los tipos de consultas que puede usar con la interfaz de consultas y los filtros que puede especificar para crear un subconjunto de datos, a fin de obtener información sobre el tráfico de Internet de su aplicación.

Temas

Cómo utilizar la interfaz de consulta

Para crear una consulta con la interfaz de consulta, se elige un tipo de consulta y, a continuación, se especifican los valores de filtro para devolver el subconjunto específico deseado de los datos del archivo de registro. A continuación, puede trabajar con el subconjunto de datos para filtrar más detalladamente, ordenar y crear informes, etc.

El proceso de consulta funciona de la siguiente manera:

1. Cuando ejecuta una consulta, Internet Monitor devuelve un `query ID`, que es exclusivo de la consulta. En esta sección se describen los tipos de consultas disponibles y las opciones para filtrar los datos de las consultas. Para entender cómo funciona esto, también puede revisar la sección en ejemplos de consultas.
2. Debe especificar el ID de consulta con el nombre de su monitor mediante la operación GetQueryResults de la API, para devolver los resultados de los datos de la consulta. Cada tipo de consulta devuelve un conjunto diferente de campos de datos. Para obtener más información, consulte Obtener resultados de las consultas.

La interfaz de consultas proporciona los tres tipos de consultas a continuación. Cada tipo de consulta devuelve un conjunto diferente de información sobre el tráfico de los archivos de registro, tal y como se muestra.

- Mediciones: proporciona la puntuación de disponibilidad, la puntuación de rendimiento, el tráfico total y los tiempos de ida y vuelta, en intervalos de 5 minutos.

- Ubicaciones principales: proporciona la puntuación de disponibilidad, la puntuación de rendimiento, el tráfico total y la información del tiempo hasta el primer byte (TTFB), para las combinaciones de ubicación principal y ASN que esté supervisando, por volumen de tráfico.
- Detalles de las principales ubicaciones: proporciona el TTFB para Amazon CloudFront, su configuración actual y la configuración de Amazon EC2 con mejor rendimiento, en intervalos de 1 hora.

Con cada uno de estos tipos de consultas, puede filtrar los datos aún más, especificando uno o más de los siguientes criterios:

- Ubicación de AWS: para la ubicación de AWS, puede especificar CloudFront o una Región de AWS, como `us-east-2`, `us-west-2`, etc.
- ASN: especifique un ASN, que suele ser un proveedor de servicios de Internet (ISP).
- Ubicación del cliente: para la ubicación, especifique una ciudad, un área metropolitana, una subdivisión o un país.
- Ubicación geográfica: especifique la ubicación `geo` para algunas consultas. Esto es obligatorio para las consultas que utilizan el tipo de consulta de `Top locations`, pero no está permitido para otros tipos de consulta. Para saber cuándo especificar los parámetros de filtro `geo`, consulte la sección ejemplos de consultas.

Los operadores que puede utilizar para filtrar los datos son `EQUALS` y `NOT_EQUALS`. Para obtener más información sobre los parámetros de filtrado, consulte la operación de la API FilterParameter.

Para obtener más información sobre las operaciones de la interfaz de consulta, consulte las siguientes operaciones de la API en la Guía de referencia de la API de Amazon CloudWatch Internet Monitor.

- Para crear y ejecutar una consulta, consulte la operación de la API StartQuery.
- Para detener una consulta, consulte la operación de la API StopQuery.
- Para devolver los datos de una consulta que haya creado, consulte la operación de la API GetQueryResults.
- Para recuperar el estado de una consulta, consulte la operación de la API GetQueryStatus.

Consultas de ejemplo

Para crear una consulta que pueda utilizar para recuperar un conjunto de datos filtrados del archivo de registro del monitor, utilice la operación de la API StartQuery. Debe especificar un tipo de consulta y filtrar los parámetros para esta. Cuando utilice la operación de la API de la interfaz de consultas de Internet Monitor para obtener los resultados de la consulta mediante el uso de la consulta, recuperará el subconjunto de datos con el que desee trabajar.

Veamos algunos ejemplos para ilustrar cómo funcionan los tipos de consulta y los parámetros de filtro.

Ejemplo 1

Supongamos que desea recuperar todos los datos del archivo de registro del monitor de un país específico, a excepción de una ciudad. El siguiente ejemplo muestra los parámetros de filtro de una consulta que podría crear mediante la operación `StartQuery` para este escenario.

```
{
   MonitorName: "TestMonitor"
   StartTime: "2023-07-12T20:00:00Z"
   EndTime: "2023-07-12T21:00:00Z"
   QueryType: "MEASUREMENTS"
   FilterParameters: [
      {
       Field: "country",
       Operator: "EQUALS",
       Values: ["Germany"]
      },
      {
       Field: "city",
       Operator: "NOT_EQUALS",
       Values: ["Berlin"]
      },
    ]
}
```

Ejemplo 2

Como otro ejemplo, supongamos que desea ver las ubicaciones principales por área metropolitana. Puede usar la siguiente consulta de ejemplo para este escenario.

```
{
```

```
   MonitorName: "TestMonitor"
   StartTime: "2023-07-12T20:00:00Z"
   EndTime: "2023-07-12T21:00:00Z"
   QueryType: "TOP_LOCATIONS"
   FilterParameters: [
      {
       Field: "geo",
       Operator: "EQUALS",
       Values: ["metro"]
      },
    ]
}
```

Ejemplo 3

Supongamos que quiere ver las principales combinaciones de redes urbanas en el área metropolitana de Los Ángeles. Para ello, especifique `geo=city` y, a continuación, establezca `metro` en Los Ángeles. Ahora, la consulta devuelve las principales redes urbanas del área metropolitana de Los Ángeles en lugar de las principales áreas metropolitanas + redes en general.

Este es el ejemplo de consulta que puede usar:

```
{
   MonitorName: "TestMonitor"
   StartTime: "2023-07-12T20:00:00Z"
   EndTime: "2023-07-12T21:00:00Z"
   QueryType: "TOP_LOCATIONS"
   FilterParameters: [
      {
       Field: "geo",
       Operator: "EQUALS",
       Values: ["city"]
      },
      {
       Field: "metro",
       Operator: "EQUALS",
       Values: ["Los Angeles"]
      }
    ]
}
```

Ejemplo 4

Por último, supongamos que desea recuperar datos del TTFB para una subdivisión específica (por ejemplo, un estado de EE. UU.).

El siguiente es un ejemplo de consulta para este escenario:

```
{
   MonitorName: "TestMonitor"
   StartTime: "2023-07-12T20:00:00Z"
   EndTime: "2023-07-12T21:00:00Z"
   QueryType: "TOP_LOCATION_DETAILS"
   FilterParameters: [
      {
       Field: "subdivision",
       Operator: "EQUALS",
       Values: ["California"]
      },
    ]
}
```

Obtención de los resultados de la consulta

Tras definir una consulta, puede devolver un conjunto de resultados mediante la consulta, ejecutando otra operación de la API de Internet Monitor, GetQueryResults. Cuando ejecuta `GetQueryResults`, especifica el ID de consulta de la consulta que definió, junto con el nombre del monitor. `GetQueryResults` recupera los datos de la consulta especificada en un conjunto de resultados.

Cuando ejecute una consulta, asegúrese de que esta haya terminado de ejecutarse antes de usar `GetQueryResults` para ver los resultados. Puede determinar si la consulta se completó mediante la operación de la API GetQueryStatus. Cuando el `Status` de la consulta sea `SUCCEEDED`, puede continuar con la revisión de los resultados.

Cuando finalice la consulta, puede usar la siguiente información como ayuda para la revisión de los resultados. Cada tipo de consulta que utilice para crear una consulta incluye un conjunto único de campos de datos de los archivos de registro, tal y como se describe en la siguiente lista:

Mediciones

El tipo de consulta `measurements` devuelve los siguientes datos:

`timestamp, availability, performance, bytes_in, bytes_out, rtt_p50, rtt_p90, rtt_p95`

Ubicaciones principales

El tipo de consulta `top locations` agrupa los datos por ubicación y proporciona los datos promediados durante un periodo. Los datos que devuelve incluyen lo siguiente:

`aws_location, city, metro, subdivision, country, asn, availability, performance, bytes_in, bytes_out, current_fbl, best_ec2, best_ec2_region, best_cf_fbl`

Tenga en cuenta que la `city`, la `metro` y la `subdivision` solo se devuelven si elige ese tipo de ubicación para el campo `geo`. Se devuelven los siguientes campos de ubicación, según el tipo de ubicación que especifique para `geo`:

```
city = city, metro, subdivision, country
metro = metro, subdivision, country
subdivision = subdivision, country
country = country
```

Detalles de las ubicaciones principales

El tipo de consulta `top locations details` devuelve datos agrupados hora por hora. La consulta devuelve los siguientes datos:

`timestamp, current_service, current_fbl, best_ec2_fbl, best_ec2_region, best_cf_fbl`

Cuando ejecuta la operación de la API `GetQueryResults`, Internet Monitor devuelve lo siguiente en la respuesta:

- Una matriz de cadenas de datos que contiene los resultados que devuelve la consulta. La información se devuelve en matrices alineadas con el campo `Fields`, y también las devuelve mediante la llamada a la API. Con el campo `Fields`, puede analizar la información del repositorio `Data` y, a continuación, filtrarla más detalladamente u ordenarla según sus necesidades.
- Una matriz de campos que muestra los campos para los que la consulta devolvió datos (en la respuesta del campo `Data`). Cada elemento de la matriz es un par de tipos de datos-nombre, como `availability_score-float`.

Resolución de problemas

Si se devuelven errores cuando utiliza las operaciones de la API de la interfaz de consulta, compruebe que dispone de los permisos necesarios para utilizar Amazon CloudWatch Internet Monitor. Asegúrese específicamente de que tiene habilitados los siguientes permisos:

```
internetmonitor:StartQuery
internetmonitor:GetQueryStatus
internetmonitor:GetQueryResults
internetmonitor:StopQuery
```

Estos permisos se incluyen en la política de AWS Identity and Access Management recomendada para usar el panel de control de Internet Monitor de la consola. Para obtener más información, consulte Permisos de IAM para Amazon CloudWatch Internet Monitor.

Crear alarmas con Amazon CloudWatch Internet Monitor

Puede crear alarmas de Amazon CloudWatch en función de las métricas de Amazon CloudWatch Internet Monitor, del mismo modo que puede hacerlo con otras métricas de Amazon CloudWatch.

Por ejemplo, puede crear una alarma basada en la métrica `PerformanceScore` de Internet Monitor y configurarla para que envíe una notificación cuando la métrica sea inferior a un valor que usted elija. Las alarmas para las métricas de Internet Monitor se configuran siguiendo las mismas pautas que para otras métricas de CloudWatch.

A continuación se muestran algunos ejemplos de métricas de Internet Monitor para las que puede crear una alarma:

- Puntuación de rendimiento
- Puntuación de disponibilidad
- RoundtripTime

Para ver todas las métricas disponibles para Internet Monitor, consulte Uso de las métricas de CloudWatch con Amazon CloudWatch Internet Monitor.

El siguiente procedimiento proporciona un ejemplo de cómo configurar una alarma en Puntuación de rendimiento navegando hasta la métrica en el panel de CloudWatch. A continuación, siga los pasos estándar de CloudWatch para crear una alarma basada en el umbral que elija y configurar una notificación o elegir otras opciones.

Crear una alarma para Puntuación de rendimiento en CloudWatch Metrics

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. Seleccione Métricas y, a continuación, Todas las métricas.
3. Para filtrar Internet Monitor, elija `AWS/InternetMonitor`.
4. Elija MeasurementSource, MonitorName..
5. En la lista, seleccione Puntuación de rendimiento.
6. En la pestaña GraphedMetrics, en Acciones, seleccione el icono de la campana para crear una alarma basada en un umbral estático.

Ahora, siga los pasos estándar de CloudWatch para elegir las opciones de la alarma. Por ejemplo, puede elegir que se le notifique mediante un mensaje de Amazon SNS si Puntuación de rendimiento está por debajo de un umbral específico. Como alternativa o de forma adicional, puede añadir la alarma a un panel de control.

Tenga en cuenta lo siguiente:

- Las métricas de Internet Monitor normalmente se calculan y publican en 20 minutos.
- Cuando cree una alarma basada en las métricas de Internet Monitor y configure el periodo retrospectivo de la alarma, asegúrese de tener en cuenta el breve retraso antes de la publicación. Recomendamos que configure los Periodos de evaluación con un periodo retrospectivo de 25 minutos como mínimo.

Para obtener más información sobre el uso de las alarmas de CloudWatch con Internet Monitor, consulte la siguiente entrada del blog: Uso de Amazon CloudWatch Internet Monitor para mejorar la observabilidad de Internet.

Para obtener más información sobre las opciones al crear una alarma de CloudWatch, consulte Cree una alarma de CloudWatch basada en un umbral estático.

Uso de Amazon CloudWatch Internet Monitor con Amazon EventBridge

Los eventos de estado que Amazon CloudWatch Internet Monitor crea para los problemas de red se publican en Amazon EventBridge, por lo que puede enviar notificaciones sobre cualquier deterioro en la experiencia de los usuarios finales con la aplicación.

Para usar EventBridge para trabajar con eventos de estado de Internet Monitor, siga las instrucciones que se indican aquí.

Configurar una regla para Internet Monitor en EventBridge

1. En la AWS Management Console, en EventBridge, elija Rules (Reglas) y, a continuación, ingrese un nombre y una descripción. Cree la regla en el bus de eventos predeterminado.
2. En el paso 2, seleccione Otro como origen del evento y, a continuación, procure que Patrón de eventos se corresponda con el siguiente origen.

```
{
  "source": ["aws.internetmonitor"]
}
```

3. En el paso 3, en el destino, seleccione Servicio de AWS y Grupo de Registro de CloudWatch y, a continuación, seleccione un grupo de registro existente o cree uno nuevo.
4. Agregue las etiquetas que desee y, a continuación, cree la regla. Esta acción debería rellenar el grupo de registros de CloudWatch seleccionado con eventos de EventBridge.

Para obtener más información acerca de cómo funcionan las reglas de EventBridge con los patrones de eventos, vea Eventos y patrones de eventos en EventBridge en la Guía del usuario de Amazon EventBridge.

Solución de problemas de errores de acceso a registros y métricas de CloudWatch

Para admitir algunas características, Amazon CloudWatch Internet Monitor debe interactuar con determinados recursos de Amazon CloudWatch, incluidos los registros y las métricas. Si Internet Monitor no puede acceder a los recursos de CloudWatch a los que necesita tener acceso, Internet Monitor establece un código de estado de `FAULT_ACCESS_CLOUDWATCH` para el monitor.

Existen varios motivos por los que su monitor puede tener el estado `FAULT_ACCESS_CLOUDWATCH`. En las siguientes secciones, se enumeran las posibles causas de estos errores y se sugieren los pasos para la solución de problemas.

Internet Monitor no pudo acceder a los Registros de CloudWatch de su cuenta

Internet Monitor publica registros de diagnóstico sobre el tráfico de aplicaciones que rastrea el monitor. Los publica en grupos de registros en Registros de CloudWatch en la siguiente ubicación: `/`

`aws/internet-monitor/monitor_name/[byCity|byMetro|bySubdivision|byCountry]`. Internet Monitor no pudo acceder a estos grupos de registros.

Estados de error y posibles soluciones:

- Error de limitación de PutLogEvents: es posible que el servicio Internet Monitor haya sufrido una limitación al intentar publicar los registros del monitor en CloudWatch. Revise los límites de limitación de su cuenta y, si es necesario, solicite un aumento del límite.
- No se encontró un grupo de registro: deshabilite y, a continuación, vuelva a habilitar el monitor. Al activar un monitor, se reinicia la creación de grupos de registros, lo que podría corregir el problema.
- Error de acceso denegado a PutLogEvents: póngase en contacto con AWS Support para obtener ayuda.
- PutLogEvents desconocido o error general: póngase en contacto con AWS Support para obtener ayuda.

Internet Monitor no pudo acceder a las métricas de CloudWatch de su cuenta

Internet Monitor proporciona métricas específicas de CloudWatch sobre el tráfico de aplicaciones que rastrea un monitor. Se produjo un error cuando Internet Monitor intentó proporcionar estas métricas a CloudWatch.

Estados de error y posibles soluciones:

- Error de limitación de PutMetricData: es posible que el servicio Internet Monitor haya sufrido una limitación al intentar publicar las métricas del monitor en CloudWatch. Revise los límites de limitación de su cuenta y, si es necesario, solicite un aumento del límite.
- Error de acceso denegado a PutMetricData: póngase en contacto con AWS Support para obtener ayuda.
- PutMetricData desconocido o error general: póngase en contacto con AWS Support para obtener ayuda.

Protección y privacidad de datos con Amazon CloudWatch Internet Monitor

El modelo de responsabilidad compartida de AWS se aplica a la protección y la privacidad de datos en Amazon CloudWatch Internet Monitor. Tal como se describe en este modelo, AWS es

responsable de proteger la infraestructura global que ejecuta toda la nube de AWS. Usted es responsable de mantener el control sobre el contenido alojado en esta infraestructura. Para obtener más información sobre la privacidad de datos, consulte Data Privacy FAQ (Preguntas frecuentes sobre la privacidad de datos). Para obtener información sobre la protección de datos en Europa, consulte la publicación de blog The AWS Shared Responsibility Model and GDPR (Modelo de responsabilidad compartida y RGPD) en el AWS Security Blog. Para obtener más recursos sobre el cumplimiento de los requisitos del RGPD, consulte el Centro de reglamento general de protección de datos (RGPD).

Le recomendamos que nunca ingrese información confidencial que lo identifique, como números de cuenta de los usuarios finales, correos electrónicos u otro tipo de información personal, en los campos de formato libre. Cualquier dato que ingrese en Amazon CloudWatch Internet Monitor o en otros servicios se puede incluir en los registros de diagnóstico.

Identity and Access Management para Amazon CloudWatch Internet Monitor

AWS Identity and Access Management (IAM) es un Servicio de AWS que ayuda a los administradores a controlar de forma segura el acceso a los recursos de AWS. Los administradores de IAM controlan quién puede estar autenticado (ha iniciado sesión) y autorizado (tiene permisos) para utilizar los recursos de Internet Monitor. IAM es un Servicio de AWS que se puede utilizar sin cargo adicional.

Important

Cambios de recursos de Internet Monitor el 24 de febrero de 2023
Si creó políticas de IAM que incluían los recursos de Internet Monitor antes del 24 de febrero de 2023, tenga en cuenta los siguientes cambios en los recursos y tipos de recursos de Internet Monitor.

- El recurso HealthEvents pasó a llamarse HealthEvent.
- Se actualizaron los formatos ARN y Regex del recurso HealthEvent.
- Se actualizaron los formatos ARN y Regex del recurso Monitor.
- Los permisos a nivel de recurso para la acción GetHealthEvent ahora solo se admiten en el tipo de recurso HealthEvent. No se admiten en el recurso Monitor.
- Se actualizaron los tipos de recurso TagResource, UntagResource y ListTagsForResource para el tipo de recurso Monitor para que fueran necesarios.

Para obtener más información sobre las acciones, los recursos y las claves de condición que puede especificar en las políticas para administrar el acceso a AWS los recursos de Internet Monitor, consulte Acciones, recursos y claves de condición de Amazon CloudWatch Internet Monitor.

Contenido

Funcionamiento de Amazon CloudWatch Internet Monitor con IAM

Antes de utilizar IAM para administrar el acceso a Internet Monitor, descubra qué características de IAM se pueden utilizar con Internet Monitor.

Para ver las tablas que muestran una perspectiva general similar de cómo funcionan los servicios de AWS con la mayoría de las características de IAM, consulte los servicios de AWS que funcionan con IAM en la Guía del usuario de IAM.

Características de IAM que puede utilizar con Amazon CloudWatch Internet Monitor

Característica de IAM	Compatibilidad de Internet Monitor
Políticas basadas en identidades	Sí
Políticas basadas en recursos	No
Acciones de políticas	Sí
Recursos de políticas	Sí
Claves de condición de política (específicas del servicio)	Sí

Característica de IAM	Compatibilidad de Internet Monitor
ACL	No
ABAC (etiquetas en políticas)	Parcial
Credenciales temporales	Sí
Permisos de entidades principales	Sí
Roles de servicio	No
Roles vinculados al servicio	Sí

Políticas de Internet Monitor basadas en la identidad

Compatibilidad con las políticas basadas en identidad	Sí

Las políticas basadas en identidad son documentos de políticas de permisos JSON que puede asociar a una identidad, como un usuario de IAM, un grupo de usuarios o un rol. Estas políticas controlan qué acciones pueden realizar los usuarios y los roles, en qué recursos y en qué condiciones. Para obtener más información sobre cómo crear una política basada en identidad, consulte Creación de políticas de IAM en la Guía del usuario de IAM.

Con las políticas basadas en identidades de IAM, puede especificar las acciones y los recursos permitidos o denegados, así como las condiciones en las que se permiten o deniegan las acciones. No es posible especificar la entidad principal en una política basada en identidad porque se aplica al usuario o rol al que está adjunto. Para más información sobre los elementos que puede utilizar en una política de JSON, consulte Referencia de los elementos de las políticas de JSON de IAM en la Guía del usuario de IAM.

Políticas basadas en recursos en Internet Monitor

Compatibilidad con las políticas basadas en recursos	No

Las políticas basadas en recursos son documentos de política JSON que se asocian a un recurso. Ejemplos de políticas basadas en recursos son las políticas de confianza de roles de IAM y las políticas de bucket de Amazon S3. En los servicios que admiten políticas basadas en recursos, los administradores de servicios pueden utilizarlos para controlar el acceso a un recurso específico.

Acciones políticas para Internet Monitor

Admite acciones de política	Sí

Los administradores pueden utilizar las políticas JSON de AWS para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones.

El elemento `Action` de una política JSON describe las acciones que puede utilizar para conceder o denegar el acceso en una política. Las acciones de la política generalmente tienen el mismo nombre que la operación de API de AWS asociada. Hay algunas excepciones, como acciones de solo permiso que no tienen una operación de API coincidente. También hay algunas operaciones que requieren varias acciones en una política. Estas acciones adicionales se denominan acciones dependientes.

Incluya acciones en una política para conceder permisos y así llevar a cabo la operación asociada.

Para ver una lista de las acciones de Internet Monitor, consulte Acciones definidas por Amazon CloudWatch Internet Monitor en la Referencia de autorizaciones de servicio.

Las acciones de políticas de Internet Monitor utilizan el siguiente prefijo antes de la acción:

```
internetmonitor
```

Para especificar varias acciones en una única instrucción, sepárelas con comas.

```
"Action": [
      "internetmonitor:action1",
      "internetmonitor:action2"
         ]
```

Puede utilizar caracteres comodín (*) para especificar varias acciones . Por ejemplo, para especificar todas las acciones que comiencen con la palabra `Describe`, incluya la siguiente acción:

```
"Action": "internetmonitor:Describe*"
```

Recursos de políticas para Internet Monitor

Admite recursos de políticas	Sí

En la Referencia de autorizaciones de servicio, puede ver la siguiente información relacionada con Internet Monitor:

- Para ver una lista de los tipos de recursos de Internet Monitor y sus ARN, consulte Recursos definidos por Amazon CloudWatch Internet Monitor.
- Para obtener información sobre las acciones con las que puede especificar el ARN de cada recurso, consulte Acciones definidas por Amazon CloudWatch Internet Monitor.

Los administradores pueden utilizar las políticas JSON de AWS para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones.

El elemento `Resource` de la política JSON especifica el objeto u objetos a los que se aplica la acción. Las instrucciones deben contener un elemento `Resource` o `NotResource`. Como práctica recomendada, especifique un recurso utilizando el Nombre de recurso de Amazon (ARN). Puede hacerlo para acciones que admitan un tipo de recurso específico, conocido como permisos de nivel de recurso.

Para las acciones que no admiten permisos de nivel de recurso, como las operaciones de descripción, utilice un carácter comodín (*) para indicar que la instrucción se aplica a todos los recursos.

```
"Resource": "*"
```

Claves de condición de política para Internet Monitor

Admite claves de condición de políticas específicas del servicio	Sí

Los administradores pueden utilizar las políticas JSON de AWS para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones.

El elemento `Condition` (o bloque de `Condition`) permite especificar condiciones en las que entra en vigor una instrucción. El elemento `Condition` es opcional. Puede crear expresiones condicionales que utilicen operadores de condición, tales como igual o menor que, para que la condición de la política coincida con los valores de la solicitud.

Si especifica varios elementos de `Condition` en una instrucción o varias claves en un único elemento de `Condition`, AWS las evalúa mediante una operación `AND` lógica. Si especifica varios valores para una única clave de condición, AWS evalúa la condición con una operación lógica `OR`. Se deben cumplir todas las condiciones antes de que se concedan los permisos de la instrucción.

También puede utilizar variables de marcador de posición al especificar condiciones. Por ejemplo, puede conceder un permiso de usuario de IAM para acceder a un recurso solo si está etiquetado con su nombre de usuario de IAM. Para más información, consulte Elementos de la política de IAM: variables y etiquetas en la Guía del usuario de IAM.

AWS admite claves de condición globales y claves de condición específicas del servicio. Para ver todas las claves de condición globales de AWS, consulte Claves de contexto de condición globales de AWSen la Guía del usuario de IAM.

Para ver una lista de las claves de condición de Internet Monitor consulte Claves de condición para Amazon CloudWatch Internet Monitor en la Referencia de autorizaciones de servicio. Para obtener más información acerca de las acciones y los recursos con los que puede utilizar una clave de condición, consulte Acciones definidas por Amazon CloudWatch Internet Monitor.

ACL en Internet Monitor

Admite las ACL	No

Las listas de control de acceso (ACL) controlan qué entidades principales (miembros de cuentas, usuarios o roles) tienen permisos para acceder a un recurso. Las ACL son similares a las políticas basadas en recursos, aunque no utilizan el formato de documento de políticas JSON.

ABAC con monitor de Internet

Admite ABAC (etiquetas en las políticas)	Parcial

Internet Monitor admite parcialmente las etiquetas en las políticas. Admite el etiquetado de un recurso, los monitores.

Para usar etiquetas con Internet Monitor, use la AWS Command Line Interface o un SDK AWS. El etiquetado para Internet Monitor no es compatible con AWS Management Console.

Para obtener más información sobre el uso de etiquetas en las políticas en general, consulte la siguiente información.

El control de acceso basado en atributos (ABAC) es una estrategia de autorización que define permisos en función de atributos. En AWS, estos atributos se denominan etiquetas. Puede adjuntar etiquetas a entidades de IAM (usuarios o roles) y a muchos recursos de AWS. El etiquetado de entidades y recursos es el primer paso de ABAC. A continuación, designa las políticas de ABAC para permitir operaciones cuando la etiqueta de la entidad principal coincida con la etiqueta del recurso al que se intenta acceder.

ABAC es útil en entornos que crecen con rapidez y ayuda en situaciones en las que la administración de las políticas resulta engorrosa.

Para controlar el acceso en función de etiquetas, debe proporcionar información de las etiquetas en el elemento de condición de una política utilizando las claves de condición `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` o `aws:TagKeys`.

Si un servicio admite las tres claves de condición para cada tipo de recurso, el valor es Sí para el servicio. Si un servicio admite las tres claves de condición solo para algunos tipos de recursos, el valor es Parcial.

Para obtener más información sobre ABAC, consulte ¿Qué es ABAC? en la Guía del usuario de IAM. Para ver un tutorial con los pasos para configurar ABAC, consulte Uso del control de acceso basado en atributos (ABAC) en la Guía del usuario de IAM.

Uso de credenciales temporales con Internet Monitor

Compatible con el uso de credenciales temporales	Sí

Algunos Servicios de AWS no funcionan cuando inicia sesión con credenciales temporales. Para obtener información adicional, incluida la información sobre qué Servicios de AWS funcionan con credenciales temporales, consulte Servicios de AWS que funcionan con IAM en la Guía del usuario de IAM.

Utiliza credenciales temporales si inicia sesión en la AWS Management Console con cualquier método, excepto un nombre de usuario y una contraseña. Por ejemplo, cuando accede a AWS utilizando el enlace de inicio de sesión único (SSO) de la empresa, ese proceso crea automáticamente credenciales temporales. También crea automáticamente credenciales temporales cuando inicia sesión en la consola como usuario y luego cambia de rol. Para más información sobre el cambio de roles, consulte Cambio a un rol (consola) en la Guía del usuario de IAM.

Puede crear credenciales temporales de forma manual mediante la AWS CLI o la API de AWS. A continuación, puede usar esas credenciales temporales para acceder a AWS. AWS recomienda generar credenciales temporales de forma dinámica en lugar de usar claves de acceso a largo plazo. Para más información, consulte Credenciales de seguridad temporales en IAM.

Permisos de entidades principales entre servicios de Internet Monitor

Admite Forward access sessions (FAS)	Sí

Cuando utiliza un usuario o un rol de IAM para llevar a cabo acciones en AWS, se lo considera una entidad principal. Cuando utiliza algunos servicios, es posible que realice una acción que desencadene otra acción en un servicio diferente. FAS utiliza los permisos de la entidad principal para llamar a un Servicio de AWS, combinados con el Servicio de AWS solicitante para realizar solicitudes a servicios posteriores. Las solicitudes de FAS solo se realizan cuando un servicio recibe una solicitud que requiere interacciones con otros Servicios de AWS o recursos para completarse. En este caso, debe tener permisos para realizar ambas acciones. Para obtener información sobre las políticas a la hora de realizar solicitudes de FAS, consulte Reenviar sesiones de acceso.

Funciones de servicio para Internet Monitor

Compatible con roles de servicio	No

Un rol de servicio es un rol de IAM que asume un servicio para realizar acciones en su nombre. Un administrador de IAM puede crear, modificar y eliminar un rol de servicio desde IAM. Para obtener más información, consulte Creación de un rol para delegar permisos a un Servicio de AWS en la Guía del usuario de IAM.

Uso de un rol vinculado a servicio para Internet Monitor

Compatible con roles vinculados al servicio	Sí

Un rol vinculado al servicio es un tipo de rol de servicio que está vinculado a un Servicio de AWS. El servicio puede asumir el rol para realizar una acción en su nombre. Los roles vinculados a servicios aparecen en la Cuenta de AWS y son propiedad del servicio. Un administrador de IAM puede ver, pero no editar, los permisos de los roles vinculados a servicios.

Para obtener más información sobre el rol vinculado a los servicios de Internet Monitor, consulte Rol vinculado al servicio para Amazon CloudWatch Internet Monitor.

Para obtener más información acerca de cómo crear o administrar roles vinculados a servicios en AWS, consulte Servicios de AWS que funcionan con IAM. Busque un servicio en la tabla que incluya `Yes` en la columna Rol vinculado a un servicio. Seleccione el vínculo Sí para ver la documentación acerca del rol vinculado a servicios para ese servicio.

Políticas administradas de AWS para Amazon CloudWatch Internet Monitor

Una política administrada de AWS es una política independiente que AWS crea y administra. Las políticas administradas de AWS se diseñan para ofrecer permisos para muchos casos de uso comunes, por lo que puede empezar a asignar permisos a los usuarios, grupos y roles.

Considere que es posible que las políticas administradas por AWS no concedan permisos de privilegio mínimo para los casos de uso concretos, ya que están disponibles para que las utilicen todos los clientes de AWS. Se recomienda definir políticas administradas por el cliente específicas para sus casos de uso a fin de reducir aún más los permisos.

No puede cambiar los permisos definidos en las políticas administradas de AWS. Si AWS actualiza los permisos definidos en una política administrada de AWS, la actualización afecta a todas las identidades de entidades principales (usuarios, grupos y roles) a las que está adjunta la política. Lo más probable es que AWS actualice una política administrada de AWS cuando se lance un nuevo Servicio de AWS o las operaciones de la API nuevas estén disponibles para los servicios existentes.

Para obtener más información, consulte Políticas administradas de AWS en la Guía del usuario de IAM.

Política administrada de AWS: CloudWatchInternetMonitorServiceRolePolicy

Esta política está asociada al rol vinculado al servicio denominado AWSServiceRoleForInternetMonitor para permitir que Internet Monitor acceda a los recursos de su cuenta, como los recursos de Amazon Virtual Private Cloud o los Equilibradores de carga de red, de modo que pueda seleccionarlos al crear un monitor. Para obtener más información, consulte Rol vinculado al servicio para Amazon CloudWatch Internet Monitor.

Permisos de IAM para Amazon CloudWatch Internet Monitor

Para acceder a las acciones para trabajar con monitores y datos en Amazon CloudWatch Internet Monitor, los usuarios deben tener los permisos correctos.

Para obtener más información sobre la seguridad en Amazon CloudWatch, consulte Identity and Access Management para Amazon CloudWatch.

Permisos para el acceso de solo lectura de Amazon CloudWatch Internet Monitor

Para acceder a las acciones de solo lectura para trabajar con monitores y datos de Amazon CloudWatch Internet Monitor, los usuarios deben haber iniciado sesión como rol o usuario con los siguientes permisos:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "cloudwatch:GetMetricData",
                "internetmonitor:Get*",
                "internetmonitor:List*",
                "internetmonitor:StartQuery",
                "internetmonitor:StopQuery",
                "logs:DescribeLogGroups",
                "logs:GetQueryResults",
                "logs:StartQuery",
                "logs:StopQuery"
                ],
            "Resource": "*"
        }
    ]
}
```

Permisos para el acceso total de Amazon CloudWatch Internet Monitor

Para crear un monitor en Amazon CloudWatch Internet Monitor y tener acceso total a las acciones para trabajar con monitores y datos de Internet Monitor, los usuarios deben iniciar sesión con un rol o usuario que tenga los siguientes permisos:

- Permisos para crear un rol vinculado a servicios asociado a Internet Monitor. Para obtener más información, consulte Rol vinculado al servicio para Amazon CloudWatch Internet Monitor.
- Permisos para las acciones que permiten el acceso total para trabajar con los monitores y datos de Internet Monitor.

> **Note**
>
> Si crea una política de permisos basada en identidades que sea más restrictiva, es posible que los usuarios que posean esa política no tengan acceso total para crear y trabajar con monitores y datos de Internet Monitor.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "internetmonitor:*"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "arn:aws:iam::*:role/aws-service-role/
internetmonitor.amazonaws.com/AWSServiceRoleForInternetMonitor",
            "Condition": {
                "StringLike": {
                    "iam:AWSServiceName": "internetmonitor.amazonaws.com"
                }
            }
        },
        {
```

```
            "Effect": "Allow",
            "Action": [
                "iam:AttachRolePolicy",
                "iam:PutRolePolicy"
            ],
            "Resource": "arn:aws:iam::*:role/aws-service-role/
internetmonitor.amazonaws.com/AWSServiceRoleForInternetMonitor"
        },
        {
            "Action": [
                "ec2:DescribeVpcs",
                "elasticloadbalancing:DescribeLoadBalancers",
                "workspaces:DescribeWorkspaceDirectories",
                "cloudfront:GetDistribution"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

Rol vinculado al servicio para Amazon CloudWatch Internet Monitor

Amazon CloudWatch Internet Monitor usa un rol vinculado al servicio de AWS Identity and Access Management (IAM). Un rol vinculado al servicio es un tipo único de rol de IAM que está vinculado directamente a Internet Monitor. Internet Monitor predefine el rol vinculado al servicio e incluye todos los permisos que el servicio requiere para llamar a otros servicios de AWS en su nombre.

Internet Monitor define los permisos del rol vinculado al servicio y, a menos que esté definido de otra manera, solo Internet Monitor puede asumir el rol. Los permisos definidos incluyen las políticas de confianza y de permisos, y que la política de permisos no se pueda adjuntar a ninguna otra entidad de IAM.

Los roles solo se pueden eliminar después de eliminar primero sus recursos relacionados. Esta restricción protege los recursos de Internet Monitor, ya que evita que se puedan quitar accidentalmente permisos de acceso a ellos.

Para obtener información sobre otros servicios que admiten roles vinculados al servicio, consulte Servicios de AWS que funcionan con IAM y busque los servicios que muestran Yes (Sí) en la columna Service-linked role (Rol vinculado al servicio). Elija una opción Sí con un enlace para ver la documentación acerca del rol vinculado a servicios en cuestión.

Permisos de un rol vinculado al servicio para Internet Monitor

Internet Monitor utiliza el rol vinculado al servicio denominado AWSServiceRoleForInternetMonitor. Esta función permite a Internet Monitor acceder a los recursos de la cuenta, como los recursos de Amazon Virtual Private Cloud, las distribuciones de Amazon CloudFront, los directorios de Amazon WorkSpaces y los Equilibradores de carga de red, de modo que pueda seleccionarlos al crear un monitor.

Este rol vinculado al servicio utiliza la política administrada `CloudWatchInternetMonitorServiceRolePolicy`.

El rol vinculado al servicio AWSServiceRoleForInternetMonitor confía en el siguiente servicio para que asuma el rol:

- `internetmonitor.amazonaws.com`

Para ver los permisos de esta política, consulte CloudWatchInternetMonitorServiceRolePolicy en la Referencia de políticas administradas de AWS.

Creación de un rol vinculado al servicio para Internet Monitor

No es necesario crear un rol vinculado al servicio de forma manual para Internet Monitor. La primera vez que cree un monitor, Internet Monitor crea a su vez AWSServiceRoleForInternetMonitor.

Para obtener más información, consulte Creating a service-linked role en la Guía del usuario de IAM.

Edición de un rol vinculado al servicio para Internet Monitor

Después de que Internet Monitor cree un rol vinculado al servicio en su cuenta, no puede cambiarle el nombre, ya que varias entidades pueden hacer referencia a este. Sin embargo, puede editar la descripción del rol mediante IAM. Para obtener más información, consulte Editar un rol vinculado a servicios en la Guía del usuario de IAM.

Eliminación de un rol vinculado al servicio para Internet Monitor

Si ya no necesita utilizar una característica o servicio que requiere un rol vinculado a un servicio, le recomendamos que elimine el rol. De esta forma no tiene una entidad no utilizada que no se supervise ni mantenga de forma activa. Sin embargo, debe limpiar los recursos del rol vinculado al servicio antes de eliminarlo manualmente.

Tras eliminar los recursos de los monitores en Internet Monitor y, a continuación, los mismos monitores, puede eliminar el rol vinculado al servicio AWSServiceRoleForInternetMonitor.

> **Note**
>
> Si el servicio Internet Monitor está usando el rol cuando intenta eliminar los recursos, la eliminación podría producir un error. Si eso sucede, espere unos minutos e inténtelo de nuevo.

Eliminación manual del rol vinculado a servicios mediante IAM

Utilice la consola de IAM, la AWS CLI o la API de AWS para eliminar el rol vinculado al servicio AWSServiceRoleForInternetMonitor. Para obtener más información, consulte Eliminación de un rol vinculado a servicios en la Guía del usuario de IAM.

Actualizaciones del rol vinculado al servicio Internet Monitor

Para ver las actualizaciones de AWSServiceRoleForInternetMonitor, la política administrada de AWS para el rol vinculado al servicio Internet Monitor, consulte Actualizaciones de CloudWatch a las políticas administradas de AWS. Para recibir alertas automáticas sobre cambios en las políticas gestionadas en CloudWatch, suscríbase a la fuente RSS en la página de Historial de documentos de CloudWatch.

Cuotas de Amazon CloudWatch Internet Monitor

Amazon CloudWatch Internet Monitor tiene las siguientes cuotas.

Recurso	Cuota predeterminada
Monitores por región	50
Recursos por monitor	50
Se retienen los días en los que se resolvieron los eventos de estado de Internet Monitor	400

Uso de Amazon CloudWatch Network Monitor

Amazon CloudWatch Network Monitor proporciona visibilidad del rendimiento de la red que conecta las aplicaciones alojadas en AWS con los destinos en las instalaciones y le permite identificar el

origen de cualquier degradación del rendimiento de la red en cuestión de minutos. Network Monitor está totalmente administrado por AWS. Por lo tanto, no necesitará instalar agentes adicionales para supervisar el rendimiento de la red. Puede visualizar rápidamente la pérdida de paquetes y la latencia de las conexiones de red híbridas, establecer alertas y umbrales y, a continuación, tomar medidas para mejorar la experiencia de red de sus usuarios finales.

Network Monitor está pensado para operadores de redes y desarrolladores de aplicaciones que desean obtener información en tiempo real sobre el rendimiento de la red.

Características principales

- Utilice Network Monitor para comparar el cambiante entorno de red híbrida con métricas continuas de latencia y pérdida de paquetes en tiempo real.
- Cuando se conecta mediante el uso de AWS Direct Connect, Network Monitor diagnostica rápidamente la degradación de la red al escribir el indicador de salud de la red de AWS en la cuenta de CloudWatch. Esta métrica proporciona una puntuación probabilística para determinar si la degradación de la red fue dentro de AWS.
- Network Monitor proporciona una supervisión fluida con un enfoque de agentes totalmente gestionado, lo que significa que no es necesario instalar los agentes ni en las VPC ni en las instalaciones. Solo tiene que especificar una subred de VPC y una dirección IP en las instalaciones para empezar.
- Network Monitor publica las métricas en CloudWatch Metrics. Puede crear paneles para ver las métricas y crear umbrales y alarmas procesables en las métricas específicas de su aplicación.

Para obtener más información, consulte the section called “Funcionamiento de Network Monitor”.

Terminología y componentes de Network Monitor

- Monitor: un monitor muestra los recursos para los que desea ver las mediciones de rendimiento y disponibilidad de la red y sobre la que desea recibir alertas sobre eventos de estado. Al crear un monitor para una aplicación, se añade un recurso alojado en AWS como origen de red. A continuación, Network Monitor crea una lista de todos los sondeos posibles entre los recursos alojados en AWS y las direcciones IP de destino.
- Sondeos: un sondeo es el tráfico que se envía desde el recurso alojado en AWS a la dirección IP de destino en las instalaciones. Las métricas de Network Monitor se escriben en la cuenta de CloudWatch para cada sonda que se configura en un monitor.

- Origen de red de AWS: es el origen de AWS de la sonda de un monitor de red, que será una subred en cualquiera de las VPC.
- Destino: es el destino del origen de red de AWS en la red en las instalaciones. El destino es una combinación de las direcciones IP en las instalaciones, los protocolos de red, los puertos y el tamaño de los paquetes de red. Se admiten tanto IPv4 como IPv6.

Limitaciones y requisitos de Network Monitor

- Network Monitor admite un máximo de cuatro direcciones IP de destino y hasta 24 sondas por monitor.
- Puede tener hasta 100 monitores por cuenta por región.
- Las subredes del monitor deben pertenecer a la misma cuenta que el monitor.
- Network Monitor no proporciona una conmutación por error automática de red en caso de que se produzca un problema con la red AWS.
- Se aplica un cargo por cada sonda que cree. Para obtener más información sobre los precios, consulte the section called “Precios”.

Funcionamiento de Amazon CloudWatch Network Monitor

Network Monitor facilita la supervisión al proporcionar una solución totalmente administrada y sin agentes. Cuando crea un monitor en el recurso alojado en AWS, AWS crea y administra toda la infraestructura en segundo plano para realizar mediciones de ida y vuelta del tiempo y de pérdida de paquetes. Como resultado, puede escalar la supervisión rápidamente sin necesidad de instalar o desinstalar ningún agente de la infraestructura AWS.

Network Monitor centra la supervisión en las rutas que siguen los flujos desde los recursos alojados en AWS, en lugar de supervisar de forma amplia todos los flujos procedentes de Región de AWS. Si las cargas de trabajo se distribuyen en varias zonas de disponibilidad (AZ), Network Monitor puede monitorear las rutas desde cada una de las subredes privadas.

Network Monitor publica las métricas de tiempo de ida y vuelta y de pérdida de paquetes en la cuenta de Amazon CloudWatch en función del intervalo de agregación establecido al crear un monitor. También puede establecer umbrales individuales de latencia y pérdida de paquetes para cada monitor mediante CloudWatch. Por ejemplo, puede crear una alarma que le notifique si el promedio de pérdida de paquetes es superior al umbral estático del 0,1 % para una carga de trabajo sensible a

la pérdida de paquetes. También puede usar la detección de anomalías de CloudWatch para alertar sobre las métricas de pérdida de paquetes o latencia fuera de los rangos deseados.

Mediciones de disponibilidad y rendimiento

Network Monitor envía periódicamente sondeos activos desde el recurso de AWS a los destinos en las instalaciones. Al crear un monitor, especifique lo siguiente:

- El intervalo de agregación. El tiempo, en segundos, durante el que CloudWatch recibe los resultados medidos. Será cada 30 o 60 segundos. El período de agregación que elija para el monitor se aplica a todas las sondas de ese monitor.
- El protocolo de la sonda. Cada sonda añadida a un monitor debe usar los protocolos del protocolo de mensajes de control de internet (ICMP) o del protocolo de control de transmisión (TCP). Consulte the section called “Protocolos de comunicación” para obtener más detalles.
- El tamaño del paquete. El tamaño, en bytes, de cada paquete transmitido entre el recurso alojado en AWS y el destino en una sola sonda. Cada sonda de un monitor puede tener su propio tamaño de paquete.

Para las métricas:

- La métrica del tiempo de ida y vuelta, medida en milisegundos, mide y registra una medida del rendimiento y registra el tiempo que tarda la sonda en transmitirse a la dirección IP de destino y en recibir la respuesta asociada.
- La métrica de pérdida de paquetes mide el porcentaje del total de paquetes enviados y registra el número de sondeos transmitidos que no recibieron una respuesta asociada, lo que implica que esos paquetes se perdieron efectivamente a lo largo de la ruta de la red.

Protocolos de comunicación compatibles

Las sondas basadas en ICMP transportan las solicitudes de eco de ICMP de los recursos alojados en AWS a la dirección de destino y esperan una respuesta de eco de ICMP de la dirección de destino. El monitor de red utiliza la información de los mensajes de solicitud y respuesta de eco del ICMP para calcular el tiempo de ida y vuelta y las métricas de pérdida de paquetes.

Las sondas basadas en TCP transportan los paquetes TCP SYN desde los recursos alojados en AWS hasta la dirección y el puerto de destino y esperan recibir un paquete TCP SYN+ACK o RST desde la dirección y el puerto de destino. El monitor de red utiliza la información de los mensajes

de solicitud y respuesta de eco del ICMP para calcular el tiempo de ida y vuelta y las métricas de pérdida de paquetes. Además, Network Monitor cambia periódicamente los puertos TCP de origen para aumentar la cobertura de la red, lo que, a su vez, puede aumentar la probabilidad de detectar la pérdida de paquetes.

Indicador de estado de la red de AWS

Network Monitor publica una métrica del indicador de estado de la red (NHI), que proporciona información sobre el rendimiento y la disponibilidad de la red para los destinos que se conectan a través de AWS Direct Connect. La métrica es una medida estadística del estado de la ruta de red AWS controlada desde el recurso alojado en AWS, que es donde se implementa el monitor, hasta la ubicación de Direct Connect.

Network Monitor emplea la detección de anomalías para calcular las caídas de disponibilidad o la degradación del rendimiento a lo largo de las rutas de red.

> **Note**
>
> Cada vez que cree un monitor nuevo, añada una sonda o vuelva a activar una sonda, el NHI de ese monitor se retrasará unas horas para permitirle a AWS recopilar datos que permitan detectar anomalías.

Para proporcionar la métrica de estado del NHI, Network Monitor aplica la correlación estadística entre los conjuntos de datos de muestra de AWS, así como a las métricas de pérdida de paquetes y latencia de ida y vuelta del tráfico para simular la ruta de la red. La métrica puede ser una de estas dos variables: 1 o 0. Un valor de 1 indica que Network Monitor observó una degradación de la red dentro de la ruta de red controlada por AWS. Un valor de 0 indica que Network Monitor no observó ninguna degradación de la red a lo largo de la ruta. Esto le permite solucionar problemas de red más rápido. Puede configurar alertas en la métrica NHI para estar informado sobre los problemas actuales en las rutas de la red.

Soporte para direcciones IPv4 e IPv6

Network Monitor proporciona métricas de disponibilidad y rendimiento en redes IPv4 o IPv6 y puede monitorear direcciones IPv4 o IPv6 desde VPC de doble pila. El monitor de red no permite configurar los destinos de IPv4 e IPv6 en el mismo monitor, pero puede crear destinos independientes para solo IPv4 y solo para IPv6.

Disponibilidad por región

Network Monitor está disponible actualmente en los siguientes elementos de Regiones de AWS:

Región	
Asia-Pacífico (Hong Kong)	ap-east-1
Asia-Pacífico (Bombay)	ap-south-1
Asia-Pacífico (Seúl)	ap-northeast-2
Asia-Pacífico (Singapur)	ap-southeast-1
Asia-Pacífico (Sídney)	ap-southeast-2
Asia-Pacífico (Tokio)	ap-northeast-1
Oeste de Canadá (Calgary	ca-west-1
Europa (Fráncfort)	eu-central-1

Región	
Europa (Irlanda)	eu-west-1
Europa (Londres)	eu-west-2
Europa (París)	eu-west-3
Europa (Estocolmo)	eu-north-1
Medio Oriente (Baréin)	me-south-1
América del Sur (São Paulo)	sa-east-1
Este de EE. UU. (Norte de Virginia)	us-east-1
US East (Ohio)	us-east-2
Oeste de EE. UU. (Norte de California)	us-west-1

Región	
Oeste de EE. UU. (Oregón)	us-west-2

Creación de un monitor de red

Los siguientes pasos describen la creación de un monitor y, a continuación, la adición de las sondas necesarias. Para las sondas, elegirá la subred de origen y hasta cuatro direcciones IP de destino para un máximo de 24 sondas por monitor. Para crear un monitor puede utilizar la consola de Amazon CloudWatch o utilizar la línea de comando o API.

Temas

- Creación de un monitor de red con la consola
- Creación de un Network Monitor mediante la línea de comandos o la API

Creación de un monitor de red con la consola

Los siguientes pasos describen la creación de un monitor mediante la consola de Amazon CloudWatch. Elegirá las subredes de origen y, a continuación, añadirá hasta cuatro destinos para crear hasta 24 sondas por monitor. Puede crear un monitor mediante la consola de Amazon CloudWatch, la línea de comandos o el SDK.

> **Important**
>
> Estos pasos están diseñados para completarse todos de una vez. No podrá guardar ningún trabajo en proceso para continuar más adelante.

Definición de los detalles del monitor

El primer paso para crear un monitor es definir los detalles básicos. Esto incluye asignar un nombre al monitor y definir el período de agregación. Puede añadir etiquetas opcionales al monitor.

Cómo definir los detalles del monitor

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/ y, a continuación, en Supervisión de red, elija Network Monitor.
2. Elija Crear monitor.
3. En Nombre del monitor, ingrese el nombre que desee utilizar para este monitor.
4. Para el Período de agregación, elija la frecuencia con la que desea enviar métricas a CloudWatch. Los períodos de agregación disponibles son:

   - 30 segundos
   - 60 segundos

   > **Note**
   >
   > Un período de agregación más corto permite detectar más rápidamente los problemas de la red; sin embargo, el período de agregación que elija puede afectar la estructura de facturación. Para obtener más información acerca de los precios, consulte la página Precios de Amazon CloudWatch.

5. (Opcional) En la sección Etiquetas, añada pares de clave y valor para ayudar a identificar este recurso, lo que le permitirá buscar o filtrar información específica.

   1. Elija Añadir nueva etiqueta.
   2. Introduzca un nombre de clave y un valor asociado.
   3. Elija Añadir nueva etiqueta para añadir esa nueva etiqueta.

      Puede añadir varias etiquetas seleccionando Añadir nueva etiqueta, o puede eliminar cualquier etiqueta pulsando Eliminar.

   4. Si desea asociar las etiquetas al monitor, mantenga marcada la opción Añadir etiquetas a las sondas creadas por el monitor. Esto añade las etiquetas a las sondas del monitor, lo que puede resultar útil si utiliza la autenticación o la medición basadas en etiquetas.

6. Para the section called “Elección del origen y el destino”, elija Siguiente.

Elección del origen y el destino

Un monitor de red utiliza un origen AWS para las VPC y las subredes asociadas en las regiones en las que opera la red. El destino de un monitor es la combinación de las direcciones IP en las instalaciones, los protocolos de red, los puertos y el tamaño de los paquetes de red.

La combinación de origen y destino se denomina sonda. Puede tener hasta cuatro sondas por subred y un total de 24 sondas por monitor.

> **Important**
>
> Estos pasos están diseñados para completarse todos de una vez. No podrá guardar ningún trabajo en proceso para continuar más adelante.

Cómo elegir un origen y un destino

1. En origen de red de AWS, elija una o más subredes para incluirlas en el monitor. Puede elegir una sola VPC, que luego elegirá todas las subredes de esa VPC, o puede elegir subredes específicas. Las VPC y las subredes que elija serán el origen del monitor de red.
2. En Destino 1, introduzca la dirección IP de destino de la red en las instalaciones. Se admiten tanto IPv4 como IPv6.
3. Seleccione Ajustes avanzados.
4. Para este destino administrado por el cliente, elija el protocolo de red. Puede ser alguno de los siguientes:
   - ICMP
   - TCP
5. Si el protocolo es TCP, introduzca la siguiente información. De no ser así, vaya al siguiente paso:
   1. Introduzca el puerto que utiliza la red para conectarse. El puerto debe ser un número comprendido entre 1 y 65535.
   2. Introduzca el tamaño del paquete. Es el tamaño, en bytes, de cada paquete que se envía a la sonda entre el origen y el destino. El tamaño del paquete debe ser un número comprendido entre 56 y 8500.
6. Elija Añadir destino para añadir otro destino en las instalaciones a este monitor. Repita estos pasos para cada destino que desee añadir.

7. Elija Siguiente cuando haya terminado para confirmar las sondas.

Confirmación de sondas

La confirmación de las sondas le permite revisar la combinación de sondas de red del monitor. Esta página muestra todas las combinaciones posibles de los orígenes y destinos que ha elegido. Por ejemplo, si tiene seis subredes de origen y cuatro IP de destino, dispondrá de un total de 24 combinaciones de sondeo posibles.

> **Important**
>
> - Estos pasos están diseñados para completarse todos de una vez. No podrá guardar ningún trabajo en proceso para continuar más adelante.
> - La página de confirmación de sondeos no indica si un sondeo es válido. Por lo tanto, le recomendamos que revise detenidamente esta página y elimine cualquier sondeo no válido. Si no elimina las sondas no válidas, es posible que se le cobre por ellas.

Cómo confirmar las sondas de monitoreo

1. Requisito previo: the section called “Elección del origen y el destino”.
2. En la página Confirmar sondas, revise la lista de combinaciones de origen y destino.
3. Elija una o más sondas que desee eliminar del monitor y, a continuación, seleccione Eliminar.

   > **Note**
   >
   > No se le pedirá que confirme la eliminación. Una vez eliminada una sonda, debe volver a configurarla. Puede volver a añadir una sonda a un monitor desde la sección Monitores de red de la página Network Monitor. Para obtener más información, consulte the section called “Adición de una sonda a un monitor”.

4. Seleccione Siguiente para revisar los detalles del monitor antes de crearlo.

Revisar y crear

El último paso para crear un monitor y sondas consiste en revisar los detalles tanto del monitor como de las sondas. Puede cambiar cualquier información en este momento. Cuando termine de revisar y

crear el monitor y empiece a realizar el seguimiento de las métricas, se le empezará a cobrar por los sondeos.

> **Important**
>
> - Este paso está diseñado para que se complete de una sola vez al crear un monitor y una sonda. No podrá guardar ningún trabajo en proceso para continuar más adelante.
> - Si decide editar alguna sección, tendrá que ir creando el monitor desde el punto en el que está editando. Sin embargo, no tendrá que volver a realizar ningún paso posterior. Estas páginas mantienen la información rellenada anteriormente.

Cómo revisar y crear un monitor

1. En la página Revisar y crear sondeos, elija Editar para cualquier sección en la que desee realizar cambios.
2. Realice los cambios que desee en esa sección.
3. Elija Siguiente.
4. Realice uno de los siguientes procedimientos:
   - Realice los cambios que desee en otras páginas del monitor y seleccione Siguiente hasta que vuelva a la página de Revisión y creación.
   - Si no hay ninguna otra página que requiera cambios, selecciona Siguiente hasta que vuelva a la página de Revisión y creación.
5. Elija Crear monitor.

   La página Network Monitor muestra el estado actual de la creación del monitor en la sección Network monitors. Durante la creación del monitor, el Estado es Pendiente. Cuando el Estado cambie a Activo, podrá acceder al panel de control del monitor para ver las métricas de CloudWatch.

   Para más información sobre cómo trabajar con el panel de control del monitor, consulte the section called “Paneles de control de Network Monitor”.

> **Note**
>
> Podría llevarle varios minutos al monitor de red recién añadido comenzar a recopilar métricas de red.

Creación de un Network Monitor mediante la línea de comandos o la API

Utilice la línea de comando o API para ver y crear un monitor de red.

Cómo crear un monitor de red mediante la línea de comando o API

1. Cree un monitor de red mediante create-monitor.
2. Cree una sonda de monitor de red mediante create-probe.

Trabajo con monitores y sondas de Network Monitor

Puede realizar cualquiera de las siguientes tareas con los monitores y sondas mediante la consola de Amazon CloudWatch, la línea de comandos o la API.

Temas:

- Edición de un monitor
- Eliminación de un monitor
- Activación o desactivación de una sonda
- Adición de una sonda a un monitor
- Edición de una sonda
- Eliminación de una sonda
- Cómo etiquetar o eliminar la etiqueta de los recursos mediante la línea de comandos o la API

Edición de un monitor

Puede editar cualquier información de Network Monitor, incluido el cambio de nombre, la configuración de un nuevo período de agregación o la adición o eliminación de etiquetas. Al cambiar la información de un monitor no se modifican ninguna de las sondas asociadas. Para crear un monitor puede utilizar la consola de Amazon CloudWatch o utilizar la línea de comando o API.

Edición de un monitor a través de la consola

Utilice la consola CloudWatch para editar un monitor.

Cómo editar un monitor con la consola

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/ y, a continuación, en Supervisión de red, elija Network Monitor.
2. En la sección Monitores de red, elija el monitor que desee editar.
3. En la página del panel de control del monitor, seleccione Editar.
4. Para Nombre del monitor, introduzca el nuevo nombre del monitor.
5. Para el Período de agregación, elija la frecuencia con la que desea enviar métricas a CloudWatch. Los períodos válidos son los siguientes:

   - 30 segundos
   - 60 segundos

   > **Note**
   >
   > Un período de agregación más corto permite detectar más rápidamente los problemas de la red; sin embargo, el período de agregación que elija puede afectar la estructura de facturación. Para obtener más información acerca de los precios, consulte la página Precios de Amazon CloudWatch.

6. (Opcional) En la sección Etiquetas, añada pares de clave y valor para ayudar a identificar este recurso, lo que le permitirá buscar o filtrar información específica. También puede simplemente cambiar el valor de cualquier clave actual.

   1. Elija Añadir nueva etiqueta.
   2. Introduzca un nombre de clave y un valor asociado.
   3. Elija Añadir nueva etiqueta para añadir esa nueva etiqueta.

      Puede añadir varias etiquetas seleccionando Añadir nueva etiqueta, o puede eliminar cualquier etiqueta pulsando Eliminar.
   4. Si desea asociar las etiquetas al monitor, mantenga marcada la opción Añadir etiquetas a las sondas creadas por el monitor. Esto añade las etiquetas a las sondas del monitor, lo que puede resultar útil si utiliza la autenticación o la medición basadas en etiquetas.

7. Elija Guardar cambios.

Edición de un monitor mediante la CLI o la API

Use la línea de comandos o la API para ver y editar un monitor.

Cómo editar un monitor mediante la línea de comandos o la API

1. Use list-monitors para obtener una lista de los monitores si no conoce el nombre del monitor. Anote el nombre del monitor que desea editar.
2. Utilice edit-monitor con el nombre del monitor del paso anterior.

Eliminación de un monitor

Antes de poder eliminar un monitor, debe desactivar o eliminar todas las sondas asociadas a ese monitor, independientemente del estado del monitor. Tras desactivar o eliminar un monitor, ya no se le cobrará por esas sondas de monitor. No se puede restaurar un monitor eliminado. Puede eliminar un monitor mediante la consola de Amazon CloudWatch o mediante la línea de comandos o la API.

Aunque es posible que se elimine o desactive una sonda, CloudWatch conserva las métricas durante 15 días.

Eliminación de un monitor a través de la consola

Utilice la consola CloudWatch para editar un monitor.

Cómo eliminar un monitor con la consola

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/ y, a continuación, en Supervisión de red, elija Network Monitor.
2. En la sección Monitores de red, elija el monitor que desea eliminar.
3. Elija Acciones y, a continuación, elija Eliminar.
4. Si tiene alguna sonda activa, se le solicitará que la desactive. Seleccione Desactivar sondas.

> **Note**
>
> No puede cancelar ni deshacer esta acción tras seleccionar Desactivar sondas. Sin embargo, las sondas desactivadas no se retiran del monitor. Puede reactivarlas más adelante. Consulte the section called “Activación o desactivación de una sonda”.

5. Ingrese **`confirm`** en el campo de confirmación y, a continuación, elija Eliminar.

Eliminación de un monitor mediante la línea de comandos o la API

Elimine un monitor mediante la línea de comandos o la API.

Cómo eliminar un monitor de red mediante la línea de comando o API

1. Necesitará el nombre del monitor que desea eliminar. Si no sabe el nombre, utilice list-monitors para obtener una lista de los monitores. Anote el nombre del monitor que desea eliminar.
2. Verifique si ese monitor contiene alguna sonda. Utilice get-monitor con el nombre del monitor del paso anterior. Esto devuelve una lista de todas las sondas asociadas a ese monitor.
3. Si el monitor contiene sondas, primero tendrá que configurarlas como inactivas o eliminarlas.
   - Para configurar una sonda como inactiva, utilice update-probe y establezca el estado en `INACTIVE`.
   - Para eliminar una sonda, utilice delete-probe.
4. Una vez que las sondas estén configuradas en `INACTIVE` o eliminadas, utilice delete-monitor para eliminar el monitor. Las sondas inactivas no se eliminan.

Activación o desactivación de una sonda

Puede activar o desactivar una sonda de monitor según sea necesario. Es posible que desee desactivar una sonda si no la está utilizando actualmente, pero es posible que desee volver a utilizarla en el futuro. Al desactivar una sonda, no tendrá que perder tiempo en volver a configurarla. No se cobrarán las sondas desactivadas.

Para crear un monitor puede utilizar la consola de Amazon CloudWatch o utilizar la línea de comando o API.

Configuración de una sonda como activa o inactiva mediante la consola

Utilice la consola CloudWatch para configurar una sonda como activa o inactiva.

Cómo configurar una sonda como activa o inactiva mediante la consola

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/ y, a continuación, en Supervisión de red, elija Network Monitor.
2. Seleccione la pestaña Detalles del monitor.
3. En la sección Sondas, elija la sonda que desee activar o desactivar.
4. Seleccione Acciones y, a continuación, seleccione Activar o Desactivar.

> **Note**
>
> Si reactiva una sonda desactivada, empezará a incurrir en gastos de facturación por esa sonda.

Configuración de una sonda como activa o inactiva mediante la línea de comando o API

Configure una sonda como activa o inactiva o desactiva mediante la línea de comandos o API. Solo puede usar este comando para una sola sonda.

Cómo configurar una sonda como activa o inactiva mediante la línea de comando o API

1. Use list-monitors para obtener una lista de los monitores si no conoce el nombre del monitor. Anote el nombre del monitor cuyo estado de sonda desea cambiar.
2. Utilice get-monitor con el nombre del monitor del paso anterior. Esto devuelve una lista de todas las sondas asociadas a ese monitor. Anote el ID de las sondas cuyo estado desea cambiar.
3. Utilice update-probe y configure la sonda a cuyo estado desee cambiar ya sea a `ACTIVE` o a `INACTIVE`.

Adición de una sonda a un monitor

Puede añadir una sonda a un monitor existente. Tenga en cuenta que si añade alguna sonda a un monitor, la estructura de facturación se actualizará para mostrar que se ha añadido una nueva sonda.

Adición de una sonda a un monitor con la consola

Cómo añadir una sonda a un monitor mediante la consola

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/ y, a continuación, en Supervisión de red, elija Network Monitor.
2. En la pestaña Monitores de red, realice una de las siguientes acciones:
   - Elija el enlace del Nombre del monitor al que desee añadir una sonda. Seleccione la pestaña Detalles del monitor y, a continuación, en la sección Sondas, elija Añadir sonda.
   - Seleccione la casilla de verificación del monitor, elija Acciones y, a continuación, elija Añadir sonda.
3. En la página Añadir sonda, haga lo siguiente:
   1. En Origen de red de AWS, elija una subred para añadirla al monitor.

      > **Note**
      >
      > Solo puede añadir una sonda a la vez y hasta cuatro sondas por monitor.

   2. Introduzca la dirección IP de destino de la red en las instalaciones. Se admiten tanto IPv4 como IPv6.
   3. Seleccione Ajustes avanzados.
   4. Elija el protocolo de red para el destino. Puede ser ICMP o TCP.
   5. Si el protocolo es TCP, introduzca la siguiente información. De no ser así, vaya al siguiente paso:
      - Introduzca el puerto que utiliza la red para conectarse. El puerto debe ser un número comprendido entre 1 y 65535.
      - Introduzca el tamaño del paquete. Es el tamaño, en bytes, de cada paquete enviado a lo largo de la sonda entre el origen y el destino. El tamaño del paquete debe ser un número comprendido entre 56 y 8500.
4. (Opcional) En la sección Etiquetas, añada pares de clave y valor para ayudar a identificar este recurso, lo que le permitirá buscar o filtrar información específica.
   1. Elija Añadir nueva etiqueta.
   2. Introduzca un nombre de clave y un valor asociado.

3. Elija Añadir nueva etiqueta, para añadir la nueva etiqueta.

   Puede añadir varias etiquetas seleccionando Añadir nueva etiqueta, o puede eliminar cualquier etiqueta pulsando Eliminar.

5. Seleccione Añadir sonda.

   Mientras se activa la sonda, el Estado muestra Pendiente. Podría llevar varios minutos para que la sonda se vuelva activa.

Adición de una sonda al monitor mediante la línea de comando o la API

Añada una sonda a un monitor mediante la línea de comandos o la API. Solo puede utilizar este comando para añadir una sola sonda a la vez.

Cómo añadir una sonda al monitor mediante la línea de comandos o una API

1. Use list-monitors para obtener una lista de los monitores si no conoce el nombre del monitor. Anote el nombre del monitor al que desea añadir una sonda.
2. Utilice create-probe para añadir una sonda al monitor.

Edición de una sonda

Puede cambiar cualquier información de una sonda actual, independientemente de si esa sonda está activada o desactivada. Puede editar una sonda utilizando la consola de Amazon CloudWatch o mediante la línea de comandos o API.

Edición de una sonda mediante la consola

Cómo editar una sonda a través de la consola

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/ y, a continuación, en Supervisión de red, elija Network Monitor.

   Elija el enlace Nombre para abrir el panel del monitor.

2. Seleccione la pestaña de Detalles del monitor.
3. En la sección Sondas, elija el enlace de la sonda que desea editar.
4. En la página del panel de control de la sonda, elija Editar o elija Acciones y, a continuación, elija Editar.

5. En la página Editar sonda, introduzca la nueva dirección IP de la sonda de destino. Se admiten tanto IPv4 como IPv6.
6. Seleccione Ajustes avanzados.
7. Elija el Protocolo de red. Puede ser ICMP o TCP.
8. Si el protocolo es TCP, introduzca la siguiente información. De no ser así, vaya al siguiente paso:

   - Introduzca el puerto que utiliza la red para conectarse. El puerto debe ser un número comprendido entre 1 y 65535.
   - Introduzca el tamaño del paquete. Es el tamaño, en bytes, de cada paquete enviado a lo largo de la sonda entre el origen y el destino. El tamaño del paquete debe ser un número comprendido entre 56 y 8500.
9. (Opcional) En la sección Etiquetas, añada pares de clave y valor para ayudar a identificar este recurso, lo que le permitirá buscar o filtrar información específica.

   1. Elija Añadir nueva etiqueta.
   2. Introduzca un nombre de clave y un valor asociado.
   3. Elija Añadir nueva etiqueta, para añadir la nueva etiqueta.

      Puede añadir varias etiquetas seleccionando Añadir nueva etiqueta, o puede eliminar cualquier etiqueta pulsando Eliminar.
10. Elija Guardar cambios.

Edición de una sonda mediante la línea de comandos o la API

Utilice la línea de comandos para editar una sonda de monitor. Solo puede usar este comando para una sola sonda.

Cómo editar una sonda mediante la línea de comandos o la API

1. Use list-monitors para obtener una lista de los monitores si no conoce el nombre del monitor. Anote el nombre del monitor cuyo estado de sonda desea cambiar.
2. Utilice get-monitor con el nombre del monitor del paso anterior. Esto devuelve una lista de todas las sondas asociadas a ese monitor. Anote el ID de las sondas que desee editar.
3. Utilice update-probe para cambiar la información de la sonda.

Eliminación de una sonda

Puede eliminar una sonda en lugar de desactivarla si sabe que no la volverá a necesitar en el futuro. No puede recuperar una sonda eliminada y, en su lugar, tendrá que volver a crearla. La facturación de esa sonda se detiene cuando se elimina la sonda. Puede eliminar una sonda utilizando la consola de Amazon CloudWatch de o la línea de comandos o API.

Eliminación de una sonda con la consola

Cómo eliminar una sonda con la consola

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/ y, a continuación, en Supervisión de red, elija Network Monitor.
2. En la sección Monitores de red, seleccione el enlace Nombre para abrir el panel del monitor.
3. Seleccione la pestaña de Detalles del monitor.
4. Seleccione la casilla de verificación del monitor, elija Acciones y, a continuación, elija Eliminar.
5. En el cuadro de diálogo Eliminar sonda, elija Eliminar para confirmar que desea eliminar la sonda.
6. Elija Eliminar para confirmar que quiere eliminar la sonda.

   El Estado de la sonda en la sección Sondas muestra Borrando. Una vez eliminada, la sonda se elimina de la sección Sondas.

Eliminación de una sonda mediante la línea de comandos o la API

Elimine la sonda mediante la línea de comandos o la API. Solo puede usar este comando para una sola sonda.

Cómo configurar una sonda como activa o inactiva mediante la línea de comando o API

1. Use list-monitors para obtener una lista de los monitores si no conoce el nombre del monitor. Cómo anotar el nombre del monitor con la sonda que desea eliminar
2. Utilice get-monitor con el nombre del monitor del paso anterior. Esto devuelve una lista de todas las sondas asociadas a ese monitor. Anote el ID de la sonda que desea eliminar.
3. Utilice delete-probe.

Cómo etiquetar o eliminar la etiqueta de los recursos mediante la línea de comandos o la API

Puede usar la línea de comandos o la CLI para añadir o actualizar etiquetas de recursos.

Cómo actualizar etiquetas de monitoreo de red mediante la línea de comandos o una API

- Para enumerar las etiquetas de un recurso, utilice list-tags-for-resources.
- Para etiquetar un recurso, utilice tag-resource.
- Para quitar la etiqueta de un recurso, utilice untag-resource.

Paneles de control de Network Monitor

Puede utilizar el panel de control de Amazon CloudWatch Network Monitor para ver el estado de la red de AWS y comprobar el tiempo de ida y vuelta y la pérdida de paquetes. Puede ver estas métricas tanto para los monitores como para las sondas individuales.

Paneles de control de Network Monitor

- Panel de control del monitor
- Panel de control de sonda

Alarmas de sonda

Puede crear alarmas de Amazon CloudWatch en función de las métricas de Amazon CloudWatch Network Monitor, del mismo modo que puede hacerlo con otras métricas de Amazon CloudWatch. Cualquier alarma que cree aparecerá en la columna Estado de la sonda en la sección Detalles del monitor del panel de Network Monitor cuando se active la alarma. El estado será OK o En alarma. Si no se muestra el estado de una sonda, significa que no se ha creado ninguna alarma para esa sonda.

Por ejemplo, puede crear una alarma basada en la métrica `PacketLoss` de Network Monitor, y configurarla para que envíe una notificación cuando la métrica sea inferior a un valor que usted elija. Las alarmas para las métricas de Network Monitor se configuran siguiendo las mismas pautas que para otras métricas de CloudWatch.

Las siguientes métricas están disponibles en la sección `AWS/NetworkMonitor` al crear una alarma de CloudWatch para Network Monitor.

- HealthIndicator
- PacketLoss
- RTT (tiempo de ida y vuelta)

Para conocer los pasos para crear una alarma de monitor de red, consulte the section called “Cree una alarma basada en un umbral estático”.

Cómo establecer un marco temporal de métricas

Las métricas y los eventos de ambos paneles utilizan un tiempo predeterminado de dos horas, calculado a partir de la hora actual. Puede cambiar el valor predeterminado para utilizar uno de los siguientes valores preestablecidos:

- 1h: una hora
- 2h: dos horas
- 1d: un día
- 1w: una semana

También puede configurar un marco de tiempo personalizado. Elija Personalizar, elija un tiempo absoluto o relativo y, a continuación, establezca el período de tiempo en el tiempo que elija. El tiempo relativo solo admite 15 días a partir de la fecha de hoy, según los valores predeterminados de CloudWatch.

Además, puede elegir la hora que se muestra en los gráficos en función de la zona horaria UTC o la zona horaria local.

Panel de control del monitor

Puede utilizar el panel de control de Amazon CloudWatch Network Monitor para ver el estado de la red de AWS y comprobar el tiempo de ida y vuelta y la pérdida de paquetes. Network Monitor tiene paneles de control tanto para monitores como para sondas.

Cómo acceder al panel de control de un monitor

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/ y, a continuación, en Supervisión de red, elija Network Monitor.
2. En la sección Monitores de red, seleccione el enlace Nombre para abrir el panel del monitor.

Información general

La página Información general muestra la siguiente información para el monitor:

- Estado de la red de AWS: el estado de la red de AWS muestra únicamente el estado general de la red de AWS. El estado será Saludable o Degradado. Un estado Saludable indica que el monitor de red no detectó ningún problema en la red AWS. Un estado Degradado indica que el monitor de red detectó un problema en la red AWS. La barra de estado de esta sección muestra el estado de la red durante un tiempo predeterminado de una hora. Pase el ratón sobre cualquier punto de la barra de estado para ver detalles adicionales.
- Resumen del tráfico de la sonda: muestra el estado actual del tráfico entre las subredes AWS de origen del monitor y las direcciones IP de destino. El resumen del tráfico de la sonda muestra lo siguiente:
  - Sondas en alarma: este número indica cuántas sondas se encuentran en un estado degradado. Se activa una alarma cuando se activa una métrica que ha configurado como alarma. Para obtener información sobre cómo crear alarmas de Network Monitor, consulte the section called “Alarmas de sonda”.
  - Pérdida de paquetes: la cantidad de paquetes que se perdieron de la subred de origen a la dirección IP de destino. Esto se representa como un porcentaje del total de paquetes enviados.
  - Tiempo de ida y vuelta: el tiempo, expresado en milisegundos, que tarda un paquete de la subred de origen en llegar a la dirección IP de destino y volver a aparecer.

Los datos se representan mediante un gráfico interactivo que permite ver los detalles.

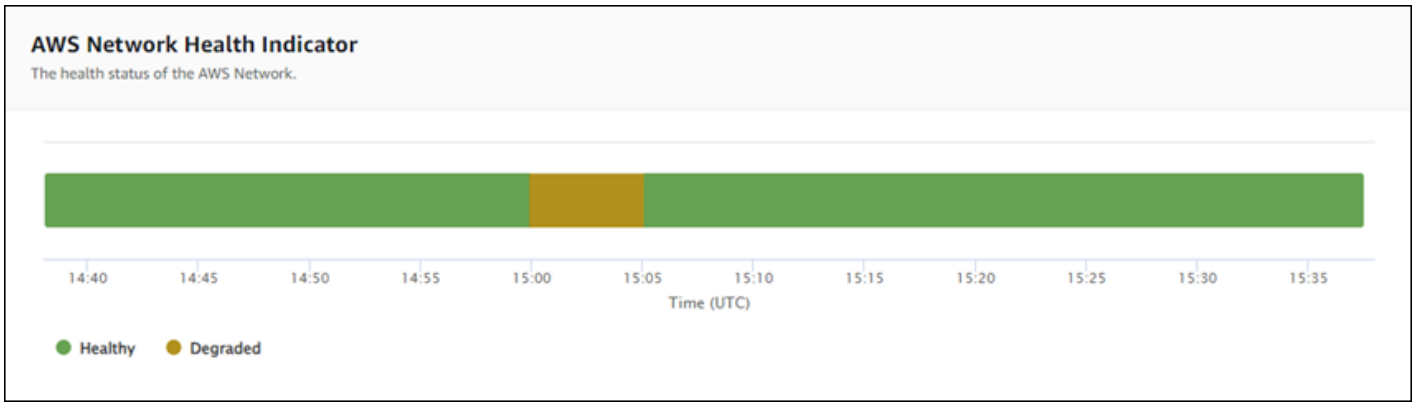
De forma predeterminada, los datos se muestran durante un período de dos horas, calculado a partir de la fecha y hora actuales. Sin embargo, puede cambiar el rango para adaptarlo a sus necesidades. Para obtener más información, consulte the section called “Cómo establecer un marco temporal de métricas”.

Métricas de seguimiento

El panel de control de Network Monitor muestra una representación gráfica de los monitores y las sondas. Están disponibles los siguientes gráficos:

- Indicador de salud de la red AWS: representa el estado de la red AWS durante un período específico. El estado será Saludable o Degradado. En el siguiente ejemplo, verá que desde las 15:00 UTC hasta las 15:05 UTC, la red AWS estuvo en un estado degradado. Después de las

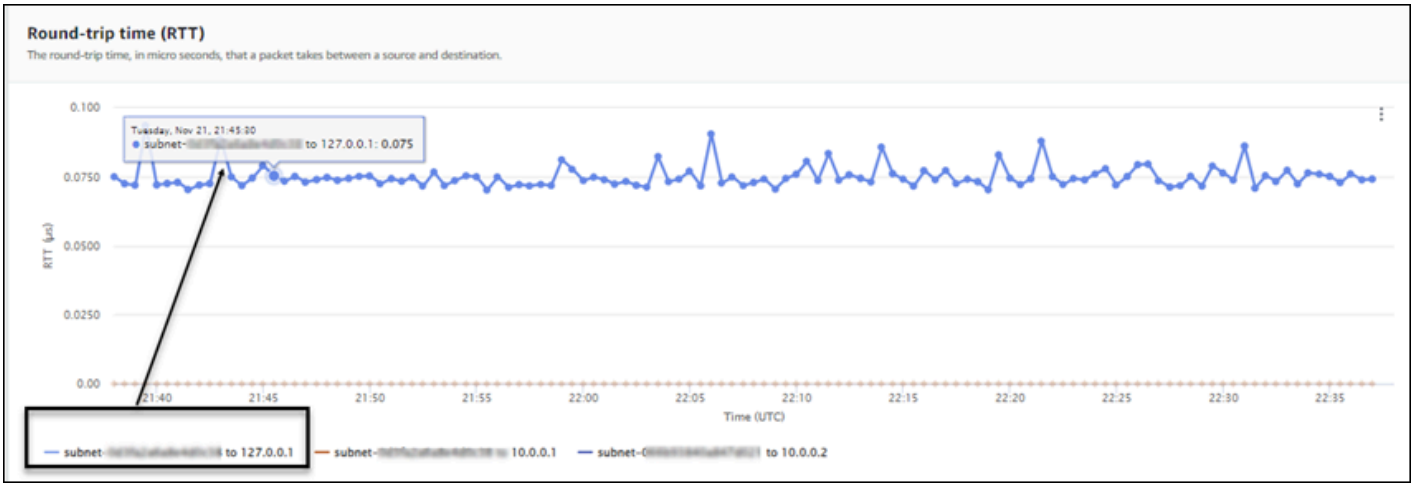
15:05 la red volvió a su estado saludable. Puede pasar el ratón sobre cualquier sección del gráfico para ver detalles adicionales.



> **Note**
>
> El indicador de salud de la red no indica el estado de la sonda, sino solo el de la red de AWS.

- Pérdida de paquetes: este gráfico muestra una línea única que muestra el porcentaje de pérdida de paquetes de cada sonda del monitor. La leyenda de la parte inferior de la página muestra cada una de las sondas del monitor, codificadas por colores para que sean únicas. Al pasar el ratón sobre una sonda en este gráfico, se muestran la subred de origen, la IP de destino y el porcentaje de pérdida de paquetes. En el siguiente ejemplo, se configuró una alarma de pérdida de paquetes para una sonda desde una subred a la dirección IP 127.0.0.1. La alarma se activaba cuando se superaba el umbral de pérdida de paquetes de la sonda. Al pasar el ratón sobre el gráfico, se muestran el origen y el destino de la sonda, y se observa que esta sonda perdió un 30,97 % de paquetes el 21 de noviembre a las 02:41:30.

- Tiempo de ida y vuelta: este gráfico muestra una línea para cada sonda, que muestra el tiempo de ida y vuelta de cada sonda. La leyenda de la parte inferior de la página muestra cada una de las sondas del monitor, codificadas por colores para que sean únicas. Al pasar el ratón sobre una sonda en este gráfico, se muestran la subred de origen, la dirección IP de destino y el tiempo de ida y vuelta. El siguiente ejemplo muestra que el martes 21 de noviembre a las 21:45:30, el tiempo de ida y vuelta de una sonda desde una subred a la dirección IP 127.0.0.1 fue de 0,075 segundos.



Detalles del monitor

La página de detalles del monitor muestra los detalles del monitor, incluidas las sondas. En esta página puede gestionar las etiquetas o añadir una sonda. La página está dividida en las tres secciones siguientes:

- Detalles del monitor: en esta página se proporcionan detalles sobre el monitor. La información de esta sección no se puede editar. Sin embargo, puede elegir el enlace del nombre del rol para ver los detalles del rol vinculado al servicio de Network Monitor.
- Sondas: en esta sección se muestra una lista de todas las sondas asociadas al monitor. Elija un enlace de VPC o ID de subred para abrir los detalles de la VPC o la subred en la consola de Amazon VPC. También puede modificar una sonda, incluida su activación o desactivación. Para obtener más información, consulte the section called “Trabajo con monitores y sondas”.

  La sección Sondas muestra información sobre cada sonda configurada para ese monitor, incluido el ID de sonda, el ID de VPC, el ID de subred, la dirección IP, el Protocolo y si el Estado de la sonda es Activo o Inactivo. Si ha configurado una alarma para una sonda, se muestra el Estado actual de esa alarma. OK indica que no hay eventos de métricas que hayan activado ninguna alarma; En alarma indica que una métrica que configuró en CloudWatch activó una alarma. Si no se muestra el estado de una sonda, significa que no se ha configurado ninguna alarma de CloudWatch. Para obtener información sobre los tipos de alarmas de sonda de Network Monitor que puede crear, consulte the section called “Alarmas de sonda”.
- Etiquetas: permite ver las etiquetas actuales de un monitor. Puede añadir o eliminar etiquetas seleccionando Administrar etiquetas. Esto abre la página Editar sonda. Para obtener más información sobre la edición de las etiquetas, consulte the section called “Edición de un monitor”.

Panel de control de sonda

Puede utilizar el panel de control de Amazon CloudWatch Network Monitor para ver el estado de la red de AWS e información sobre el tiempo de ida y vuelta específico y la pérdida de paquetes para sondeos específicos. Hay dos paneles de control de la sonda: Información general y Detalles de la sonda.

Puede crear alarmas de CloudWatch para establecer los umbrales de métricas de pérdida de paquetes y tiempo de ida y vuelta. Cuando se alcanza un umbral para una métrica, se lo notifica una alarma de CloudWatch. Para obtener más información sobre la creación de alarmas de sondas, consulte the section called “Alarmas de sonda”.

Acceso al panel de control de una sonda

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/ y, a continuación, en Supervisión de red, elija Network Monitor.
2. En la sección Monitores de red, seleccione el enlace Nombre para abrir el panel del monitor.
3. Elija el enlace de ID para ver el panel de control de esa sonda.

Información general

La página Información general muestra la siguiente información para la sonda:

- Detalles del indicador de salud de la red de AWS: proporciona el estado general únicamente de la red de AWS. El estado será Saludable o Degradado. El estado Degradado indica que hay un problema con la red AWS y no indica si hay un problema con la sonda.
- Pérdida de paquetes: la cantidad de paquetes que se perdieron de la subred de origen a la dirección IP de destino de esta sonda.
- Tiempo de ida y vuelta: el tiempo, en milisegundos, que tarda un paquete de la subred de origen en llegar a la dirección IP de destino y volver a aparecer.

Detalles de la sonda

La página de detalles de la sonda muestra los detalles de una sonda. En esta página puede editar la sonda. Para obtener más información, consulte the section called “Trabajo con monitores y sondas”.

- Detalles de la sonda: esta página proporciona información general sobre la sonda. La información de esta sección no se puede editar.
- Origen y destino de la sonda: en esta sección se muestran detalles sobre la sonda. Elija un enlace de VPC o ID de subred para abrir los detalles de la VPC o la subred en la consola de Amazon VPC. También puede modificar una sonda, incluida su activación o desactivación.
- Etiquetas: permite ver las etiquetas actuales de un monitor. Puede añadir o eliminar etiquetas seleccionando Administrar etiquetas. Esto abre la página Editar sonda. Para obtener más información sobre la edición de las etiquetas, consulte the section called “Edición de una sonda ”.

Cuotas de Network Monitor

Las cuotas de Network Monitor son las siguientes:

Cuota	Predeterminado	Ajustable
Número máximo de monitores por cuenta por Región de AWS	100	Sí
Número máximo de sondas por monitor	24	Sí
Número máximo de sondas por subred por monitor	4	Sí

Seguridad y protección de datos en Network Monitor

La seguridad en la nube de AWS es la mayor prioridad. Como cliente de AWS, se beneficiará de una arquitectura de red y de centros de datos diseñados para satisfacer los requisitos de seguridad de las organizaciones más exigentes.

La seguridad es una responsabilidad compartida entre AWS y usted. El modelo de responsabilidad compartida la describe como seguridad de la nube y seguridad en la nube:

- Seguridad de la nube: AWS es responsable de proteger la infraestructura que ejecuta servicios de AWS en la Nube de AWS. AWS también le proporciona servicios que puede utilizar de forma segura. Los auditores externos prueban y verifican periódicamente la eficacia de nuestra seguridad como parte de los Programas de conformidad de AWS . Para obtener información sobre los programas de conformidad que se aplican a Amazon CloudWatch Network Monitor, consulte Servicios en el ámbito por programa de conformidad de AWS.
- Seguridad en la nube: su responsabilidad se determina según el servicio de AWSque utilice. También es responsable de otros factores, incluida la confidencialidad de los datos, los requisitos de la empresa y la legislación y la normativa aplicables.

Esta documentación lo ayuda a comprender cómo se aplica el modelo de responsabilidad compartida cuando se utiliza CloudWatch Network Monitor. En los siguientes temas, se le mostrará cómo configurar CloudWatch Network Monitor para satisfacer los objetivos de seguridad y conformidad. También obtendrá información sobre cómo utilizar otros servicios de AWS que lo ayudarán a monitorear y proteger los recursos de CloudWatch Network Monitor.

Temas

Protección de datos en Amazon CloudWatch Network Monitor

El Modelo de responsabilidad compartida de AWS se aplica a la protección de datos en Amazon CloudWatch Network Monitor. Como se describe en este modelo, AWS es responsable de proteger la infraestructura global que ejecuta toda la Nube de AWS. Usted es responsable de mantener el control sobre el contenido alojado en esta infraestructura. Usted también es responsable de las tareas de administración y configuración de seguridad para los Servicios de AWS que utiliza. Para obtener más información sobre la privacidad de los datos, consulte las Preguntas frecuentes sobre la privacidad de datos. Para obtener información sobre la protección de datos en Europa, consulte la publicación de blog sobre el Modelo de responsabilidad compartida de AWS y GDPR en el Blog de seguridad de AWS.

Con fines de protección de datos, recomendamos proteger las credenciales de la Cuenta de AWS y configurar cuentas de usuario individuales con AWS IAM Identity Center o AWS Identity and Access Management (IAM). De esta manera, solo se otorgan a cada usuario los permisos necesarios para cumplir sus obligaciones laborales. También recomendamos proteger sus datos de la siguiente manera:

- Utilice la autenticación multifactor (MFA) en cada cuenta.
- Utilice SSL/TLS para comunicarse con los recursos de AWS. Se recomienda el uso de TLS 1.2 y recomendamos TLS 1.3.
- Configure los registros de API y de actividad de los usuarios con AWS CloudTrail.
- Utilice las soluciones de cifrado de AWS, junto con todos los controles de seguridad predeterminados dentro de los servicios de Servicios de AWS.
- Utilice servicios de seguridad administrados avanzados, como Amazon Macie, que lo ayuden a detectar y proteger los datos confidenciales almacenados en Amazon S3.
- Si necesita módulos criptográficos validados FIPS 140-2 al acceder a AWS a través de una interfaz de línea de comandos o una API, utilice un punto de conexión de FIPS. Para obtener más información sobre los puntos de conexión de FIPS disponibles, consulte Estándar de procesamiento de la información federal (FIPS) 140-2.

Se recomienda encarecidamente no introducir nunca información confidencial o sensible, como, por ejemplo, direcciones de correo electrónico de clientes, en etiquetas o campos de formato libre, tales como el campo Nombre. Esto incluye cuando trabaja con CloudWatch Network Monitor u otros Servicios de AWS a través de la consola, la API, la AWS CLI o los SDK de AWS. Cualquier dato que ingrese en etiquetas o campos de formato libre utilizados para nombres se puede emplear para los registros de facturación o diagnóstico. Si proporciona una URL a un servidor externo, recomendamos encarecidamente que no incluya información de credenciales en la URL a fin de validar la solicitud para ese servidor.

Seguridad de la infraestructura en Amazon CloudWatch Network Monitor

Al tratarse de un servicio administrado, Amazon CloudWatch Network Monitor está protegido por los procedimientos de seguridad de red globales de AWS, que se describen en el informe oficial Amazon Web Services: Información general acerca de los procesos de seguridad.

Puede utilizar llamadas a la API de AWS publicadas para obtener acceso a CloudWatch Network Monitor a través de la red. Los clientes deben ser compatibles con la seguridad de la capa de transporte (TLS) 1.0 o una versión posterior. Recomendamos TLS 1.2 o una versión posterior. Los clientes también deben ser compatibles con conjuntos de cifrado con confidencialidad directa total (PFS) tales como DHE (Ephemeral Diffie-Hellman) o ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). La mayoría de los sistemas modernos como Java 7 y posteriores son compatibles con estos modos.

Además, las solicitudes deben estar firmadas mediante un ID de clave de acceso y una clave de acceso secreta que esté asociada a una entidad de seguridad de IAM principal. También puede utilizar AWS Security Token Service (AWS STS) para generar credenciales de seguridad temporales para firmar solicitudes.

Identity and Access Management para Amazon CloudWatch Network Monitor

AWS Identity and Access Management (IAM) es un servicio de AWS que ayuda a los administradores a controlar de forma segura el acceso a los recursos de AWS. Los administradores de IAM controlan quién está autenticado (ha iniciado sesión) y autorizado (tiene permisos) para utilizar recursos de CloudWatch Network Monitor. IAM es un servicio de AWS que puede utilizar sin cargo adicional. Puede utilizar las características de IAM para permitir que otros usuarios, servicios y aplicaciones usen sus recursos de AWS total o parcialmente, sin necesidad de compartir sus credenciales de seguridad.

De forma predeterminada, los usuarios de IAM no tienen permiso para crear, consultar ni modificar recursos de AWS. Para permitir que un usuario de IAM acceda a los recursos, por ejemplo, una red global, y lleve a cabo tareas, debe:

- Creación de una política de IAM que conceda permiso al usuario para utilizar los recursos específicos y las acciones de la API que necesita
- Asociación de la política al usuario de IAM o al grupo al que pertenece el usuario

Cuando se asocia una política a un usuario o un grupo de usuarios, esta les concede o deniega permisos para realizar las tareas especificadas en los recursos indicados.

Claves de condición

El elemento `Condition` (o bloque Condition) permite especificar condiciones en las que entra en vigor una instrucción. El elemento Condition es opcional. Puede crear expresiones condicionales que utilizan operadores de condición, tales como igual o menor que, para que coincida la condición de la política con valores de la solicitud. Para obtener más información, consulte Elementos de la política de JSON de IAM: Condition Operators en la Guía del usuario de Identity and Access Management de AWS.

Si especifica varios elementos de `Condition` en una instrucción o varias claves en un único elemento de `Condition`, AWS las evalúa mediante una operación `AND` lógica. Si especifica varios valores para una única clave de condición, AWS evalúa la condición con una operación lógica `OR`. Se deben cumplir todas las condiciones antes de que se concedan los permisos de la instrucción.

También puede utilizar variables de marcador de posición al especificar condiciones. Por ejemplo, puede conceder un permiso de usuario de IAM para acceder a un recurso solo si está etiquetado con su nombre de usuario de IAM.

Puede adjuntar etiquetas a los recursos de CloudWatch Network Monitor o transferirlas en una solicitud a Cloud WAN. Para controlar el acceso en función de etiquetas, debe proporcionar información de las etiquetas en el elemento de condición de una política utilizando las claves de condición `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` o `aws:TagKeys`. Consulte Elementos de la política JSON de IAM: condición en la Guía del usuario de Identity and Access Management AWS para obtener más información.

Para ver todas las claves de condición globales de AWS, consulte Claves de contexto de condición globales AWS en la Guía del usuario de Identity and Access Management AWS.

Etiquetas en los recursos principales de la red

Una etiqueta es una etiqueta de metadatos que usted o AWS asignan a un recurso de AWS. Cada etiqueta consta de una clave y un valor. En el caso de etiquetas que usted asigna, debe definir la clave y el valor. Por ejemplo, puede definir la clave como `purpose` y el valor como `test` para un recurso. Las etiquetas le ayudan a hacer lo siguiente:

- Identificar y organizar sus recursos de AWS. Muchos servicios de AWS admiten el etiquetado, por lo que puede asignar la misma etiqueta a los recursos de diferentes servicios para indicar que los recursos están relacionados.
- Controle el acceso a los recursos de AWS. Para más información, consulte Control del acceso a recursos de AWS con etiquetas en la Guía del usuario de Identity and Access Management AWS.

Funcionamiento de Amazon CloudWatch Network Monitor con IAM

Antes de utilizar IAM para administrar el acceso a CloudWatch Network monitor, obtenga información sobre qué características de IAM se pueden utilizar con CloudWatch Network Monitor.

Características de IAM que puede utilizar con Amazon CloudWatch Network Monitor

Característica de IAM	Compatibilidad con CloudWatch Network Monitor
Políticas basadas en identidades	Sí
Políticas basadas en recursos	No
Acciones de políticas	Sí
Recursos de políticas	Sí
Claves de condición de política	Sí
ACL	No
ABAC (etiquetas en políticas)	Parcial
Credenciales temporales	Sí
Permisos de entidades principales	Sí

Característica de IAM	Compatibilidad con CloudWatch Network Monitor
Roles de servicio	No
Roles vinculados al servicio	Sí

Para obtener una perspectiva general sobre cómo funcionan CloudWatch Network Monitor y otros servicios de AWS con la mayoría de características de IAM, consulte Servicios de AWS que funcionan con IAM en la Guía del usuario de IAM.

Políticas de Amazon CloudWatch Network Monitor basadas en identidades

Compatibilidad con las políticas basadas en identidad	Sí

Las políticas basadas en identidad son documentos de políticas de permisos JSON que puede asociar a una identidad, como un usuario de IAM, un grupo de usuarios o un rol. Estas políticas controlan qué acciones pueden realizar los usuarios y los roles, en qué recursos y en qué condiciones. Para obtener más información sobre cómo crear una política basada en identidad, consulte Creación de políticas de IAM en la Guía del usuario de IAM.

Con las políticas basadas en identidades de IAM, puede especificar las acciones y los recursos permitidos o denegados, así como las condiciones en las que se permiten o deniegan las acciones. No es posible especificar la entidad principal en una política basada en identidad porque se aplica al usuario o rol al que está adjunto. Para más información sobre los elementos que puede utilizar en una política de JSON, consulte Referencia de los elementos de las políticas de JSON de IAM en la Guía del usuario de IAM.

Ejemplos de políticas basadas en identidades para CloudWatch Network Monitor

Para ver ejemplos de políticas basadas en identidad de CloudWatch Network Monitor, consulte Ejemplos de políticas basadas en identidades para Amazon CloudWatch.

Políticas basadas en recursos dentro de CloudWatch Network Monitor

Compatibilidad con las políticas basadas en recursos	No

Las políticas basadas en recursos son documentos de política JSON que se asocian a un recurso. Ejemplos de políticas basadas en recursos son las políticas de confianza de roles de IAM y las políticas de bucket de Amazon S3. En los servicios que admiten políticas basadas en recursos, los administradores de servicios pueden utilizarlos para controlar el acceso a un recurso específico. Para el recurso al que se asocia la política, la política define qué acciones puede realizar una entidad principal especificada en ese recurso y en qué condiciones. Debe especificar una entidad principal en una política en función de recursos. Las entidades principales pueden incluir cuentas, usuarios, roles, usuarios federados o Servicios de AWS.

Para habilitar el acceso entre cuentas, puede especificar toda una cuenta o entidades de IAM de otra cuenta como la entidad principal de una política en función de recursos. Añadir a una política en función de recursos una entidad principal entre cuentas es solo una parte del establecimiento de una relación de confianza. Cuando la entidad principal y el recurso se encuentran en Cuentas de AWS diferentes, un administrador de IAM de la cuenta de confianza también debe conceder a la entidad principal (usuario o rol) permiso para acceder al recurso. Para conceder el permiso, adjunte la entidad a una política basada en identidad. Sin embargo, si la política en función de recursos concede el acceso a una entidad principal de la misma cuenta, no es necesaria una política basada en identidad adicional. Para más información, consulte Cómo los roles de IAM difieren de las políticas basadas en recursos en la Guía del usuario de IAM.

Acciones de política para CloudWatch Network Monitor

Admite acciones de política	Sí

Los administradores pueden utilizar las políticas JSON de AWS para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones.

El elemento `Action` de una política JSON describe las acciones que puede utilizar para conceder o denegar el acceso en una política. Las acciones de la política generalmente tienen el mismo nombre que la operación de API de AWS asociada. Hay algunas excepciones, como acciones de solo permiso que no tienen una operación de API coincidente. También hay algunas operaciones

que requieren varias acciones en una política. Estas acciones adicionales se denominan acciones dependientes.

Incluya acciones en una política para conceder permisos y así llevar a cabo la operación asociada.

Para ver una lista de las acciones de CloudWatch Network Monitor, consulte Acciones definidas por Amazon CloudWatch Network Monitor en la Referencia de autorizaciones de servicio.

Las acciones de políticas de CloudWatch Network Monitor utilizan el siguiente prefijo antes de la acción:

```
networkmonitor
```

Para especificar varias acciones en una única instrucción, sepárelas con comas.

```
"Action": [
      "networkmonitor:action1",
      "networkmonitor:action2"
         ]
```

Para ver ejemplos de políticas basadas en identidad de CloudWatch Network Monitor, consulte Ejemplos de políticas basadas en identidades para Amazon CloudWatch.

Recursos de política para CloudWatch Network Monitor

Admite recursos de políticas	Sí

Los administradores pueden utilizar las políticas JSON de AWS para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones.

El elemento `Resource` de la política JSON especifica el objeto u objetos a los que se aplica la acción. Las instrucciones deben contener un elemento `Resource` o `NotResource`. Como práctica recomendada, especifique un recurso utilizando el Nombre de recurso de Amazon (ARN). Puede hacerlo para acciones que admitan un tipo de recurso específico, conocido como permisos de nivel de recurso.

Para las acciones que no admiten permisos de nivel de recurso, como las operaciones de descripción, utilice un carácter comodín (*) para indicar que la instrucción se aplica a todos los recursos.

```
"Resource": "*"
```

Para ver una lista de tipos de recursos de CloudWatch Network Monitor y los ARN, consulte Recursos definidos por Amazon CloudWatch Network Monitor en la Referencia de autorizaciones de servicio. Para obtener información sobre las acciones con las que puede especificar el ARN de cada recurso, consulte Acciones definidas por Amazon CloudWatch Network Monitor.

Política de condición para CloudWatch Network Monitor

Admite claves de condición de políticas específicas del servicio	Sí

Los administradores pueden utilizar las políticas JSON de AWS para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones.

El elemento `Condition` (o bloque de `Condition`) permite especificar condiciones en las que entra en vigor una instrucción. El elemento `Condition` es opcional. Puede crear expresiones condicionales que utilicen operadores de condición, tales como igual o menor que, para que la condición de la política coincida con los valores de la solicitud.

Si especifica varios elementos de `Condition` en una instrucción o varias claves en un único elemento de `Condition`, AWS las evalúa mediante una operación `AND` lógica. Si especifica varios valores para una única clave de condición, AWS evalúa la condición con una operación lógica `OR`. Se deben cumplir todas las condiciones antes de que se concedan los permisos de la instrucción.

También puede utilizar variables de marcador de posición al especificar condiciones. Por ejemplo, puede conceder un permiso de usuario de IAM para acceder a un recurso solo si está etiquetado con su nombre de usuario de IAM. Para más información, consulte Elementos de la política de IAM: variables y etiquetas en la Guía del usuario de IAM.

AWS admite claves de condición globales y claves de condición específicas del servicio. Para ver todas las claves de condición globales de AWS, consulte Claves de contexto de condición globales de AWSen la Guía del usuario de IAM.

Para ver una lista de las claves de condición de CloudWatch Network Monitor, consulte Claves de condición para Amazon CloudWatch Network Monitor en la Referencia de autorizaciones de servicio. Para obtener más información acerca de las acciones y los recursos con los que puede utilizar una clave de condición, consulte Acciones definidas por Amazon CloudWatch Network Monitor.

ACL en CloudWatch Network Monitor

Admite las ACL	No

Las listas de control de acceso (ACL) controlan qué entidades principales (miembros de cuentas, usuarios o roles) tienen permisos para acceder a un recurso. Las ACL son similares a las políticas basadas en recursos, aunque no utilizan el formato de documento de políticas JSON.

ABAC con CloudWatch Network Monitor

Admite ABAC (etiquetas en las políticas)	Parcial

El control de acceso basado en atributos (ABAC) es una estrategia de autorización que define permisos en función de atributos. En AWS, estos atributos se denominan etiquetas. Puede adjuntar etiquetas a entidades de IAM (usuarios o roles) y a muchos recursos de AWS. El etiquetado de entidades y recursos es el primer paso de ABAC. A continuación, designa las políticas de ABAC para permitir operaciones cuando la etiqueta de la entidad principal coincida con la etiqueta del recurso al que se intenta acceder.

ABAC es útil en entornos que crecen con rapidez y ayuda en situaciones en las que la administración de las políticas resulta engorrosa.

Para controlar el acceso en función de etiquetas, debe proporcionar información de las etiquetas en el elemento de condición de una política utilizando las claves de condición `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` o `aws:TagKeys`.

Si un servicio admite las tres claves de condición para cada tipo de recurso, el valor es Sí para el servicio. Si un servicio admite las tres claves de condición solo para algunos tipos de recursos, el valor es Parcial.

Para obtener más información sobre ABAC, consulte ¿Qué es ABAC? en la Guía del usuario de IAM. Para ver un tutorial con los pasos para configurar ABAC, consulte Uso del control de acceso basado en atributos (ABAC) en la Guía del usuario de IAM.

Uso de credenciales temporales con CloudWatch Network Monitor

Compatible con el uso de credenciales temporales	Sí

Algunos Servicios de AWS no funcionan cuando inicia sesión con credenciales temporales. Para obtener información adicional, incluida la información sobre qué Servicios de AWS funcionan con credenciales temporales, consulte Servicios de AWS que funcionan con IAM en la Guía del usuario de IAM.

Utiliza credenciales temporales si inicia sesión en la AWS Management Console con cualquier método, excepto un nombre de usuario y una contraseña. Por ejemplo, cuando accede a AWS utilizando el enlace de inicio de sesión único (SSO) de la empresa, ese proceso crea automáticamente credenciales temporales. También crea automáticamente credenciales temporales cuando inicia sesión en la consola como usuario y luego cambia de rol. Para más información sobre el cambio de roles, consulte Cambio a un rol (consola) en la Guía del usuario de IAM.

Puede crear credenciales temporales de forma manual mediante la AWS CLI o la API de AWS. A continuación, puede usar esas credenciales temporales para acceder a AWS. AWS recomienda generar credenciales temporales de forma dinámica en lugar de usar claves de acceso a largo plazo. Para más información, consulte Credenciales de seguridad temporales en IAM.

Permisos de entidades principales entre servicios para CloudWatch Network Monitor

Admite Forward access sessions (FAS)	Sí

Cuando utiliza un usuario o un rol de IAM para llevar a cabo acciones en AWS, se lo considera una entidad principal. Cuando utiliza algunos servicios, es posible que realice una acción que desencadene otra acción en un servicio diferente. FAS utiliza los permisos de la entidad principal para llamar a un Servicio de AWS, combinados con el Servicio de AWS solicitante para realizar solicitudes a servicios posteriores. Las solicitudes de FAS solo se realizan cuando un servicio recibe una solicitud que requiere interacciones con otros Servicios de AWS o recursos para completarse. En este caso, debe tener permisos para realizar ambas acciones. Para obtener información sobre las políticas a la hora de realizar solicitudes de FAS, consulte Reenviar sesiones de acceso.

Roles de servicio para CloudWatch Network Monitor

Compatible con roles de servicio	No

Un rol de servicio es un rol de IAM que asume un servicio para realizar acciones en su nombre. Un administrador de IAM puede crear, modificar y eliminar un rol de servicio desde IAM. Para obtener más información, consulte Creación de un rol para delegar permisos a un Servicio de AWS en la Guía del usuario de IAM.

> **Warning**
>
> Es posible que cambiar los permisos de un rol de servicio interrumpa la funcionalidad de CloudWatch Network Monitor. Edite los roles de servicio solo cuando CloudWatch Network Monitor proporcione orientación para hacerlo.

Uso de un rol vinculado a servicio para CloudWatch Network Monitor

Compatible con roles vinculados al servicio	Sí

Un rol vinculado al servicio es un tipo de rol de servicio que está vinculado a un Servicio de AWS. El servicio puede asumir el rol para realizar una acción en su nombre. Los roles vinculados a servicios aparecen en la Cuenta de AWS y son propiedad del servicio. Un administrador de IAM puede ver, pero no editar, los permisos de los roles vinculados a servicios.

Para más información sobre cómo crear o administrar roles vinculados a servicios, consulte Servicios de AWS que funcionan con IAM. Busque un servicio en la tabla que incluya `Yes` en la columna Rol vinculado a un servicio. Seleccione el vínculo Sí para ver la documentación acerca del rol vinculado a servicios para ese servicio.

Ejemplos de políticas basadas en identidades para CloudWatch Network Monitor

De forma predeterminada, los usuarios y roles no tienen permiso para crear o modificar recursos de CloudWatch Network Monitor. Tampoco pueden realizar tareas mediante la AWS Management Console, la AWS Command Line Interface (AWS CLI) o la API de AWS. Un administrador de IAM puede crear políticas de IAM para conceder permisos a los usuarios para realizar acciones en los

recursos que necesitan. A continuación, el administrador puede añadir las políticas de IAM a roles y los usuarios pueden asumirlos.

Para obtener información acerca de cómo crear una política basada en identidades de IAM mediante el uso de estos documentos de políticas JSON de ejemplo, consulte Creación de políticas de IAM en la Guía del usuario de IAM.

Para obtener más información sobre las acciones y los tipos de recursos definidos por CloudWatch Network Monitor, incluido el formato de los ARN para cada tipo de recurso, consulte Acciones, recursos y claves de condición de Amazon CloudWatch Network Monitor en la Referencia de autorizaciones de servicio.

Temas

Prácticas recomendadas sobre las políticas

Las políticas basadas en identidades determinan si alguien puede crear, eliminar o acceder a los recursos de CloudWatch Network Monitor de la cuenta. Estas acciones pueden generar costos adicionales para su Cuenta de AWS. Siga estas directrices y recomendaciones al crear o editar políticas basadas en identidades:

- Comience con las políticas administradas por AWSy continúe con los permisos de privilegio mínimo: a fin de comenzar a conceder permisos a los usuarios y las cargas de trabajo, utilice las políticas administradas por AWS, que conceden permisos para muchos casos de uso comunes. Están disponibles en su Cuenta de AWS. Se recomienda definir políticas administradas por el cliente de AWS específicas para sus casos de uso a fin de reducir aún más los permisos. Con el fin de obtener más información, consulte las políticas administradas por AWS o las políticas administradas por AWS para funciones de trabajo en la Guía de usuario de IAM.
- Aplique permisos de privilegio mínimo: cuando establezca permisos con políticas de IAM, conceda solo los permisos necesarios para realizar una tarea. Para ello, debe definir las acciones que se pueden llevar a cabo en determinados recursos en condiciones específicas, también conocidos como permisos de privilegios mínimos. Con el fin de obtener más información sobre el uso de IAM para aplicar permisos, consulte Políticas y permisos en IAM en la Guía del usuario de IAM.

- Utilice condiciones en las políticas de IAM para restringir aún más el acceso: puede agregar una condición a sus políticas para limitar el acceso a las acciones y los recursos. Por ejemplo, puede escribir una condición de políticas para especificar que todas las solicitudes deben enviarse utilizando SSL. También puede usar condiciones para conceder acceso a acciones de servicios si se emplean a través de un Servicio de AWS determinado como, por ejemplo, AWS CloudFormation. Para obtener más información, consulte Elementos de la política de JSON de IAM: Condición en la Guía del usuario de IAM.
- Utilice el analizador de acceso de IAM para validar las políticas de IAM con el fin de garantizar la seguridad y funcionalidad de los permisos: el analizador de acceso de IAM valida políticas nuevas y existentes para que respeten el lenguaje (JSON) de las políticas de IAM y las prácticas recomendadas de IAM. El analizador de acceso de IAM proporciona más de 100 verificaciones de políticas y recomendaciones procesables para ayudar a crear políticas seguras y funcionales. Para más información, consulte Política de validación de Analizador de acceso de IAM en la Guía de usuario de IAM.
- Solicite la autenticación multifactor (MFA): si se encuentra en una situación en la que necesita usuarios raíz o de IAM en su Cuenta de AWS, active la MFA para mayor seguridad. Para solicitar la MFA cuando se invocan las operaciones de la API, agregue las condiciones de la MFA a sus políticas. Para más información, consulte Configuración del acceso a una API protegido por MFA en la Guía de usuario de IAM.

Para obtener más información sobre las prácticas recomendadas de IAM, consulte las Prácticas recomendadas de seguridad en IAM en la Guía del usuario de IAM.

Uso de la consola de CloudWatch Network Monitor

Para acceder a la consola de Amazon CloudWatch Network Monitor, debe tener un conjunto mínimo de permisos. Estos permisos deben permitirle registrar y consultar los detalles acerca de los recursos de CloudWatch Network Monitor en la Cuenta de AWS. Si crea una política basada en identidades que sea más restrictiva que el mínimo de permisos necesarios, la consola no funcionará del modo esperado para las entidades (usuarios o roles) que tengan esa política.

No es necesario que conceda permisos mínimos para la consola a los usuarios que solo realizan llamadas a la AWS CLI o a la API de AWS. En su lugar, permite acceso únicamente a las acciones que coincidan con la operación de API que intentan realizar.

Para asegurarse de que los usuarios y los roles puedan seguir utilizando la consola de CloudWatch Network Monitor, asocie también la `ConsoleAccess` de CloudWatch Network Monitor o la política

administrada *ReadOnly* de AWS a las entidades. Para más información, consulte Adición de permisos a un usuario en la Guía del usuario de IAM:

Cómo permitir a los usuarios consultar sus propios permisos

En este ejemplo, se muestra cómo podría crear una política que permita a los usuarios de IAM ver las políticas administradas e insertadas que se asocian a la identidad de sus usuarios. Esta política incluye permisos para llevar a cabo esta acción en la consola o mediante programación con la AWS CLI o la API de AWS.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

Resolución de problemas de identidad y acceso de CloudWatch Network Monitor

Utilice la siguiente información para diagnosticar y solucionar los problemas comunes que pueden surgir al trabajar con CloudWatch Network Monitor e IAM.

Temas

Sin autorización para realizar una acción en CloudWatch Network Monitor

Si recibe un error que indica que no tiene autorización para realizar una acción, las políticas se deben actualizar para permitirle realizar la acción.

En el siguiente ejemplo, el error se produce cuando el usuario de IAM `mateojackson` intenta utilizar la consola para consultar los detalles acerca de un recurso ficticio *`my-example-widget`*, pero no tiene los permisos ficticios `networkmonitor:`*`GetWidget`*.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 networkmonitor:GetWidget on resource: my-example-widget
```

En este caso, la política del usuario `mateojackson` debe actualizarse para permitir el acceso al recurso *`my-example-widget`* mediante la acción `networkmonitor:`*`GetWidget`*.

Si necesita ayuda, póngase en contacto con su administrador de AWS. El administrador es la persona que le proporcionó las credenciales de inicio de sesión.

No tengo autorización para realizar la operación iam:PassRole

Si recibe un error que indica que no tiene autorización para realizar la acción `iam:PassRole`, las políticas se deben actualizar a fin de permitirle pasar un rol a CloudWatch Network Monitor.

Algunos servicios de Servicios de AWS le permiten transferir un rol existente a dicho servicio en lugar de crear un nuevo rol de servicio o uno vinculado al servicio. Para ello, debe tener permisos para transferir el rol al servicio.

En el siguiente ejemplo, el error se produce cuando un usuario de IAM denominado `marymajor` intenta utilizar la consola para realizar una acción en CloudWatch Network Monitor. Sin embargo, la

acción requiere que el servicio cuente con permisos que otorguen un rol de servicio. Mary no tiene permisos para transferir el rol al servicio.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

En este caso, las políticas de Mary se deben actualizar para permitirle realizar la acción `iam:PassRole`.

Si necesita ayuda, póngase en contacto con su administrador de AWS. El administrador es la persona que le proporcionó las credenciales de inicio de sesión.

Deseo permitir que personas ajenas a mi Cuenta de AWS puedan acceder a mis recursos de CloudWatch Network Monitor.

Puede crear un rol que los usuarios de otras cuentas o las personas externas a la organización puedan utilizar para acceder a sus recursos. Puede especificar una persona de confianza para que asuma el rol. En el caso de los servicios que admitan las políticas basadas en recursos o las listas de control de acceso (ACL), puede utilizar dichas políticas para conceder a las personas acceso a sus recursos.

Para más información, consulte lo siguiente:

- Para obtener información acerca de si CloudWatch Network Monitor admite estas características, consulte Cómo funciona Amazon CloudWatch con IAM.
- Para obtener información acerca de cómo proporcionar acceso a los recursos de las Cuentas de AWS de su propiedad, consulte Proporcionar acceso a un usuario de IAM a otra cuenta de Cuenta de AWS propia en la Guía del usuario de IAM.
- Para obtener información acerca de cómo proporcionar acceso a tus recursos a Cuentas de AWS de terceros, consulte Proporcionar acceso a Cuentas de AWS que son propiedad de terceros en la Guía del usuario de IAM.
- Para obtener información sobre cómo proporcionar acceso mediante una federación de identidades, consulte Proporcionar acceso a usuarios autenticados externamente (identidad federada) en la Guía del usuario de IAM.
- Para obtener información sobre la diferencia entre los roles y las políticas basadas en recursos para el acceso entre cuentas, consulte Cómo los roles de IAM difieren de las políticas basadas en recursos en la Guía del usuario de IAM.

Políticas administradas de AWS para CloudWatch Network Monitor

Para agregar permisos a usuarios, grupos y roles, es más fácil utilizar las políticas administradas de AWS que escribirlas uno mismo. Se necesita tiempo y experiencia para crear políticas administradas por el cliente de IAM que le brinden a su equipo solo los permisos necesarios. Para comenzar a hacerlo con rapidez, puede utilizar nuestras políticas administradas de AWS. Estas políticas cubren casos de uso comunes y están disponibles en su cuenta de AWS. Para obtener más información sobre las políticas administradas por AWS, consulte Políticas administradas por AWS en la Guía del usuario de IAM.

Los servicios de AWS mantienen y actualizan las políticas administradas por AWS. No puede cambiar los permisos en las políticas administradas de AWS. En ocasiones, los servicios agregan permisos adicionales a una política administrada por AWS para admitir características nuevas. Este tipo de actualización afecta a todas las identidades (usuarios, grupos y roles) donde se asocia la política. Es más probable que los servicios actualicen una política administrada por AWScuando se lanza una nueva característica o cuando se ponen a disposición nuevas operaciones. Los servicios no quitan los permisos de una política administrada de AWS, por lo tanto, las actualizaciones de las políticas no deteriorarán los permisos existentes.

Además, AWS admite políticas administradas para funciones de trabajo que abarcan varios servicios. Por ejemplo, la política `ReadOnlyAccess` administrada por AWS proporciona acceso de solo lectura a todos los servicios y recursos de AWS. Cuando un servicio lanza una nueva característica, AWS agrega permisos de solo lectura para las operaciones y los recursos nuevos. Para obtener una lista y descripciones de las políticas de funciones de trabajo, consulte Políticas administradas de AWS para funciones de trabajo en la Guía del usuario de IAM.

Política administrada de AWS: CloudWatchNetworkMonitorServiceRolePolicy

La `CloudWatchNetworkMonitorServiceRolePolicy` se asocia a un rol vinculado a servicios que permite al servicio realizar acciones en su nombre y acceder a los recursos asociados a CloudWatch Network Monitor. No puede adjuntar esta política a las identidades de IAM. Para obtener más información, consulte the section called “Roles vinculados al servicio”.

Actualizaciones de CloudWatch Network Monitoring para las políticas administradas de AWS

Es posible consultar los detalles sobre las actualizaciones de las políticas administradas de AWS para CloudWatch Network Monitoring debido a que este servicio comenzó a realizar cambios en noviembre de 2023.

Cambio	Descripción	Fecha
Política de roles de servicio de CloudWatchNetworkMonitorServiceRolePolicy: nueva política.	Se ha añadido una nueva política a CloudWatch Network Monitor.	27 de noviembre de 2023
the section called "AWSServiceRoleForNetworkMonitor". Nuevo rol.	Se ha añadido un nuevo rol a CloudWatch Network Monitor.	27 de noviembre de 2023

Permisos de IAM para CloudWatch Network Monitor

Para usar Amazon CloudWatch Network Monitor, debe contar con los permisos correctos.

Para obtener más información sobre la seguridad en Amazon CloudWatch, consulte Identity and Access Management para Amazon CloudWatch.

Permisos necesarios para visualizar un monitor

Para visualizar un monitor de Amazon CloudWatch Network Monitor en la AWS Management Console, debe haber iniciado sesión como rol que tenga los siguientes permisos:

```
{
"Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "cloudwatch:GetMetricData",
                "networkmonitor:Get*",
                "networkmonitor:List*"
                ],
            "Resource": "*"
        }
    ]
}
```

Permisos necesarios para crear un monitor

Para crear un monitor en Amazon CloudWatch Network Monitor, los usuarios deben tener permiso para crear un rol vinculado a un servicio asociado a Network Monitor. Para obtener más información sobre el rol vinculado a servicios, consulte Uso de un rol vinculado a servicio para CloudWatch Network Monitor.

Para crear un monitor de Amazon CloudWatch Network Monitor en la AWS Management Console, debe haber iniciado sesión como usuario o rol que tenga los permisos incluidos en la siguiente política.

> **Note**
>
> Si crea una política de permisos basados en identidad que sea más restrictiva, los usuarios que posean esa política no podrán crear un monitor.

```
{
"Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "networkmonitor:*"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "arn:aws:iam::*:role/aws-service-role/
networkmonitor.amazonaws.com/AWSServiceRoleForNetworkMonitor",
            "Condition": {
                "StringLike": {
                    "iam:AWSServiceName": "networkmonitor.amazonaws.com"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
```

```
                "iam:AttachRolePolicy",
                "iam:GetRole",
                "iam:PutRolePolicy"
            ],
            "Resource": "arn:aws:iam::*:role/aws-service-role/
networkmonitor.amazonaws.com/AWSServiceRoleForNetworkMonitor"
        },
        {
            "Action": [
                "ec2:CreateSecurityGroup",
                "ec2:CreateNetworkInterface",
                "ec2:CreateTags"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

Uso de un rol vinculado a servicio para CloudWatch Network Monitor

Amazon CloudWatch Network Monitor utiliza los siguientes roles vinculados a servicios para los permisos que necesita para llamar a otros servicios AWS en su nombre:

- AWSServiceRoleForNetworkMonitor

AWSServiceRoleForNetworkMonitor

CloudWatch Network Monitoring utiliza el rol vinculado al servicio denominado `AWSServiceRoleForNetworkMonitor` para actualizar y administrar los monitores de red de CloudWatch.

El rol vinculado a servicio de `AWSServiceRoleForNetworkMonitor` confía en el siguiente servicio para asumir el rol:

- `networkmonitor.amazonaws.com`

La `CloudWatchNetworkMonitorServiceRolePolicy` se adjunta a la función vinculada al servicio y otorga acceso al servicio para acceder a los recursos de VPC y EC2 de la cuenta, así como para administrar los monitores de red que se crearon.

Agrupaciones de permisos

La política se agrupa en los siguientes conjuntos de permisos:

- **`cloudwatch`**: esto permite a la entidad principal del servicio publicar las métricas de supervisión de la red en los recursos de CloudWatch.
- **`ec2`**: esto permite a la entidad principal del servicio describir las VPC y las subredes de la cuenta para crear o actualizar monitores y sondas. Esto también permite a la entidad principal del servicio crear, modificar y eliminar grupos de seguridad, interfaces de red y sus permisos asociados para configurar el monitor o la sonda a fin de enviar el tráfico de supervisión a sus puntos de conexión.

Para obtener más información acerca de la política, consulte the section called “Políticas administradas de AWS”.

A continuación se muestra la `CloudWatchNetworkMonitorServiceRolePolicy`:

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Sid": "PublishCw",
   "Effect": "Allow",
   "Action": "cloudwatch:PutMetricData",
   "Resource": "*",
   "Condition": {
    "StringEquals": {
     "cloudwatch:namespace": "AWS/NetworkMonitor"
    }
   }
  },
  {
   "Sid": "DescribeAny",
   "Effect": "Allow",
   "Action": [
    "ec2:DescribeNetworkInterfaces",
    "ec2:DescribeNetworkInterfaceAttribute",
    "ec2:DescribeVpcs",
    "ec2:DescribeNetworkInterfacePermissions",
    "ec2:DescribeSubnets",
    "ec2:DescribeSecurityGroups"
   ],
   "Resource": "*"
```

```
  },
  {
   "Sid": "DeleteModifyEc2Resources",
   "Effect": "Allow",
   "Action": [
    "ec2:AuthorizeSecurityGroupEgress",
    "ec2:CreateNetworkInterfacePermission",
    "ec2:DeleteNetworkInterfacePermission",
    "ec2:RevokeSecurityGroupEgress",
    "ec2:ModifyNetworkInterfaceAttribute",
    "ec2:DeleteNetworkInterface",
    "ec2:DeleteSecurityGroup"
   ],
   "Resource": [
    "arn:aws:ec2:*:*:network-interface/*",
    "arn:aws:ec2:*:*:security-group/*"
   ],
   "Condition": {
    "StringEquals": {
     "aws:ResourceTag/ManagedByCloudWatchNetworkMonitor": "true"
    }
   }
  }
 ]
}
```

Creación del rol vinculado a servicios

`AWSServiceRoleForNetworkMonitor`

No necesita crear manualmente un rol `AWSServiceRoleForNetworkMonitor`.

- CloudWatch Network Monitor crea el rol `AWSServiceRoleForNetworkMonitor` al crear el primer monitor de red. Esta función se aplicará a todos los monitores posteriores que cree.

Para crear un rol vinculado al servicio en su nombre, debe contar con los permisos necesarios. Para obtener más información, consulte Permisos de roles vinculados a servicios en la Guía del usuario de IAM.

Editar el rol vinculado a servicios

Puede modificar la descripción de `AWSServiceRoleForNetworkMonitor` mediante IAM. Para obtener más información, consulte Editar un rol vinculado a servicios en la Guía del usuario de IAM.

Eliminar el rol vinculado a un servicio

Si ya no tiene que utilizar CloudWatch Network Monitor, le recomendamos que elimine el rol `AWSServiceRoleForNetworkMonitor`.

Solo puede eliminar estos roles vinculados a servicios después de eliminar la red de monitoreo. Para obtener información sobre cómo eliminar el monitor de red, consulte Eliminar un monitor de red.

Puede utilizar la consola, la CLI o la API de IAM para eliminar los roles vinculados a servicios. Para obtener más información, consulte Eliminación de un rol vinculado a servicios en la Guía del usuario de IAM.

Después de eliminar `AWSServiceRoleForNetworkMonitor` , CloudWatch Network Monitor creará de nuevo el rol si crea un nuevo monitor.

Regiones admitidas para roles vinculados al servicio de CloudWatch Network Monitor

CloudWatch Network Monitor admite el uso de roles vinculados al servicio Regiones de AWSdonde el servicio está disponible. Para obtener más información, consulte los puntos de conexión de AWS en la Referencia general de AWS.

Eliminar el rol vinculado a un servicio

Si ya no tiene que utilizar CloudWatch Network Monitor, le recomendamos que elimine el rol `AWSServiceRoleForNetworkMonitor`.

Solo puede eliminar estos roles vinculados a servicios después de eliminar la red de monitoreo. Para obtener información sobre cómo eliminar el monitor de red, consulte Eliminar un monitor de red.

Puede utilizar la consola, la CLI o la API de IAM para eliminar los roles vinculados a servicios. Para obtener más información, consulte Eliminación de un rol vinculado a servicios en la Guía del usuario de IAM.

Después de eliminar `AWSServiceRoleForNetworkMonitor` , CloudWatch Network Monitor creará de nuevo el rol si crea un nuevo monitor.

Precios

Con Amazon CloudWatch Network Monitor, no hay costes iniciales ni compromisos a largo plazo. Los precios de Network Monitor incluyen los dos componentes siguientes:

- una tarifa por hora por recurso monitoreado, y
- tarifas de métricas de CloudWatch.

Al crear un monitor de red, se le asocian los recursos que se van a supervisar. En el caso del Network Monitor, se tratará de subredes en Amazon Virtual Private Cloud (VPC). Cada recurso monitoreado le permite crear hasta cuatro sondas desde cada subred de sus VPC a cuatro destinos. Para ayudarlo a controlar la factura, puede ajustar la cobertura de subred y la cobertura IP en las instalaciones, reduciendo la cantidad de recursos monitoreados.

Para obtener más información acerca de los precios, consulte la página Precios de Amazon CloudWatch.

Monitoreo de infraestructuras

Los temas de esta sección explican las características de CloudWatch que pueden ayudar a obtener visibilidad operativa de su recursos de AWS.

Temas

Información de contenedores

Utilice Información de contenedores de CloudWatch para recopilar, agregar y resumir métricas y registros de las aplicaciones en contenedores y de los microservicios. Información de contenedores está disponible para las plataformas Amazon Elastic Container Service (Amazon ECS), Amazon Elastic Kubernetes Service (Amazon EKS) y Kubernetes en Amazon EC2. Información de contenedores admite la recopilación de métricas de clústeres implementados en AWS Fargate tanto para Amazon ECS como Amazon EKS.

CloudWatch recopila automáticamente métricas de muchos recursos, como la CPU, la memoria, el disco y la red. Información de contenedores también proporciona información de diagnóstico, como, por ejemplo, errores de reinicio de contenedores, para ayudarle a aislar problemas y solucionarlos rápidamente. También puede establecer alarmas de CloudWatch en las métricas que recopila Información de contenedores.

Información de contenedores recopila datos como Eventos de registro de rendimiento con embedded metric format (formato de métricas integradas). Estos eventos de registro de rendimiento son entradas que utilizan un esquema JSON estructurado que permite capturar y almacenar datos de cardinalidad alta a escala. A partir de estos datos, CloudWatch crea métricas agregadas a nivel de clúster, nodo pod y servicio como métricas de CloudWatch. Las métricas que recopila Información de contenedores están disponibles en los paneles automáticos de CloudWatch y también se pueden ver en la sección Métricas de la consola de CloudWatch. Las métricas no están visibles hasta que las tareas del contenedor hayan estado ejecutándose durante algún tiempo.

Al implementar Información de contenedores, este crea automáticamente un grupo de registro para los eventos del registro de rendimiento. No es necesario que cree este grupo de registro.

CloudWatch no crea automáticamente todas las métricas posibles a partir de los datos del registro con el objetivo de ayudarle a administrar los costos de Información de contenedores. Sin embargo, puede visualizar métricas adicionales y niveles adicionales de granularidad con CloudWatch Logs Insights para analizar los eventos de registro de rendimiento sin procesar.

Con la versión original de Información de contenedores, las métricas recopiladas y los registros incorporados se cobran como métricas personalizadas. Con Información de contenedores, con una observabilidad mejorada para Amazon EKS, las métricas y los registros de Información de contenedores se cobran por observación en lugar de cobrarse por métrica almacenada o registro incorporado. Para obtener más información sobre los precios de CloudWatch, consulte Precios de Amazon CloudWatch.

En Amazon EKS y Kubernetes, Información de contenedores utiliza una versión en contenedores del agente de CloudWatch para detectar todos los contenedores en ejecución en un clúster. A continuación, recopila datos de rendimiento en cada nivel de la pila de rendimiento.

Información de contenedores admite el cifrado con la AWS KMS key para los registros y las métricas que recopila. Para habilitar este cifrado, debe activar manualmente el cifrado de AWS KMS para el grupo de registro que recibe datos de Información de contenedores. Esto hace que Información de contenedores cifre esos datos con la clave KMS proporcionada. Solo se admiten claves simétricas. No utilice claves KMS asimétricas para cifrar sus grupos de registro.

Para obtener más información, consulte Encrypt Log Data in CloudWatch Logs Using AWS KMS (Cifrar datos de registro en CloudWatch Logs mediante).

Información de contenedores con observabilidad mejorada para Amazon EKS

El 6 de noviembre de 2023, se lanzó una nueva versión de Información de contenedores. Esta versión admite la observabilidad mejorada de los clústeres de Amazon EKS que se ejecutan en Amazon EC2 y puede recopilar métricas más detalladas estos clústeres. Tras la instalación, recopila automáticamente registros detallados de telemetría de infraestructura y de contenedores para sus clústeres de Amazon EKS. A continuación, puede utilizar paneles seleccionados y de uso inmediato para profundizar en la telemetría de aplicaciones e infraestructuras.

Información de contenedores con observabilidad mejorada para Amazon EKS, recopila métricas detalladas del estado granular, del rendimiento y del estado hasta el nivel del contenedor, así como

métricas del plano de control. Para obtener más información acerca de las métricas y dimensiones adicionales recopiladas, consulte Métricas de Información de contenedores de Kubernetes y de Amazon EKS.

Si instaló Información de contenedores mediante el agente CloudWatch en un clúster de Amazon EKS en Amazon EC2, después del 6 de noviembre de 2023, dispondrá de Información de contenedores con observabilidad mejorada para Amazon EKS. De lo contrario, puede actualizar un clúster de Amazon EKS a esta nueva versión siguiendo las instrucciones que se indican en Actualización a Información de contenedores con observabilidad mejorada para Amazon EKS.

La Información de contenedores admite la observabilidad entre cuentas de CloudWatch. Utiliza una cuenta de supervisión para supervisar y solucionar problemas en las aplicaciones que abarcan varias cuentas de AWS dentro de una sola región. Para obtener más información, consulte Observabilidad entre cuentas de CloudWatch.

Información de contenedores con observabilidad mejorada para Amazon EKS también es compatible con los nodos de trabajo de Windows.

Fargate no admite Información de contenedores con observabilidad mejorada para Amazon EKS.

> **Note**
>
> Navegue hasta la consola de Información de contenedores para descubrir si tiene clústeres que puedan actualizarse a Información de contenedores con observabilidad mejorada para Amazon EKS. Para ello, elija Información, Información de contenedores en el panel de navegación de la consola de CloudWatch. En la consola de Información de contenedores, un anuncio indica si hay algún clúster de Amazon EKS que pueda actualizarse y enlaza con la página de actualización.

Plataformas admitidas

Información de contenedores está disponible para las plataformas Amazon Elastic Container Service, Amazon Elastic Kubernetes Service y Kubernetes en instancias de Amazon EC2.

- Para Amazon ECS, Información de contenedores recopila métricas a nivel de clúster, tarea y servicio en las instancias de Linux y Windows Server. Puede recopilar métricas en el nivel de instancia solo en instancias de Linux.

Para Amazon ECS, las métricas de red solo están disponibles para los contenedores en los modos de red `bridge` y `awsvpc`. No están disponibles para contenedores en el modo de red `host`.

- Para Amazon Elastic Kubernetes Service y las plataformas de Kubernetes en instancias de Amazon EC2, Información de contenedores solo se admite en instancias de Linux.

Imagen del contenedor del agente de CloudWatch

Amazon proporciona una imagen de contenedor de agente de CloudWatch en el registro de contenedores de Amazon Elastic. Para obtener más información, consulte cloudwatch-agent en Amazon ECR.

Regiones admitidas

Información de contenedores para Amazon ECS es compatible en las siguientes regiones:

- Este de EE. UU. (Norte de Virginia)
- Este de EE. UU. (Ohio)
- Oeste de EE. UU. (Norte de California)
- Oeste de EE. UU. (Oregón)
- África (Ciudad del Cabo)
- Asia-Pacífico (Hong Kong)
- Asia-Pacífico (Hyderabad)
- Asia-Pacífico (Yakarta)
- Asia-Pacífico (Bombay)
- Asia-Pacífico (Osaka)
- Asia-Pacífico (Seúl)
- Asia-Pacífico (Singapur)
- Asia-Pacífico (Tokio)
- Asia-Pacífico (Sídney)
- Oeste de Canadá (Calgary)
- Canadá (centro)

- Europa (Fráncfort)
- Europa (Irlanda)
- Europa (Londres)
- Europa (Milán)
- Europa (París)
- Europa (España)
- Europa (Estocolmo)
- Europa (Zúrich)
- Medio Oriente (Baréin)
- Medio Oriente (EAU)
- América del Sur (São Paulo)
- AWS GovCloud (Este de EE. UU.)
- AWS GovCloud (Oeste de EE. UU.)
- China (Pekín)
- China (Ningxia)

Regiones compatibles con Amazon EKS y Kubernetes

Información de contenedores para Amazon EKS y Kubernetes es compatible en las siguientes regiones:

- Este de EE. UU. (Norte de Virginia)
- Este de EE. UU. (Ohio)
- Oeste de EE. UU. (Norte de California)
- Oeste de EE. UU. (Oregón)
- Asia-Pacífico (Hong Kong)
- Asia-Pacífico (Bombay)
- Asia-Pacífico (Seúl)
- Asia-Pacífico (Singapur)
- Asia-Pacífico (Sídney)

- Asia-Pacífico (Tokio)
- Canadá (centro)
- China (Pekín)
- China (Ningxia)
- Europa (Fráncfort)
- Europa (Irlanda)
- Europa (Londres)
- Europa (París)
- Europa (Estocolmo)
- Medio Oriente (Baréin)
- América del Sur (São Paulo)
- AWS GovCloud (Este de EE. UU.)
- AWS GovCloud (Oeste de EE. UU.)

Configuración de Información de contenedores

El proceso de configuración de Información de contenedores es diferente para Amazon ECS, Amazon EKS y Kubernetes.

Temas

Configuración de Información de contenedores en Amazon ECS

Puede utilizar una de las siguientes opciones o ambas para habilitar Información de contenedores en los clústeres de Amazon ECS:

- Utilice la AWS Management Console o la AWS CLI para comenzar a recopilar métricas de nivel de clúster, de nivel de tarea y de nivel de servicio.
- Implemente el agente de CloudWatch como servicio daemon para comenzar a recopilar métricas a nivel de instancia en clústeres alojados en instancias de Amazon EC2.

Temas

Configuración de Información de contenedores en Amazon ECS para métricas de nivel de clúster y de nivel de servicio

Puede habilitar Información de contenedores en clústeres de Amazon ECS nuevos y existentes. Información de contenedores recopila métricas en los niveles de clúster, tarea y servicio. Puede habilitar Container Insights mediante la consola de Amazon ECS o la AWS CLI.

Si está utilizando Amazon ECS en una instancia de Amazon EC2 y desea recopilar las métricas de red y almacenamiento desde Información de contenedores, lance dicha instancia con una AMI que incluye el agente de Amazon ECS versión 1.29. Para obtener información sobre cómo se actualiza la versión del agente, consulte Actualización del agente de Amazon ECS Container

Puede utilizar AWS CLI para establecer el permiso de nivel de cuenta para habilitar Información de contenedores para cualquier clúster de Amazon ECS que se haya creado en su cuenta. Para ello, introduzca el siguiente comando.

```
aws ecs put-account-setting --name "containerInsights" --value "enabled"
```

> **Note**
>
> Si la clave AWS KMS administrada por el cliente que utiliza para sus métricas de Información de contenedores de Amazon ECS aún no está configurada para funcionar con CloudWatch, debe actualizar la política de claves para permitir los registros cifrados en los registros de CloudWatch. También debe asociar su propia clave AWS KMS al grupo de registros que se indica en `/aws/ecs/containerinsights/ClusterName/performance`. Para obtener

más información, consulte Cifrar datos de registro en los registros de CloudWatch mediante AWS Key Management Service.

Configuración de Información de contenedores en clústeres de Amazon ECS existentes

Para habilitar Información de contenedores en un clúster existente de Amazon ECS, ingrese el siguiente comando. Debe ejecutar la versión 1.16.200 o posterior de la AWS CLI para que el siguiente comando funcione.

```
aws ecs update-cluster-settings --cluster myCICluster --settings
 name=containerInsights,value=enabled
```

Configuración de Información de contenedores en clústeres nuevos de Amazon ECS

Hay dos maneras de habilitar Información de contenedores en clústeres nuevos de Amazon ECS. Puede configurar Amazon ECS de forma que todos los clústeres nuevos estén habilitados para Información de contenedores de forma predeterminada. De lo contrario, puede habilitar un nuevo clúster al crearlo.

Uso de la AWS Management Console

Puede habilitar Información de contenedores en todos los clústeres nuevos de forma predeterminada o en un clúster individual cuando lo cree.

Habilitar Información de contenedores en todos los clústeres nuevos de forma predeterminada

1. Abra la consola en https://console.aws.amazon.com/ecs/v2.
2. En la página de navegación, elija Account Settings (Configuración de cuenta).
3. Elija Actualizar.
4. Para utilizar Información de contenedores de CloudWatch de forma predeterminada para los clústeres, en Información de contenedores de CloudWatch, seleccione o desactive Información de contenedores de CloudWatch.
5. Elija Guardar cambios.

Si no ha usado el procedimiento anterior para habilitar Información de contenedores en todos los nuevos clústeres de forma predeterminada, puede utilizar los siguientes pasos para crear un clúster con Información de contenedores habilitado.

Para crear un clúster con Información de contenedores habilitado

1. Abra la consola en https://console.aws.amazon.com/ecs/v2.
2. En el panel de navegación, seleccione Clusters (Clústeres).
3. En la página Clusters (Clústeres), elija Create Cluster (Crear clúster).
4. En CLuster configuration (Configuración de clúster), para Cluster name (Nombre del clúster), introduzca un nombre único.

   El nombre puede contener hasta 255 letras (minúsculas y mayúsculas), números y guiones.
5. Para activar Información de contenedores, expanda Supervisión y, a continuación, active Uso de Información de contenedores.

Ahora puede crear definiciones de tareas, ejecutar tareas y lanzar servicios en el clúster. Para más información, consulte los siguientes temas:

- Creación de una definición de tareas
- Ejecución de tareas
- Crear un servicio

Configuración de Información de contenedores en clústeres nuevos de Amazon ECS mediante AWS CLI

Para habilitar Información de contenedores en todos los nuevos clústeres de forma predeterminada, escriba el siguiente comando.

```
aws ecs put-account-setting --name "containerInsights" --value "enabled"
```

Si no ha utilizado el comando anterior para habilitar Información de contenedores en todos los nuevos clústeres de forma predeterminada, escriba el siguiente comando para crear un nuevo clúster con Información de contenedores habilitada. Debe ejecutar la versión 1.16.200 o posterior de la AWS CLI para que el siguiente comando funcione.

```
aws ecs create-cluster --cluster-name myCICluster --settings
 "name=containerInsights,value=enabled"
```

Desactivación de Información de contenedores en clústeres de Amazon ECS

Para desactivar Información de contenedores en un clúster existente de Amazon ECS, ingrese el siguiente comando.

```
aws ecs update-cluster-settings --cluster myCICluster --settings
 name=containerInsights,value=disabled
```

Configuración de Información de contenedores en Amazon ECS mediante AWS Distro para OpenTelemetry

Utilice esta sección si desea utilizar AWS Distro para OpenTelemetry con el fin de configurar Información de contenedores de CloudWatch en un clúster de Amazon ECS. Para obtener más información acerca de AWS Distro para Open Telemetry, consulte AWS Distro para OpenTelemetry.

En estos pasos se da por sentado que ya cuenta con un clúster que ejecuta Amazon ECS. Para obtener más información sobre el uso de AWS Distro para OpenTelemetry con Amazon ECS y la configuración de un clúster de Amazon ECS para este fin, consulte Configuración del recopilador de AWS Distro para OpenTelemetry en Amazon Elastic Container Service.

Paso 1: Cree un rol de tarea

El primer paso consiste en crear un rol de tarea en el clúster que el recopilador de OpenTelemetry de AWS usará.

Cómo crear un rol de tarea para AWS Distro para OpenTelemetry

1. Abra la consola de IAM en https://console.aws.amazon.com/iam/.
2. En el panel de navegación, elija Policies (Políticas) y, a continuación, seleccione Create policy (Crear política).
3. Elija la pestaña JSON y copie la siguiente política:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "logs:PutLogEvents",
                "logs:CreateLogGroup",
```

```
                "logs:CreateLogStream",
                "logs:DescribeLogStreams",
                "logs:DescribeLogGroups",
                "ssm:GetParameters"
            ],
            "Resource": "*"
        }
    ]
}
```

4. Elija Revisar política.
5. En nombre, ingrese **`AWSDistroOpenTelemetryPolicy`**, después, elija Create policy (Crear política).
6. En el panel de navegación de la izquierda, elija Roles y, a continuación, seleccione Create Role (Crear rol).
7. En la lista de servicios, elija Elastic Container Service.
8. Debajo en la misma página, elija Elastic Container Service Task (Tarea de Elastic Container Service), luego Next: Permissions (Siguiente: Permisos).
9. En la lista de políticas, busque AWSDistroOpenTelemetryPolicy.
10. Marque la casilla situada junto a AWSDistroOpenTelemetryPolicy.
11. Elija Next: Tags (Siguiente: Etiquetas) y, a continuación, seleccione Next: Review (Siguiente: Revisar).
12. En Role name (Nombre de rol), ingrese **`AWSOpenTelemetryTaskRole`** y luego elija Create role (Crear rol).

Paso 2: Cree un rol de ejecución de tarea

El siguiente paso es crear un rol de ejecución de tareas para el recopilador OpenTelemetry de AWS.

Cómo crear un rol de ejecución de tareas para AWS Distro para OpenTelemetry

1. Abra la consola de IAM en https://console.aws.amazon.com/iam/.
2. En el panel de navegación de la izquierda, seleccione Roles y, a continuación, seleccione Create Role (Crear rol).
3. En la lista de servicios, elija Elastic Container Service.
4. Debajo en la misma página, elija Elastic Container Service Task (Tarea de Elastic Container Service), luego Next: Permissions (Siguiente: Permisos).

5. En la lista de políticas, busque AmazonECSTaskExecutionRolePolicy y, a continuación, seleccione la casilla de selección situada junto a AmazonECSTaskExecutionRolePolicy.
6. En la lista de políticas, busque CloudWatchLogsFullAccess y, a continuación, seleccione la casilla de selección situada junto a CloudWatchLogsFullAccess.
7. En la lista de políticas, busque AmazonSSMReadOnlyAccess y, a continuación, seleccione la casilla de selección situada junto a AmazonSSMReadOnlyAccess.
8. Elija Next: Tags (Siguiente: Etiquetas) y, a continuación, seleccione Next: Review (Siguiente: Revisar).
9. En Role name (Nombre de rol), ingrese **`AWSOpenTelemetryTaskExecutionRole`** y luego elija Create role (Crear rol).

Paso 3: Cree una definición de tarea

El siguiente paso es crear una definición de tarea.

Cómo crear una definición de tarea para AWS Distro para OpenTelemetry

1. Abra la consola en https://console.aws.amazon.com/ecs/v2.
2. En el panel de navegación, elija Task definitions (Definiciones de tareas).
3. Elija Create new task definition (Crear nueva definición de tarea) y Create new task definition (Crear nueva definición de tarea).
4. Para Task definition family (Familia de definiciones de tareas), especifique un nombre único para la definición de tareas.
5. Configure los contenedores y, a continuación, elija Siguiente.
6. En Métricas y registro, seleccione Usar recopilación de métricas.
7. Elija Siguiente.
8. Seleccione Crear.

Para obtener más información sobre el uso del recopilador de AWS OpenTelemetry con Amazon ECS, consulte Configuración del recopilador de AWS Distro para OpenTelemetry en Amazon Elastic Container Service.

Paso 4: Ejecute la tarea

El paso final es ejecutar la tarea que ha creado.

Cómo ejecutar la tarea de AWS Distro para OpenTelemetry

1. Abra la consola en https://console.aws.amazon.com/ecs/v2.
2. En el panel de navegación izquierdo, elija Task Definitions (Definiciones de tarea) y, a continuación, seleccione la tarea que acaba de crear.
3. Seleccione Acciones, Implementar y Ejecutar tarea.
4. Elija Deploy (Implementar), Run task (Ejecución de tareas).
5. En la sección Opciones de procesamiento, en Clúster existente, elija el clúster.
6. Seleccione Crear.
7. A continuación, puede verificar las métricas nuevas en la consola de CloudWatch.
8. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
9. En el panel de navegación izquierdo, en Metrics (Métricas).

   Debería ver un espacio de nombres ECS/ContainerInsights Elija ese espacio de nombres, se deberían ver ocho métricas.

Implementación del agente de CloudWatch para recopilar métricas de nivel de instancia EC2 en Amazon ECS

Para implementar el agente de CloudWatch a fin de recopilar métricas de nivel de instancia de los clústeres de Amazon ECS alojados en instancias EC2, utilice una configuración de inicio rápido con una configuración predeterminada o instale el agente manualmente para poder personalizarlo.

Ambos métodos requieren que tenga al menos un clúster de Amazon ECS implementado con un tipo de lanzamiento de EC2 y que el contenedor del agente de CloudWatch tenga acceso al servicio de metadatos de instancias de Amazon EC2 (IMDS). Para más información, consulte Metadatos de instancia y datos de usuario.

Estos métodos también suponen que tiene instalada la AWS CLI. Además, para ejecutar los comandos en los procedimientos siguientes, debe iniciar sesión en una cuenta o rol que tenga las políticas IAMFullAccess y AmazonECS_FullAccess.

Temas

- Configuración rápida mediante AWS CloudFormation
- Configuración manual y personalizada

Configuración rápida mediante AWS CloudFormation

Para utilizar la configuración rápida, escriba el siguiente comando para usar AWS CloudFormation a fin de instalar el agente. Sustituya *cluster-name* (cluster-nombre) y *cluster-region* (clúster-región) por el nombre y la región del clúster de Amazon ECS.

Este comando crea los roles de IAM CWAgentECSTaskRole y CWAgentECSExecutionRole. Si estos roles ya existen en su cuenta, utilice `ParameterKey=CreateIAMRoles,ParameterValue=False` en lugar de `ParameterKey=CreateIAMRoles,ParameterValue=True` cuando escriba el comando. De lo contrario, el comando fallará.

```
ClusterName=cluster-name
Region=cluster-region
curl -O https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
insights/latest/ecs-task-definition-templates/deployment-mode/daemon-service/cwagent-
ecs-instance-metric/cloudformation-quickstart/cwagent-ecs-instance-metric-cfn.json
aws cloudformation create-stack --stack-name CWAgentECS-${ClusterName}-${Region} \
    --template-body file://cwagent-ecs-instance-metric-cfn.json \
    --parameters ParameterKey=ClusterName,ParameterValue=${ClusterName} \
                 ParameterKey=CreateIAMRoles,ParameterValue=True \
    --capabilities CAPABILITY_NAMED_IAM \
    --region ${Region}
```

(Alternativa) Uso de sus propios roles de IAM

Si desea utilizar su propio rol de tarea de ECS personalizado y el rol de ejecución de tareas de ECS en lugar de los roles CWAgentECSTaskRole y CWAgentECSExecutionRole, asegúrese primero de que el rol que se va a usar como rol de tarea de ECS tiene CloudWatchAgentServerPolicy asociada. Además, asegúrese de que el rol que se va a utilizar como rol de ejecución de tareas de ECS tiene asociadas las políticas CloudWatchAgentServerPolicy y AmazonECSTaskExecutionRolePolicy.
A continuación, escriba el siguiente comando. En el comando, sustituya *task-role-arn* por el ARN de su rol de tarea de ECS personalizado y *execution-role-arn* por el ARN de su rol de ejecución de tareas de ECS personalizado.

```
ClusterName=cluster-name
Region=cluster-region
TaskRoleArn=task-role-arn
ExecutionRoleArn=execution-role-arn
```

```
curl -O https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
insights/latest/ecs-task-definition-templates/deployment-mode/daemon-service/cwagent-
ecs-instance-metric/cloudformation-quickstart/cwagent-ecs-instance-metric-cfn.json
aws cloudformation create-stack --stack-name CWAgentECS-${ClusterName}-${Region} \
    --template-body file://cwagent-ecs-instance-metric-cfn.json \
    --parameters ParameterKey=ClusterName,ParameterValue=${ClusterName} \
                 ParameterKey=TaskRoleArn,ParameterValue=${TaskRoleArn} \
                 ParameterKey=ExecutionRoleArn,ParameterValue=${ExecutionRoleArn} \
    --capabilities CAPABILITY_NAMED_IAM \
    --region ${Region}
```

Solución de problemas de la configuración rápida

Para comprobar el estado de la pila de AWS CloudFormation, escriba el siguiente comando.

```
ClusterName=cluster-name
Region=cluster-region
aws cloudformation describe-stacks --stack-name CWAgentECS-$ClusterName-$Region --
region $Region
```

Si ve que el `StackStatus` es distinto de `CREATE_COMPLETE` o `CREATE_IN_PROGRESS`, compruebe los eventos de pila para encontrar el error. Escriba el siguiente comando.

```
ClusterName=cluster-name
Region=cluster-region
aws cloudformation describe-stack-events --stack-name CWAgentECS-$ClusterName-$Region
 --region $Region
```

Para verificar el estado del servicio del servicio del daemon `cwagent`, ingrese el siguiente comando. En la salida, debería ver que el `runningCount` es igual al `desiredCount` en la sección `deployment`. Si no es igual, compruebe la sección `failures` en la salida.

```
ClusterName=cluster-name
Region=cluster-region
aws ecs describe-services --services cwagent-daemon-service --cluster $ClusterName --
region $Region
```

También puede utilizar la consola de CloudWatch Logs para verificar el registro del agente. Busque el grupo de registro /ecs/ecs-cwagent-daemon-service.

Eliminación de la pila de AWS CloudFormation del agente de CloudWatch

Si necesita eliminar la pila de AWS CloudFormation, escriba el siguiente comando.

```
ClusterName=cluster-name
Region=cluster-region
aws cloudformation delete-stack --stack-name CWAgentECS-${ClusterName}-${Region} --
region ${Region}
```

Configuración manual y personalizada

Siga los pasos de esta sección para implementar manualmente el agente de CloudWatch para recopilar métricas de nivel de instancia de los clústeres de Amazon ECS alojados en instancias EC2.

Políticas y roles de IAM necesarios

Se requieren dos roles de IAM. Debe crearlos si aún no existen. Para obtener más información sobre estos roles, consulte IAM roles for Tasks (Roles IAM para tareas) y Amazon ECS Task Execution Role (Rol de ejecución de tareas de Amazon ECS).

- Un ECS task role (rol de tarea de ECS), que el agente de CloudWatch utiliza para publicar métricas. Si este rol ya existe, debe asegurarse de que tiene la política `CloudWatchAgentServerPolicy` asociada.
- Un ECS task execution role (rol de ejecución de tareas de ECS), que el agente de Amazon ECS utiliza para lanzar el agente de CloudWatch. Si este rol ya existe, debe asegurarse de que tiene las políticas `CloudWatchAgentServerPolicy` y `AmazonECSTaskExecutionRolePolicy` asociadas.

Si aún no dispone de estos roles, puede utilizar los siguientes comandos para crearlos y asociar las políticas necesarias. Este primer comando crea el rol de tarea de ECS.

```
aws iam create-role --role-name CWAgentECSTaskRole \
    --assume-role-policy-document "{\"Version\": \"2012-10-17\",\"Statement\": [{\"Sid
\": \"\",\"Effect\": \"Allow\",\"Principal\": {\"Service\": \"ecs-tasks.amazonaws.com
\"},\"Action\": \"sts:AssumeRole\"}]}"
```

Después de escribir el comando anterior, anote el valor de `Arn` de la salida de los comandos como “TaskRoleArn”. Tendrá que usarse más tarde cuando se cree la definición de tarea. A continuación, escriba el siguiente comando para asociar las políticas necesarias.

```
aws iam attach-role-policy --policy-arn arn:aws:iam::aws:policy/
CloudWatchAgentServerPolicy \
```

```
    --role-name CWAgentECSTaskRole
```

Este siguiente comando crea el rol de ejecución de tareas de ECS.

```
aws iam create-role --role-name CWAgentECSExecutionRole \
    --assume-role-policy-document "{\"Version\": \"2012-10-17\",\"Statement\": [{\"Sid
\": \"\",\"Effect\": \"Allow\",\"Principal\": {\"Service\": \"ecs-tasks.amazonaws.com
\"},\"Action\": \"sts:AssumeRole\"}]}"
```

Después de escribir el comando anterior, anote el valor de `Arn` de la salida de los comandos como “ExecutionRoleArn”. Tendrá que usarse más tarde cuando se cree la definición de tarea. A continuación, escriba los siguientes comandos para asociar las políticas necesarias.

```
aws iam attach-role-policy --policy-arn arn:aws:iam::aws:policy/
CloudWatchAgentServerPolicy \
    --role-name CWAgentECSExecutionRole

aws iam attach-role-policy --policy-arn arn:aws:iam::aws:policy/service-role/
AmazonECSTaskExecutionRolePolicy \
    --role-name CWAgentECSExecutionRole
```

Cree la definición de tarea y lance el servicio del daemon

Cree una definición de tarea y úsela para lanzar el agente de CloudWatch como servicio del daemon. Para crear la definición de tarea, escriba el siguiente comando. En las primeras líneas, sustituya los marcadores de posición con los valores reales de la implementación. *`logs-region`* es la Región donde se encuentra CloudWatch Logs, y *`cluster-region`* es la Región donde se encuentra el clúster. *`task-role-arn`* es el Arn del rol de tarea de ECS que utiliza y *`execution-role-arn`* es el Arn del rol de ejecución de tarea de ECS.

```
TaskRoleArn=task-role-arn
ExecutionRoleArn=execution-role-arn
AWSLogsRegion=logs-region
Region=cluster-region
curl https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
insights/latest/ecs-task-definition-templates/deployment-mode/daemon-service/cwagent-
ecs-instance-metric/cwagent-ecs-instance-metric.json \
    | sed "s|{{task-role-arn}}|${TaskRoleArn}|;s|{{execution-role-arn}}|
${ExecutionRoleArn}|;s|{{awslogs-region}}|${AWSLogsRegion}|" \
    | xargs -0 aws ecs register-task-definition --region ${Region} --cli-input-json
```

A continuación, ejecute el siguiente comando para lanzar el servicio de daemon. Sustituya *cluster-name* y *cluster-region* por el nombre y la Región del clúster de Amazon ECS.

> **Important**
>
> Elimine todas las estrategias del proveedor de capacidad antes de ejecutar este comando. De lo contrario, el comando no funcionará.

```
ClusterName=cluster-name
Region=cluster-region
aws ecs create-service \
    --cluster ${ClusterName} \
    --service-name cwagent-daemon-service \
    --task-definition ecs-cwagent-daemon-service \
    --scheduling-strategy DAEMON \
    --region ${Region}
```

Si aparece este mensaje de error, `An error occurred (InvalidParameterException) when calling the CreateService operation: Creation of service was not idempotent`, ya ha creado un servicio de daemon llamado `cwagent-daemon-service`. Debe eliminar ese servicio primero, utilizando el siguiente comando como ejemplo.

```
ClusterName=cluster-name
Region=cluster-region
aws ecs delete-service \
    --cluster ${ClusterName} \
    --service cwagent-daemon-service \
    --region ${Region} \
    --force
```

(Opcional) Configuración avanzada

Opcionalmente, puede utilizar SSM para especificar otras opciones de configuración para el agente de CloudWatch en los clústeres de Amazon ECS alojados en instancias EC2. Estas opciones son las siguientes:

- `metrics_collection_interval`: la frecuencia en segundos con la que el agente de CloudWatch recopila métricas. El valor predeterminado es 60. El rango va de 1 a 172 000.

- `endpoint_override`: (opcional) especifica un punto de enlace diferente al que enviar los registros. Es posible que desee hacerlo si realiza la publicación desde un clúster de una VPC y desea que los datos de registro vayan a un punto de enlace de la VPC.

  El valor de `endpoint_override` debe ser una cadena que sea una URL.
- `force_flush_interval`: especifica en segundos la cantidad máxima de tiempo que los registros permanecen en el búfer de memoria antes de enviarse al servidor. Independientemente del valor de este campo, si el tamaño de los registros en el búfer alcanza 1 MB, los registros se envían inmediatamente al servidor. El valor de predeterminado es de 5 segundos.
- `region`: de forma predeterminada, el agente publica métricas en la misma Región en la que se encuentra la instancia de contenedor de Amazon ECS. Para anular esto, puede especificar una región diferente aquí. Por ejemplo, `"region" : "us-east-1"`

A continuación, se muestra un ejemplo de una configuración personalizada:

```
{
    "agent": {
        "region": "us-east-1"
    },
    "logs": {
        "metrics_collected": {
            "ecs": {
                "metrics_collection_interval": 30
            }
        },
        "force_flush_interval": 5
    }
}
```

Para personalizar la configuración del agente de CloudWatch en los contenedores de Amazon ECS

1. Asegúrese de que la política AmazonSSMReadOnlyAccess esté asociada al rol de ejecución de tarea de Amazon ECS. Para ello, puede escribir el siguiente comando. Este ejemplo presupone que el rol de ejecución de tarea de Amazon ECS es CWAgentECSExecutionRole. Si utiliza otro rol, sustituya el nombre de ese rol en el siguiente comando.

   ```
   aws iam attach-role-policy --policy-arn arn:aws:iam::aws:policy/
   AmazonSSMReadOnlyAccess \
           --role-name CWAgentECSExecutionRole
   ```

2. Cree el archivo de configuración personalizado de forma similar al ejemplo anterior. Asigne a este archivo el nombre `/tmp/ecs-cwagent-daemon-config.json`.

3. Ejecute el siguiente comando para colocar esta configuración en el almacén de parámetros. Sustituya *`cluster-region`* por la Región del clúster de Amazon ECS. Para ejecutar este comando, debe iniciar sesión en un usuario o rol que tenga la política AmazonSSMFullAccess.

```
Region=cluster-region
aws ssm put-parameter \
    --name "ecs-cwagent-daemon-service" \
    --type "String" \
    --value "`cat /tmp/ecs-cwagent-daemon-config.json`" \
    --region $Region
```

4. Descargue el archivo de definición de tarea en un archivo local, como `/tmp/cwagent-ecs-instance-metric.json`

```
curl https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
insights/latest/ecs-task-definition-templates/deployment-mode/daemon-service/
cwagent-ecs-instance-metric/cwagent-ecs-instance-metric.json -o /tmp/cwagent-ecs-
instance-metric.json
```

5. Modifique el archivo de definición de tarea. Elimine la siguiente sección:

```
"environment": [
                {
                    "name": "USE_DEFAULT_CONFIG",
                    "value": "True"
                }
            ],
```

Sustituya esa sección por la siguiente:

```
"secrets": [
                {
                    "name": "CW_CONFIG_CONTENT",
                    "valueFrom": "ecs-cwagent-daemon-service"
                }
            ],
```

6. Reinicie el agente como servicio de daemon siguiendo estos pasos:

a. Ejecute el siguiente comando de la .

```
TaskRoleArn=task-role-arn
ExecutionRoleArn=execution-role-arn
AWSLogsRegion=logs-region
Region=cluster-region
cat /tmp/cwagent-ecs-instance-metric.json \
    | sed "s|{{task-role-arn}}|${TaskRoleArn}|;s|{{execution-role-arn}}|
${ExecutionRoleArn}|;s|{{awslogs-region}}|${AWSLogsRegion}|" \
    | xargs -0 aws ecs register-task-definition --region ${Region} --cli-input-
json
```

b. Ejecute el siguiente comando para lanzar el servicio de daemon. Sustituya *cluster-name* y *cluster-region* por el nombre y la Región del clúster de Amazon ECS.

```
ClusterName=cluster-name
Region=cluster-region
aws ecs create-service \
    --cluster ${ClusterName} \
    --service-name cwagent-daemon-service \
    --task-definition ecs-cwagent-daemon-service \
    --scheduling-strategy DAEMON \
    --region ${Region}
```

Si aparece este mensaje de error, `An error occurred (InvalidParameterException) when calling the CreateService operation: Creation of service was not idempotent`, ya ha creado un servicio de daemon llamado `cwagent-daemon-service`. Debe eliminar ese servicio primero, utilizando el siguiente comando como ejemplo.

```
ClusterName=cluster-name
Region=Region
aws ecs delete-service \
    --cluster ${ClusterName} \
    --service cwagent-daemon-service \
    --region ${Region} \
    --force
```

Implementación de AWS Distro para OpenTelemetry a fin de recopilar métricas de nivel de instancia EC2 en clústeres de Amazon ECS

Siga los pasos de esta sección para usar AWS Distro para OpenTelemetry a fin de recopilar métricas de nivel de instancia EC2 en un clúster de Amazon ECS. Para obtener más información acerca de AWS Distro para OpenTelemetry, consulte AWS Distro para OpenTelemetry.

En estos pasos se presupone que ya tiene un clúster que ejecuta Amazon ECS. Este clúster debe implementarse con el tipo de lanzamiento EC2. Para obtener más información acerca del uso de AWS Distro para OpenTelemetry con Amazon ECS y acerca de la configuración de un clúster de Amazon ECS para este fin, consulte Configuración del recopilador de AWS Distro para OpenTelemetry en las métricas de nivel de instancia EC2 de Amazon Elastic Container Service para ECS EC2.

Temas

- Configuración rápida mediante AWS CloudFormation
- Configuración manual y personalizada

Configuración rápida mediante AWS CloudFormation

Descargue el archivo de plantilla de AWS CloudFormation para instalar el recopilador AWS Distro para OpenTelemetry para Amazon ECS en EC2. Ejecute el siguiente comando curl.

```
curl -O https://raw.githubusercontent.com/aws-observability/aws-otel-collector/main/
deployment-template/ecs/aws-otel-ec2-instance-metrics-daemon-deployment-cfn.yaml
```

Después de descargar el archivo de plantilla, ábralo y reemplace *PATH_TO_CloudFormation_TEMPLATE* con la ruta donde guardó el archivo de plantilla. A continuación, exporte los siguientes parámetros y ejecute el comando AWS CloudFormation, tal y como se muestra en el siguiente comando.

- Cluster_Name: el nombre del clúster de Amazon ECS
- AWS_Region: la Región a la que se enviarán los datos
- PATH_TO_CloudFormation_TEMPLATE: la ruta en la que guardó el archivo de plantilla de AWS CloudFormation.
- comando: para habilitar el recopilador de AWS Distro para OpenTelemetry para que recopile las métricas de nivel de instancia para Amazon ECS en Amazon EC2, debe especificar `--config=/etc/ecs/otel-instance-metrics-config.yaml` para este parámetro.

```
ClusterName=Cluster_Name
Region=AWS_Region
command=--config=/etc/ecs/otel-instance-metrics-config.yaml
aws cloudformation create-stack --stack-name AOCECS-${ClusterName}-${Region} \
--template-body file://PATH_TO_CloudFormation_TEMPLATE \
--parameters ParameterKey=ClusterName,ParameterValue=${ClusterName} \
ParameterKey=CreateIAMRoles,ParameterValue=True \
ParameterKey=command,ParameterValue=${command} \
--capabilities CAPABILITY_NAMED_IAM \
--region ${Region}
```

Después de ejecutar este comando, utilice la consola de Amazon ECS para ver si la tarea se está ejecutando.

Solución de problemas de la configuración rápida

Para comprobar el estado de la pila de AWS CloudFormation, escriba el siguiente comando.

```
ClusterName=cluster-name
Region=cluster-region
aws cloudformation describe-stack --stack-name AOCECS-$ClusterName-$Region --region
 $Region
```

Si ve que el valor de `StackStatus` es distinto de `CREATE_COMPLETE` o `CREATE_IN_PROGRESS`, verifique los eventos de pila para encontrar el error. Escriba el siguiente comando.

```
ClusterName=cluster-name
Region=cluster-region
aws cloudformation describe-stack-events --stack-name AOCECS-$ClusterName-$Region --
region $Region
```

Para verificar el estado del servicio del daemon `AOCECS`, ingrese el siguiente comando. En la salida, debería ver que el `runningCount` es igual al `desiredCount` en la sección de implementación. Si no es igual, verifique la sección de errores en la salida.

```
ClusterName=cluster-name
Region=cluster-region
aws ecs describe-services --services AOCECS-daemon-service --cluster $ClusterName --
region $Region
```

También puede utilizar la consola de CloudWatch Logs para verificar el registro del agente. Busque el grupo de registro /aws/ecs/containerinsights/{ClusterName}/performance.

Configuración manual y personalizada

Siga los pasos de esta sección para implementar manualmente AWS Distro para OpenTelemetry a fin de recopilar métricas de nivel de instancia de los clústeres de Amazon ECS alojados en instancias de Amazon EC2.

Paso 1: Políticas y roles necesarios

Se requieren dos roles de IAM. Debe crearlos si aún no existen. Para obtener más información sobre los roles, consulte Create IAM policy (Crear una política de IAM) y Create IAM role (Crear un rol de IAM).

Paso 2: Cree una definición de tarea

Cree una definición de tarea y úsela para lanzar AWS Distro para OpenTelemetry como servicio del daemon.

Si desea utilizar la plantilla de definición de tareas para la creación de uno de esos recursos, siga las instrucciones que aparecen en Crear una definición de tarea de EC2 de ECS para una instancia de EC2 con el recopilador de AWS OTel.

Si desea utilizar la consola de Amazon ECS para crear la definición de tarea, siga las instrucciones que aparecen en Instalar el recopilador de AWS OTel mediante la creación de una definición de tarea a través de la consola de AWS para métricas de instancias de EC2 de Amazon ECS.

Paso 3: Lance el servicio del daemon

Para lanzar AWS Distro para OpenTelemetry como un servicio daemon, siga las instrucciones que se indican en Ejecutar la tarea en Amazon Elastic Container Service (Amazon ECS) con el servicio del daemon.

(Opcional) Configuración avanzada

Opcionalmente, puede utilizar SSM para especificar otras opciones de configuración para AWS Distro para OpenTelemetry en los clústeres de Amazon ECS alojados en instancias de Amazon EC2. Para obtener más información sobre la creación de un archivo de configuración, consulte Custom OpenTelemetry Configuration (Configuración personalizada de OpenTelemetry). Para obtener más información acerca de las opciones que pueden usarse en el archivo de configuración, consulte Receptor de Información de contenedores de AWS.

Configurar Firelens para enviar registros a registros de CloudWatch

FireLens para Amazon ECS le permite utilizar parámetros de definición de tarea para dirigir registros a Amazon CloudWatch Logs para el almacenamiento y el análisis de registros. FireLens funciona con Fluent Bit y Fluentd. AWS le proporciona una imagen Fluent Bit o puede utilizar su propia imagen de Fluentd o Fluent Bit. La creación de definiciones de tareas de Amazon ECS con una configuración de FireLens se admite mediante los SDK de AWS, la AWS CLI y la AWS Management Console. Para obtener más información sobre CloudWatch Logs, consulte What is CloudWatch Logs? (¿Qué es CloudWatch Logs?).

Existen consideraciones clave al utilizar FireLens para Amazon ECS. Para obtener información, consulte Considerations (Consideraciones).

Para buscar las imágenes de AWS para Fluent Bit, consulte Uso de AWS para la imagen de Fluent Bit.

Para crear una definición de tarea que utilice una configuración FireLens, consulte Creating a task definition that uses a FireLens configuration (Creación de una definición de tarea que utilice una configuración de FireLens).

Ejemplo

En el siguiente ejemplo de definición de tarea se muestra cómo se especifica una configuración de registro que reenvíe registros a un grupo de registros de CloudWatch Logs. Para obtener más información, consulte ¿Qué es Amazon CloudWatch Logs? en la Guía del usuario de Amazon CloudWatch Logs.

En las opciones de configuración de registro, especifique el nombre del grupo de registro y la región en la que existe. Para que Fluent Bit cree el grupo de registro en su nombre, especifique `"auto_create_group":"true"`. También puede especificar el ID de la tarea como prefijo del flujo de registros, para facilitar el filtrado. Para obtener más información, consulte la sección Fluent Bit Plugin for CloudWatch Logs.

```
{
 "family": "firelens-example-cloudwatch",
 "taskRoleArn": "arn:aws:iam::123456789012:role/ecs_task_iam_role",
 "containerDefinitions": [
  {
   "essential": true,
   "image": "906394416424.dkr.ecr.us-west-2.amazonaws.com/aws-for-fluent-bit:latest",
```

```
   "name": "log_router",
   "firelensConfiguration": {
    "type": "fluentbit"
   },
   "logConfiguration": {
    "logDriver": "awslogs",
    "options": {
     "awslogs-group": "firelens-container",
     "awslogs-region": "us-west-2",
     "awslogs-create-group": "true",
     "awslogs-stream-prefix": "firelens"
    }
   },
   "memoryReservation": 50
   },
   {
    "essential": true,
    "image": "nginx",
    "name": "app",
    "logConfiguration": {
     "logDriver":"awsfirelens",
     "options": {
     "Name": "cloudwatch",
     "region": "us-west-2",
     "log_key": "log",
                                  "log_group_name": "/aws/ecs/containerinsights/
$(ecs_cluster)/application",
     "auto_create_group": "true",
     "log_stream_name": "$(ecs_task_id)"
    }
   },
   "memoryReservation": 100
  }
 ]
}
```

Configuración de Información de contenedores en Amazon EKS y Kubernetes

Información de contenedores es compatible con las versiones 1.23 y posteriores de Amazon EKS. El método de instalación de inicio rápido solo es compatible con las versiones 1.24 y posteriores.

El proceso general para configurar Información de contenedores en Amazon EKS o Kubernetes es el siguiente:

1. Compruebe que se cumplen los requisitos previos necesarios.
2. Configure el complemento de observabilidad de EKS de Amazon CloudWatch, el agente de CloudWatch o Distro para OpenTelemetry de AWS en el clúster para que envíe métricas a CloudWatch.

   > **Note**
   >
   > Para usar Información de contenedores con observabilidad mejorada para Amazon EKS, debe usar el complemento de observabilidad de EKS de Amazon CloudWatch o el agente CloudWatch. Para obtener más información sobre Información de contenedores, consulte Información de contenedores con observabilidad mejorada para Amazon EKS.
   > Para usar Información de contenedores con Fargate, debe usar Distro para OpenTelemetry de AWS. Fargate no admite Información de contenedores con observabilidad mejorada para Amazon EKS.

   > **Note**
   >
   > Información de contenedores ahora es compatible con los nodos de trabajo de Windows en un clúster de Amazon EKS. Información de contenedores con observabilidad mejorada para Amazon EKS también es compatible con Windows. Para obtener información sobre cómo habilitar Información de contenedores en Windows, consulte Uso del agente CloudWatch con observabilidad mejorada de Información de contenedores habilitada.

   Configure Fluent Bit o FluentD para enviar registros a registros de CloudWatch. (Esta opción está habilitada de forma predeterminada si instala el complemento de observabilidad de EKS de Amazon CloudWatch).

   Puede realizar estos pasos a la vez como parte de la configuración de inicio rápido si está utilizando el agente de CloudWatch o realizarlos por separado.
3. (Opcional) Configure el registro del plano de control de Amazon EKS.
4. (Opcional) Configure el agente de CloudWatch como un punto de enlace de StatsD en el clúster para que envíe las métricas de StatsD a CloudWatch.
5. (Opcional) Habilite App Mesh Envoy Access Logs.

Con la versión original de Información de contenedores, las métricas recopiladas y los registros incorporados se cobran como métricas personalizadas. Con Información de contenedores, con una observabilidad mejorada para Amazon EKS, las métricas y los registros de Información de contenedores se cobran por observación en lugar de cobrarse por métrica almacenada o registro incorporado. Para obtener más información sobre los precios de CloudWatch, consulte Precios de Amazon CloudWatch.

Temas

- Verificación de los requisitos previos de
- Uso del agente CloudWatch con observabilidad mejorada de Información de contenedores habilitada
- Uso de AWS Distro para OpenTelemetry
- Envíe los registros a CloudWatch Logs
- Actualización o eliminación de Información de contenedores en Amazon EKS y en Kubernetes

Verificación de los requisitos previos de

Antes de instalar Información de contenedores en Amazon EKS o Kubernetes, verifique lo siguiente: Estos prerrequisitos se aplican tanto si utiliza el agente de CloudWatch como AWS Distro para OpenTelemetry a fin de configurar Información de contenedores en clústeres de Amazon EKS.

- Cuenta con un clúster funcional de Amazon EKS o de Kubernetes con nodos asociados en una de las Regiones que admite Información de contenedores para Amazon EKS y Kubernetes. Para obtener las lista de las regiones admitidas, consulte Información de contenedores.
- Tiene `kubectl` instalado y se está ejecutando. Para obtener más información, consulte Installing `kubectl` en la Guía del usuario de Amazon EKS.
- Si utiliza Kubernetes con AWS en lugar de utilizar Amazon EKS, los siguientes prerrequisitos también son necesarios:
  - Asegúrese de que el clúster de Kubernetes tiene activado el control de acceso basado en roles (RBAC). Para obtener más información, consulte Using RBAC Authorization en la documentación de Kubernetes.
  - El kubelet ha habilitado el modo de autorización de Webhook. Para obtener más información, consulte Kubelet authentication/authorization en la documentación de Kubernetes.

También debe conceder permisos de IAM para permitir que los nodos de trabajo de Amazon EKS envíen métricas y registros a CloudWatch. Hay dos formas de hacer esto:

- Adjunte una política al rol de IAM de los nodos de trabajo. Esto funciona tanto para clústeres de Amazon EKS como para otros clústeres de Kubernetes.
- Utilice un rol de IAM para las cuentas de servicio para el clúster y adjunte la política a este rol. Esto solo funciona para clústeres de Amazon EKS.

La primera opción concede permisos a CloudWatch para todo el nodo, mientras que el uso de un rol de IAM para la cuenta de servicio da acceso a CloudWatch sólo a los pods adecuados de DaemonSet .

Asociación de una política al rol de IAM de los nodos de trabajo

Siga estos pasos para adjuntar la política al rol de IAM de los nodos de trabajo. Esto funciona tanto para los clústeres de Amazon EKS como para clústeres de Kubernetes fuera de Amazon EKS.

Para adjuntar la política necesaria al rol de IAM de los nodos de trabajo

1. Abra la consola de Amazon EC2 en https://console.aws.amazon.com/ec2/.
2. Seleccione una de las instancias de nodo de trabajo y elija el rol de IAM en la descripción.
3. En la página del rol de IAM, elija Attach policies (Adjuntar políticas).
4. En la lista de políticas, seleccione la casilla junto a CloudWatchAgentServerPolicy. Si es necesario, utilice el cuadro de búsqueda para encontrar la política.
5. Seleccione Asociar políticas.

Si está ejecutando un clúster de Kubernetes fuera de Amazon EKS, es posible que no tenga adjunto un rol de IAM a los nodos de trabajo. En ese caso, primero debe adjuntar un rol de IAM a la instancia y, a continuación, agregar la política tal y como se explica en los pasos anteriores. Para obtener más información sobre cómo se asocia un rol a una instancia, consulte Attaching an IAM Role to an Instance (Adjuntar un rol de IAM a una instancia) en la Guía del usuario de Amazon EC2 para instancias de Windows.

Si ejecuta un clúster de Kubernetes fuera de Amazon EKS y desea que se recopilen las id. de volumen de EBS en las métricas, debe agregar otra política al rol de IAM asociado a la instancia. Añada lo siguiente como política insertada. Para obtener más información consulte Adding and

Removing IAM Identity Permissions (Incorporación y eliminación de permisos de identidad de IAM) en la Guía del usuario de IAM.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "ec2:DescribeVolumes"
            ],
            "Resource": "*",
            "Effect": "Allow"
        }
    ]
}
```

Uso de un rol de cuenta de servicio de IAM

Este método solo funciona en clústeres de Amazon EKS.

Para conceder permiso a CloudWatch con un rol de cuenta de servicio IAM

1. Si aún no lo ha hecho, debe habilitar los roles de IAM para las cuentas de servicio en el clúster. Para obtener más información, consulte Habilitación de roles de IAM para cuentas de servicio en el clúster.
2. Si aún no lo ha hecho, debe configurar la cuenta de servicio para usar un rol de IAM. Para obtener más información, consulte Configuración de una cuenta de servicio de Kubernetes para asumir un rol de IAM.

   Cuando cree el rol, adjunte la política de IAM CloudWatchAgentServerPolicy al rol además de la política que cree para el rol. Además, la cuenta de servicio de Kubernetes asociada que está vinculada a este rol debe crearse en el espacio de nombres de `amazon-cloudwatch`, donde se implementarán los daemonsets CloudWatch y Fluent Bit en los próximos pasos
3. Si aún no lo ha hecho, debe adjuntar el rol de IAM con una cuenta de servicio en el clúster. Para obtener más información, consulte Configuración de una cuenta de servicio de Kubernetes para asumir un rol de IAM.

Uso del agente CloudWatch con observabilidad mejorada de Información de contenedores habilitada

Utilice las instrucciones de una de las siguientes secciones para configurar Información de contenedores en un clúster de Amazon EKS o un clúster de Kubernetes mediante el agente de CloudWatch. El método de instalación de inicio rápido solo es compatible con las versiones 1.24 y posteriores de Amazon EKS.

> **Note**
>
> Puede instalar Información de contenedores siguiendo las instrucciones de cualquiera de las siguientes secciones. No es necesario seguir los tres conjuntos de instrucciones.

Temas

Instalación del complemento de observabilidad de EKS de Amazon CloudWatch

Puede usar el complemento Amazon EKS para instalar Información de contenedores con observabilidad mejorada para Amazon EKS. El complemento instala el agente de CloudWatch para enviar métricas de infraestructura desde el clúster, instala Fluent Bit para enviar los registros del contenedor y también permite a Application Signals de CloudWatch enviar la telemetría de rendimiento de las aplicaciones.

Cuando utiliza la versión 1.5.0 o posterior del complemento de Amazon EKS, Información de contenedores se habilita en los nodos de trabajo de Linux y Windows del clúster. Actualmente, Application Signals no es compatible con Windows en Amazon EKS.

El complemento Amazon EKS no es compatible con los clústeres que ejecutan Kubernetes en lugar de Amazon EKS.

Para obtener más información acerca del complemento observabilidad EKS de Amazon CloudWatch, consulte Instalar el agente de CloudWatch mediante el complemento de EKS de observabilidad de Amazon CloudWatch.

Cómo instalar el complemento de observabilidad de EKS de Amazon CloudWatch

1. En primer lugar, configure los permisos necesarios adjuntando la política de IAM CloudWatchAgentServerPolicy a sus nodos de trabajo. Para ello, introduzca el siguiente comando. Sustituya *my-worker-node-role* por el rol de IAM que utilizan sus nodos de trabajo de Kubernetes.

```
aws iam attach-role-policy \
--role-name my-worker-node-role \
--policy-arn arn:aws:iam::aws:policy/CloudWatchAgentServerPolicy
```

2. Ingrese el siguiente comando para agregar el complemento:

```
aws eks create-addon --cluster-name my-cluster-name --addon-name amazon-cloudwatch-observability
```

Configuración de inicio rápido para Información de contenedores en Amazon EKS y Kubernetes

> **Important**
>
> Si va a instalar Información de contenedores en un clúster de Amazon EKS, recomendamos que utilice el complemento observabilidad de EKS de Amazon CloudWatch para la instalación, en lugar de seguir las instrucciones de esta sección. Además, para recuperar las redes de computación acelerada, debe usar el complemento de EKS Observabilidad de Amazon CloudWatch. Para obtener más información e instrucciones, consulte Instalación del complemento de observabilidad de EKS de Amazon CloudWatch.

Para completar la configuración de Información de contenedores, puede seguir las instrucciones de inicio rápido de esta sección. Si instala Información de contenedores en un clúster de Amazon EKS y sigue las instrucciones de esta sección a partir del 6 de noviembre de 2023, se instalará con observabilidad mejorada para Amazon EKS en el clúster.

> **Important**
>
> Antes de completar los pasos de esta sección, debe haber verificado los prerrequisitos incluidos los permisos de IAM. Para obtener más información, consulte Verificación de los requisitos previos de .

También puede seguir las instrucciones de las dos secciones siguientes: Configuración del agente de CloudWatch para recopilar las métricas del clúster y Envíe los registros a CloudWatch Logs. Estas secciones proporcionan más detalles sobre cómo funciona el agente de CloudWatch con Amazon EKS y con Kubernetes, pero se requiere que realice más pasos de instalación.

Con la versión original de Información de contenedores, las métricas recopiladas y los registros incorporados se cobran como métricas personalizadas. Con Información de contenedores, con una observabilidad mejorada para Amazon EKS, las métricas y los registros de Información de contenedores se cobran por observación en lugar de cobrarse por métrica almacenada o registro incorporado. Para obtener más información sobre los precios de CloudWatch, consulte Precios de Amazon CloudWatch.

> **Note**
>
> Amazon ha lanzado recientemente Fluent Bit como la solución de registro predeterminada para Información de contenedores con importantes mejoras de rendimiento. Se recomienda que utilice Fluent Bit en lugar de Fluentd.

Inicio rápido con el operador del agente de CloudWatch y Fluent Bit

Hay dos configuraciones para Fluent Bit: una versión optimizada y una versión que proporciona una experiencia más similar a Fluentd. La configuración de inicio rápido utiliza la versión optimizada. Para obtener más detalles sobre la configuración compatible con Fluentd, consulte Configure Fluent Bit como DaemonSet para enviar registros a CloudWatch Logs.

El operador del agente de CloudWatch es un contenedor adicional que se instala en un clúster de Amazon EKS. Sigue el modelo del operador OpenTelemetry para Kubernetes. El operador administra el ciclo de vida de los recursos de Kubernetes en un clúster. Instala el agente deCloudWatch, DCGM Exporter (NVIDIA) y AWS Neuron Monitor en un clúster de Amazon EKS y los administra. Fluent Bit y el agente de CloudWatch para Windows se instalan directamente en un clúster de Amazon EKS sin que el operador los administre.

Si busca una solución de autoridad de certificación más segura y con más características, el operador del agente de CloudWatch necesita cert-manager, una solución ampliamente adoptada para la administración de certificados TLS en Kubernetes. El uso de cert-manager simplifica el proceso de obtención, renovación, administración y uso de estos certificados. Asegura que los certificados sean válidos y estén actualizados, e intenta renovarlos en un momento configurado

antes de que caduquen. Además, cert-manager también facilita la emisión de certificados desde una variedad de orígenes compatibles, como AWS Certificate Manager Private Certificate Authority.

Implementación de Información de contenedores mediante el inicio rápido

1. Instale cert-manager si aún no está instalado en el clúster. Para obtener más información, consulte cert-manager Installation.
2. Introduzca el siguiente comando para instalar las definiciones de recursos personalizados (CRDS).

```
curl https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
insights/main/k8s-quickstart/cwagent-custom-resource-definitions.yaml | kubectl
 apply --server-side -f -
```

3. Introduzca el siguiente comando para instalar el operador. En este comando, *my_cluster_name* es el nombre del clúster de Amazon EKS o el de Kubernetes y *my_cluster_region* es el nombre de la región en la que se publican los registros. Le recomendamos que utilice la misma región en la que implemente el clúster para reducir los costos de transferencia de datos salientes de AWS.

```
ClusterName=my-cluster-name
RegionName=my-cluster-region
curl https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-
container-insights/main/k8s-quickstart/cwagent-operator-rendered.yaml | sed 's/
{{cluster_name}}/'${ClusterName}'/g;s/{{region_name}}/'${RegionName}'/g' | kubectl
 apply -f -
```

   Por ejemplo, para implementar Información de contenedores en el clúster denominado `MyCluster` y publicar los registros y las métricas en el Oeste de EE. UU (Oregón), ingrese el siguiente comando.

```
ClusterName='MyCluster'
RegionName='us-west-2'
curl https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-
container-insights/main/k8s-quickstart/cwagent-operator-rendered.yaml | sed 's/
{{cluster_name}}/'${ClusterName}'/g;s/{{region_name}}/'${RegionName}'/g' | kubectl
 apply -f -
```

Migración desde Información de contenedores

Si ya tiene Información de contenedores configurado en un clúster de Amazon EKS y desea migrar a Información de contenedores con observabilidad mejorada para Amazon EKS, consulte Actualización a Información de contenedores con observabilidad mejorada para Amazon EKS

Eliminación de Información de contenedores

Si desea eliminar Información de contenedores después de utilizar la configuración de inicio rápido, escriba los siguientes comandos.

```
ClusterName=my-cluster-name
RegionName=my-cluster-region
curl https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-
container-insights/main/k8s-quickstart/cwagent-operator-rendered.yaml | sed 's/
{{cluster_name}}/'${ClusterName}'/g;s/{{region_name}}/'${RegionName}'/g' | kubectl
 delete -f -
curl https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
insights/main/k8s-quickstart/cwagent-custom-resource-definitions.yaml | kubectl delete
 -f -
```

Configuración del agente de CloudWatch para recopilar las métricas del clúster

> **Important**
>
> Si instala Información de contenedores en un clúster de Amazon EKS, recomendamos que utilice el complemento observabilidad de EKS de Amazon CloudWatch, en lugar de seguir las instrucciones de esta sección. Para obtener más información e instrucciones, consulte Instalación del complemento de observabilidad de EKS de Amazon CloudWatch.

Para configurar Información de contenedores para recopilar métricas de, puede seguir los pasos de Configuración de inicio rápido para Información de contenedores en Amazon EKS y Kubernetes o los de esta sección. En los pasos que se describen a continuación, configure el agente de CloudWatch para que pueda recopilar las métricas de los clústeres.

Si instala Información de contenedores en un clúster de Amazon EKS y sigue las instrucciones de esta sección a partir del 6 de noviembre de 2023, se instalará con observabilidad mejorada para Amazon EKS en el clúster.

Paso 1: Cree un espacio de nombres para CloudWatch

Siga el paso que se describe a continuación para crear un espacio de nombres denominado `amazon-cloudwatch` para CloudWatch. Puede omitir este paso si ya ha creado este espacio de nombres.

Para crear un espacio de nombres para CloudWatch

- Escriba el siguiente comando.

```
kubectl apply -f https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-
container-insights/latest/k8s-deployment-manifest-templates/deployment-mode/
daemonset/container-insights-monitoring/cloudwatch-namespace.yaml
```

Paso 2: Cree una cuenta de servicio en el clúster

Siga el paso que se describe a continuación para crear una cuenta de servicio para el agente de CloudWatch, si todavía no dispone de una.

Para crear una cuenta de servicio para el agente de CloudWatch

- Escriba el siguiente comando.

```
kubectl apply -f https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-
container-insights/latest/k8s-deployment-manifest-templates/deployment-mode/
daemonset/container-insights-monitoring/cwagent/cwagent-serviceaccount.yaml
```

Si no ha seguido los pasos anteriores, pero ya tiene una cuenta de servicio que desea utilizar para el agente de CloudWatch, debe asegurarse de que sigue las siguientes reglas. Además, en el resto de los pasos de la instalación de Información de contenedores debe utilizar el nombre de esa cuenta de servicio en lugar de `cloudwatch-agent`.

```
rules:
  - apiGroups: [""]
    resources: ["pods", "nodes", "endpoints"]
    verbs: ["list", "watch"]
  - apiGroups: [ "" ]
    resources: [ "services" ]
    verbs: [ "list", "watch" ]
  - apiGroups: ["apps"]
```

```
    resources: ["replicasets", "daemonsets", "deployments", "statefulsets"]
    verbs: ["list", "watch"]
  - apiGroups: ["batch"]
    resources: ["jobs"]
    verbs: ["list", "watch"]
  - apiGroups: [""]
    resources: ["nodes/proxy"]
    verbs: ["get"]
  - apiGroups: [""]
    resources: ["nodes/stats", "configmaps", "events"]
    verbs: ["create", "get"]
  - apiGroups: [""]
    resources: ["configmaps"]
    resourceNames: ["cwagent-clusterleader"]
    verbs: ["get","update"]
  - nonResourceURLs: ["/metrics"]
    verbs: ["get", "list", "watch"]
```

Paso 3: Cree un ConfigMap para el agente de CloudWatch

Siga los pasos que se describen a continuación para crear un ConfigMap para el agente de CloudWatch.

Para crear un ConfigMap para el agente de CloudWatch

1. Descargue el archivo YAML de ConfigMap en el host de cliente de `kubectl` ejecutando el siguiente comando:

   ```
   curl -O https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
   insights/latest/k8s-deployment-manifest-templates/deployment-mode/daemonset/
   container-insights-monitoring/cwagent/cwagent-configmap.yaml
   ```

2. Edite el archivo YAML que ha descargado, tal y como se indica a continuación:

   - cluster_name: en la sección `kubernetes`, reemplace `{{cluster_name}}` con el nombre del clúster. Elimine los caracteres `{{}}`. De forma alternativa, si utiliza un clúster de Amazon EKS, puede eliminar el campo `"cluster_name"` y el valor. Si lo hace, el agente de CloudWatch detecta el nombre del clúster a partir de las etiquetas de Amazon EC2.

3. (Opcional) Realice más cambios en ConfigMap en función de sus requisitos de supervisión, tal y como se indica a continuación:

- metrics_collection_interval: en la sección `kubernetes`, puede especificar la frecuencia con la que el agente recopila las métricas. El valor predeterminado es de 60 segundos. El valor predeterminado del intervalo de recopilación de cadvisor en kubelet es de 15 segundos, por lo que no debe establecer este valor en menos de 15 segundos.
- endpoint_override: en la sección `logs`, puede especificar el punto de enlace de CloudWatch Logs si desea anular el punto de enlace predeterminado. Es posible que desee hacerlo si realiza la publicación desde un clúster de una VPC y desea que los datos vayan a un punto de enlace de la VPC.
- force_flush_interval: en la sección `logs`, puede especificar el intervalo para los eventos de registro por lotes antes de que se publiquen en CloudWatch Logs. El valor predeterminado es de 5 segundos.
- Región: de forma predeterminada, el agente publica las métricas en la Región donde se encuentra el nodo de trabajo. Para cambiar este comportamiento, puede agregar un campo `region` en la sección `agent`: por ejemplo `"region":"us-west-2"`
- Sección statsd: si desea que el agente CloudWatch Logs se ejecute también como agente de escucha StatsD en cada nodo de trabajo de su clúster, puede agregar una la sección`statsd` a las `metrics`, como en el ejemplo siguiente. Para obtener información sobre otras opciones de StatsD para esta sección, consulte Recuperación de las métricas personalizadas con StatsD .

```
"metrics": {
  "metrics_collected": {
    "statsd": {
      "service_address":":8125"
    }
  }
}
```

A continuación, se muestra un ejemplo completo de la sección JSON.

```
{
    "agent": {
        "region": "us-east-1"
    },
    "logs": {
        "metrics_collected": {
            "kubernetes": {
```

```
                "cluster_name": "MyCluster",
                "metrics_collection_interval": 60
            }
        },
        "force_flush_interval": 5,
        "endpoint_override": "logs.us-east-1.amazonaws.com"
    },
    "metrics": {
        "metrics_collected": {
            "statsd": {
                "service_address": ":8125"
            }
        }
    }
}
```

4. Cree el ConfigMap en el clúster ejecutando el siguiente comando.

```
kubectl apply -f cwagent-configmap.yaml
```

Paso 4: Implemente el agente de CloudWatch como un DaemonSet

Para terminar la instalación del agente de CloudWatch y comenzar a recopilar las métricas de contenedor, siga los pasos que se describen a continuación.

Para implementar el agente de CloudWatch como un DaemonSet

1. • Si no desea utilizar StatsD en el clúster, escriba el siguiente comando.

```
kubectl apply -f https://raw.githubusercontent.com/aws-samples/amazon-
cloudwatch-container-insights/latest/k8s-deployment-manifest-templates/
deployment-mode/daemonset/container-insights-monitoring/cwagent/cwagent-
daemonset.yaml
```

   • Si desea utilizar StatsD, siga estos pasos:

      a. Descargue el archivo YAML del DaemonSet en el host de cliente de `kubectl` ejecutando el siguiente comando.

```
curl -O  https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-
container-insights/latest/k8s-deployment-manifest-templates/deployment-
mode/daemonset/container-insights-monitoring/cwagent/cwagent-daemonset.yaml
```

b. Borre el comentario de la sección `port` en el archivo `cwagent-daemonset.yaml` como se indica a continuación:

```
ports:
  - containerPort: 8125
    hostPort: 8125
    protocol: UDP
```

c. Implemente el agente de CloudWatch en el clúster con el siguiente comando.

```
kubectl apply -f cwagent-daemonset.yaml
```

d. Implemente el agente de CloudWatch en los nodos de Windows de su clúster ejecutando el siguiente comando. El oyente StatsD no es compatible con el agente de CloudWatch en Windows.

```
kubectl apply -f cwagent-daemonset-windows.yaml
```

2. Compruebe que el agente está implementado ejecutando el siguiente comando.

```
kubectl get pods -n amazon-cloudwatch
```

Cuando finaliza, el agente de CloudWatch crea un grupo de registros denominado `/aws/containerinsights/`*`Cluster_Name`*`/performance` y envía los eventos de registro de rendimiento a este grupo de registros. Si también configura el agente como un agente de escucha de StatsD, el agente también escucha las métricas de StatsD en el puerto 8125 con la dirección IP del nodo en el que está programado el pod de la aplicación.

Resolución de problemas

Si el agente no se implementa correctamente, pruebe lo siguiente:

- Ejecute el siguiente comando para obtener la lista de pods.

```
kubectl get pods -n amazon-cloudwatch
```

- Ejecute el siguiente comando y compruebe los eventos de la parte inferior de la salida.

```
kubectl describe pod pod-name -n amazon-cloudwatch
```

- Ejecute el siguiente comando para comprobar los registros.

```
kubectl logs pod-name  -n amazon-cloudwatch
```

Uso de AWS Distro para OpenTelemetry

Puede configurar Información de contenedores de manera que recopile métricas de clústeres de Amazon EKS mediante el recopilador de AWS Distro para OpenTelemetry. Para obtener más información acerca de AWS Distro para OpenTelemetry, consulte AWS Distro para OpenTelemetry.

> **Important**
>
> Si hace la instalación mediante AWS Distro para OpenTelemetry, instalará Información de contenedores, pero no obtendrá Información de contenedores con observabilidad mejorada para Amazon EKS. No recopilarás las métricas detalladas compatibles con Información de contenedores con observabilidad mejorada para Amazon EKS.

La forma de configurar Información de contenedores depende de si el clúster está alojado en instancias de Amazon EC2 o en AWS Fargate (Fargate).

Clústeres de Amazon EKS alojados en Amazon EC2

Si aún no lo ha hecho, asegúrese de que ha cumplido los prerrequisitos incluidos los roles de IAM necesarios. Para obtener más información, consulte Verificación de los requisitos previos de .

Amazon proporciona un gráfico de Helm que puede utilizar para configurar la supervisión de Amazon Elastic Kubernetes Service en Amazon EC2. Esta supervisión utiliza el recolector de AWS Distro para OpenTelemetry (ADOT) para las métricas, y Fluent Bit para los registros. Por lo tanto, el gráfico de Helm es útil para los clientes que utilizan Amazon EKS en Amazon EC2 y desean recopilar métricas y registros para enviarlos a Información de contenedores de CloudWatch. Para obtener más información acerca de este gráfico de Helm, consulte Gráfico de Helm de ADOT para EKS en métricas y registros de EC2 en Información de contenedores de Amazon CloudWatch.

También puede utilizar las instrucciones del resto de esta sección.

En primer lugar, implemente el recopilador de AWS Distro para OpenTelemetry como DaemonSet si escribe el siguiente comando.

```
curl https://raw.githubusercontent.com/aws-observability/aws-otel-collector/main/
deployment-template/eks/otel-container-insights-infra.yaml |
kubectl apply -f -
```

Utilice el siguiente comando para confirmar que el recopilador se está ejecutando.

```
kubectl get pods -l name=aws-otel-eks-ci -n aws-otel-eks
```

Si el resultado de este comando incluye varios pods en el estado `Running`, el recopilador se está ejecutando y recopilando métricas del clúster. El recopilador crea un grupo de registros denominado `aws/containerinsights/`*`cluster-name`*`/performance` y le envía los eventos de registro de rendimiento.

Para obtener información sobre cómo ver las métricas Información de contenedores en CloudWatch, consulte Visualización de las métricas de Información de contenedores.

AWS también ha proporcionado documentación sobre GitHub para esta situación. Si desea personalizar las métricas y los registros que Información de contenedores publica, consulte https://aws-otel.github.io/docs/getting-started/container-insights/eks-infra.

Clústeres de Amazon EKS alojados en Fargate

Para obtener instrucciones acerca de cómo configurar e implementar un recopilador ADOT de manera que recopile métricas del sistema de cargas de trabajo implementadas en un clúster de Amazon EKS en Fargate y las envíe a Información de contenedores de CloudWatch, consulte Información de contenedores de EKS en Fargate en la documentación de AWS Distro para OpenTelemetry.

Envíe los registros a CloudWatch Logs

Para enviar registros desde los contenedores a Amazon CloudWatch Logs, puede utilizar Fluent Bit o Fluentd. Para obtener más información, consulte Fluent Bit y Fluentd.

Si aún no está utilizando Fluentd, se recomienda que utilice Fluent Bit por los siguientes motivos:

- Fluent Bit tiene un espacio de recursos más pequeño y es más eficiente en el uso de la memoria y en la CPU que Fluentd. Si desea ver una comparación más detallada, consulte Comparación de rendimiento entre Fluent Bit y Fluentd.

- AWS desarrolla y mantiene la imagen de Fluent Bit. Esto le da a AWS la capacidad de adoptar nuevas características de imagen Fluent Bit y responder a los problemas mucho más rápido.

Temas

Comparación de rendimiento entre Fluent Bit y Fluentd

Las siguientes tablas muestran la ventaja de rendimiento que tiene Fluent Bit sobre Fluentd en los usos de la memoria y la CPU. Los siguientes números son solo para referencia y pueden cambiar según el entorno.

Registros por segundo	Uso de la CPU de Fluentd	Uso de la CPU de Fluent Bit con configuración compatible con Fluentd	Uso de la CPU de Fluent Bit con configuración optimizada
100	0,35 vCPU	0,02 vCPU	0,02 vCPU
1 000	0,32 vCPU	0,14 vCPU	0,11 vCPU
5 000	0,85 vCPU	0,48 vCPU	0,30 vCPU
10 000	0,94 vCPU	0,60 vCPU	0,39 vCPU

Registros por segundo	Uso de memoria de Fluentd	Uso de memoria de Fluent Bit con configuración compatible con Fluentd	Uso de memoria de Fluent Bit con configuración optimizada
100	153 MB	46 MB	37 MB
1 000	270 MB	45 MB	40 MB
5 000	320 MB	55 MB	45 MB
10 000	375 MB	92 MB	75 MB

Configure Fluent Bit como DaemonSet para enviar registros a CloudWatch Logs

Las siguientes secciones le ayudan a implementar Fluent Bit para enviar registros desde contenedores a CloudWatch Logs.

Temas

Diferencias en caso de que ya está usando Fluentd

Si ya está utilizando Fluentd para enviar registros desde contenedores a CloudWatch Logs, lea esta sección para ver las diferencias entre Fluentd y Fluent Bit. Si aún no está utilizando Fluentd con Información de contenedores de, puede saltar a Configuración de Fluent Bit.

Se le proporcionan dos configuraciones predeterminadas para Fluent Bit:

- Configuración optimizada de Fluent Bit: una configuración alineada con las mejores prácticas de Fluent Bit.

- Configuración compatible con Fluentd: una configuración que está alineada con el comportamiento de Fluentd tanto como sea posible.

En la siguiente lista se explican detalladamente las diferencias entre Fluentd y cada configuración de Fluent Bit.

- Diferencias en los nombres de flujo de registro: si utiliza la configuración optimizada de Fluent Bit, los nombres de flujo de registro serán diferentes.

  Bajo el título `/aws/containerinsights/Cluster_Name/application`

  - La configuración optimizada de Fluent Bit envía registros a *kubernetes-nodeName*-`application.var.log.containers.`*kubernetes-podName*_*kubernetes-namespace*_*kubernetes-container-name*-*kubernetes-containerID*
  - Fluentd envía registros a *kubernetes-podName*_*kubernetes-namespace*_*kubernetes-containerName*_*kubernetes-containerID*

  Bajo el título `/aws/containerinsights/Cluster_Name/host`

  - La configuración optimizada de Fluent Bit envía registros a *kubernetes-nodeName*`.`*host-log-file*
  - Fluentd envía registros a *host-log-file*-*Kubernetes-NodePrivateIp*

  Bajo el título `/aws/containerinsights/Cluster_Name/dataplane`

  - La configuración optimizada de Fluent Bit envía registros a *kubernetes-nodeName*`.`*dataplaneServiceLog*
  - Fluentd envía registros a *dataplaneServiceLog*-*Kubernetes-nodeName*
- Los archivos de registros `kube-proxy` y `aws-node` que Información de contenedores de ingresa se encuentran en diferentes ubicaciones. En la configuración de Fluentd, se encuentran en `/aws/containerinsights/Cluster_Name/application`. En la configuración optimizada de Fluent Bit, se encuentran en `/aws/containerinsights/Cluster_Name/dataplane`.
- La mayoría de los metadatos, como `pod_name` y `namespace_name` son los mismos en Fluent Bit y Fluentd, pero los siguientes son diferentes.
  - La configuración optimizada de Fluent Bit utiliza `docker_id` y el uso de Fluentd `Docker.container_id`.
  - Ambas configuraciones de Bit Fluent no utilizan los siguientes metadatos. Están presentes sólo en Fluentd: `container_image_id`, `master_url`, `namespace_id`, y `namespace_labels`.

Configuración de Fluent Bit

Para configurar Fluent Bit para recopilar registros de los contenedores, puede seguir los pasos de Configuración de inicio rápido para Información de contenedores en Amazon EKS y Kubernetes o puede seguir los pasos de esta sección.

Con cualquiera de los métodos, el rol de IAM que está adjunto a los nodos del clúster debe tener permisos suficientes. Para obtener más información sobre los permisos necesarios para ejecutar un clúster de Amazon EKS, consulte Amazon EKS IAM Policies, Roles, and Permissions (Permisos, roles y políticas de IAM de Amazon EKS) en la Guía del usuario de Amazon EKS.

En los pasos que se describen a continuación, va a configurar Fluent Bit como DaemonSet para enviar registros a CloudWatch Logs. Cuando se completa este paso, Fluent Bit crea los siguientes grupos de registros si todavía no existen.

> **Important**
>
> Si ya tiene Fluentd configurado en Información de contenedores y el DaemonSet de Fluentd no funciona según lo esperado (esto puede ocurrir si utiliza el tiempo de ejecución `containerd`), debe desinstalarlo antes de instalar FluentBit para evitar que Fluentd procese los mensajes de registro de errores de Fluentd. De lo contrario, debe desinstalar Fluentd inmediatamente después de haber instalado FluentBit correctamente. Desinstalar Fluentd después de instalar Fluent Bit garantiza la ontinuidad del registro durante este proceso de migración. Solo se necesita uno de Fluent Bit o Fluentd para enviar registros a registros de CloudWatch.

Nombre de grupo de registro	Fuente de registros
`/aws/containerinsights/` *`Cluster_Name`* `/application`	Todos los archivos de registros de `/var/log/containers`
`/aws/containerinsights/` *`Cluster_Name`* `/host`	Archivos de registros de `/var/log/dmesg` , `/var/log/secure` y `/var/log/messages`
`/aws/containerinsights/` *`Cluster_Name`* `/dataplane`	Los registros en `/var/log/journal` para `kubelet.service` , `kubeproxy.service` y `docker.service` .

Para instalar Fluent Bit con el fin de enviar registros desde contenedores a CloudWatch Logs

1. Si aún no tiene un espacio de nombres llamado `amazon-cloudwatch`, cree uno con el siguiente comando:

```
kubectl apply -f https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-
container-insights/latest/k8s-deployment-manifest-templates/deployment-mode/
daemonset/container-insights-monitoring/cloudwatch-namespace.yaml
```

2. Ejecute el siguiente comando para crear un ConfigMap denominado `cluster-info` con el nombre del clúster y la Región a la que se enviarán los registros. Sustituya *`cluster-name`* y *`cluster-region`* por el nombre y la Región del clúster.

```
ClusterName=cluster-name
RegionName=cluster-region
FluentBitHttpPort='2020'
FluentBitReadFromHead='Off'
[[ ${FluentBitReadFromHead} = 'On' ]] && FluentBitReadFromTail='Off'||
 FluentBitReadFromTail='On'
[[ -z ${FluentBitHttpPort} ]] && FluentBitHttpServer='Off' ||
 FluentBitHttpServer='On'
kubectl create configmap fluent-bit-cluster-info \
--from-literal=cluster.name=${ClusterName} \
--from-literal=http.server=${FluentBitHttpServer} \
--from-literal=http.port=${FluentBitHttpPort} \
--from-literal=read.head=${FluentBitReadFromHead} \
--from-literal=read.tail=${FluentBitReadFromTail} \
--from-literal=logs.region=${RegionName} -n amazon-cloudwatch
```

   En este comando, `FluentBitHttpServer` para supervisar las métricas del complemento está activado de forma predeterminada. Para desactivarlo, cambie la tercera línea del comando a `FluentBitHttpPort=''` (cadena vacía) en el comando.

   Además, de forma predeterminada, Fluent Bit lee los archivos de registro en cola y capturará solo los registros nuevos después de implementarlos. Si desea lo contrario, establezca `FluentBitReadFromHead='On'` y recopilará todos los registros en el sistema de archivos.

3. Descargue e implemente el daemonSet de Fluent Bit en el clúster con uno de los siguientes comandos.

   - Si desea la configuración optimizada de Fluent Bit para computadoras con Linux, ejecute este comando.

```
kubectl apply -f https://raw.githubusercontent.com/aws-samples/amazon-
cloudwatch-container-insights/latest/k8s-deployment-manifest-templates/
deployment-mode/daemonset/container-insights-monitoring/fluent-bit/fluent-
bit.yaml
```

- Si desea la configuración optimizada de Fluent Bit para computadoras con Windows, ejecute este comando.

```
kubectl apply -f https://raw.githubusercontent.com/aws-samples/amazon-
cloudwatch-container-insights/latest/k8s-deployment-manifest-templates/
deployment-mode/daemonset/container-insights-monitoring/fluent-bit/fluent-bit-
windows.yaml
```

- Si utiliza computadoras con Linux y desea la configuración de Fluent Bit que sea más similar a Fluentd, ejecute este comando.

```
kubectl apply -f https://raw.githubusercontent.com/aws-samples/amazon-
cloudwatch-container-insights/latest/k8s-deployment-manifest-templates/
deployment-mode/daemonset/container-insights-monitoring/fluent-bit/fluent-bit-
compatible.yaml
```

> **Important**
>
> De forma predeterminada, la configuración daemonset de Fluent Bit establece el nivel de registro en INFO, lo que puede generar mayores costes de ingesta de registros de CloudWatch. Si quiere reducir el volumen y los costes de la ingesta de registros, puede cambiar el nivel de registro a ERROR.
> Para obtener más información acerca de cómo reducir el volumen del registro, consulte (Opcional) Reducción del volumen de registros del Fluent Bit

4. Especifique el siguiente comando para validar la implementación. Cada nodo debe tener un pod denominado fluent-bit-*.

```
kubectl get pods -n amazon-cloudwatch
```

Los pasos anteriores crean los siguientes recursos en el clúster:

- Una cuenta de servicio denominada `Fluent-Bit` en el espacio de nombres `amazon-cloudwatch`. Esta cuenta de servicio se utiliza para ejecutar el DaemonSet de FluentD. Para obtener más información, consulte Managing Service Accounts en la documentación de Kubernetes.
- Un rol de clúster denominado `Fluent-Bit-role` en el espacio de nombres `amazon-cloudwatch`. Este rol de clúster concede los permisos `get`, `list` y `watch` para los registros de los pods a la cuenta de servicio `Fluent-Bit`. Para obtener información, consulte API Overview en la documentación de Kubernetes.
- Un ConfigMap denominado `Fluent-Bit-config` en el espacio de nombres `amazon-cloudwatch`. Este ConfigMap contiene la configuración que va a utilizar Fluent Bit. Para obtener más información, consulte Configure a Pod to Use a ConfigMap en la documentación de tareas de Kubernetes.

Si desea verificar la configuración de Fluent Bit, siga estos pasos.

Verifique la configuración de Fluent Bit

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, seleccione Grupos de registro.
3. Asegúrese de que está en la Región donde ha implementado Fluent Bit.
4. Verifique la lista de los grupos de registros de la Región. Debería ver lo siguiente:
   - `/aws/containerinsights/`*`Cluster_Name`*`/application`
   - `/aws/containerinsights/`*`Cluster_Name`*`/host`
   - `/aws/containerinsights/`*`Cluster_Name`*`/dataplane`
5. Desplácese a uno de estos grupos de registro y verifique la Last Event Time (Hora del último evento) para los flujos de registro. Si es reciente en relación con el momento en el que implementó Fluent Bit, se verifica la configuración.

   Puede haber un ligero retraso en la creación del grupo de registro de `/dataplane`. Esto es normal, ya que estos grupos de registro sólo se crean cuando el Fluent Bit comienza a enviar registros para ese grupo de registros.

Compatibilidad con registros de varias líneas

Para obtener información sobre cómo usar Fluent Bit con registros de varias líneas, consulte las siguientes secciones de la documentación de Fluent Bit:

- Análisis multilínea
- Líneas múltiples y contenedores (v1.8)
- Núcleo multilínea (v1.8)
- Utilice siempre líneas múltiples en la entrada trasera

(Opcional) Reducción del volumen de registros del Fluent Bit

De forma predeterminada, se envían registros de aplicaciones de Fluent Bit y metadatos de Kubernetes a CloudWatch. Si desea reducir el volumen de datos que se envía a CloudWatch, puede detener el envío de uno o ambos de estos orígenes de datos a CloudWatch.

Para detener los registros de aplicaciones de Fluent Bit, elimine la siguiente sección del archivo `Fluent-Bit.yaml`.

```
[INPUT]
        Name                tail
        Tag                 application.*
        Path                /var/log/containers/fluent-bit*
        Parser              docker
        DB                  /fluent-bit/state/flb_log.db
        Mem_Buf_Limit       5MB
        Skip_Long_Lines     On
        Refresh_Interval    10
```

Para evitar que los metadatos de Kubernetes se añadan a los eventos de registro que se envían a CloudWatch, agregue los siguientes filtros a la sección `application-log.conf` en el archivo `Fluent-Bit.yaml`. Sustituya *<Metadata_1>* y los campos similares por los identificadores de metadatos reales.

```
application-log.conf: |
    [FILTER]
        Name                nest
        Match               application.*
        Operation           lift
        Nested_under        kubernetes
```

```
        Add_prefix          Kube.

    [FILTER]
        Name                modify
        Match               application.*
        Remove              Kube.<Metadata_1>
        Remove              Kube.<Metadata_2>
        Remove              Kube.<Metadata_3>

    [FILTER]
        Name                nest
        Match               application.*
        Operation           nest
        Wildcard            Kube.*
        Nested_under        kubernetes
        Remove_prefix       Kube.
```

Resolución de problemas

Si no ve estos grupos de registros y está mirando en la Región correcta, verifique los registros de los pods del DaemonSet de Fluent Bit para buscar el error.

Ejecute el siguiente comando y asegúrese de que el estado es `Running`.

```
kubectl get pods -n amazon-cloudwatch
```

Si los registros tienen errores relacionados con los permisos de IAM, consulte el rol de IAM que está adjunto a los nodos del clúster. Para obtener más información sobre los permisos necesarios para ejecutar un clúster de Amazon EKS, consulte Amazon EKS IAM Policies, Roles, and Permissions (Permisos, roles y políticas de IAM de Amazon EKS) en la Guía del usuario de Amazon EKS.

Si el estado del pod es `CreateContainerConfigError`, obtenga el error exacto ejecutando el siguiente comando.

```
kubectl describe pod pod_name -n amazon-cloudwatch
```

Panel de control

Puede crearse un panel para supervisar las métricas de cada complemento en ejecución. Puede ver los datos de los bytes de entrada y salida y de las tasas de procesamiento de registros, así como los errores de salida y las tasas de intento y error. Para visualizar estas métricas, se deberá instalar el

agente de CloudWatch con la colección de métricas de Prometheus para los clústeres de Amazon EKS y de Kubernetes. Para obtener más información acerca de cómo se configuran estos paneles, consulte Instale el agente de CloudWatch con la colección de métricas de Prometheus en clústeres de Amazon EKS y de Kubernetes.

> **Note**
>
> Antes de configurar este panel, debe configurar Información de contenedores de para las métricas de Prometheus. Para obtener más información, consulte Supervisión de métricas de Información de contenedores de Prometheus.

Para crear un panel para las métricas de Prometheus de Fluent Bit

1. Cree variables de entorno, mediante el reemplazo de los valores de la derecha en las siguientes líneas para que concuerden con la implementación.

```
DASHBOARD_NAME=your_cw_dashboard_name
REGION_NAME=your_metric_region_such_as_us-west-1
CLUSTER_NAME=your_kubernetes_cluster_name
```

2. Ejecute el siguiente comando para crear el panel.

```
curl https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
insights/latest/k8s-deployment-manifest-templates/deployment-mode/service/cwagent-
prometheus/sample_cloudwatch_dashboards/fluent-bit/cw_dashboard_fluent_bit.json \
| sed "s/{{YOUR_AWS_REGION}}/${REGION_NAME}/g" \
| sed "s/{{YOUR_CLUSTER_NAME}}/${CLUSTER_NAME}/g" \
| xargs -0 aws cloudwatch put-dashboard --dashboard-name ${DASHBOARD_NAME} --
dashboard-body
```

(Opcional) Configure Fluentd como DaemonSet para que envíe registros a registros de CloudWatch

> **Warning**
>
> La compatibilidad con Información de contenedores para Fluentd está en mantenimiento actualmente, lo que significa que AWS no proporcionará ninguna actualización adicional para Fluentd y que se está planeando darlo de baja próximamente. Además, la configuración actual de Fluentd para Información de contenedores está utilizando una versión antigua de

la imagen de Fluentd `fluent/fluentd-kubernetes-daemonset:v1.10.3-debian-cloudwatch-1.0` que no tiene las últimas mejoras y parches de seguridad. Para obtener la imagen de Fluentd más reciente compatible con la comunidad de código abierto, consulte fluentd-kubernetes-daemonset.
Se recomienda que migre para utilizar FluentBit con Información de contenedores siempre que sea posible. El uso de FluentBit como reenviador de registros para Información de contenedores proporciona importantes ganancias de rendimiento.
Para obtener más información, consulte Configure Fluent Bit como DaemonSet para enviar registros a CloudWatch Logs y Diferencias en caso de que ya está usando Fluentd.

Para configurar Fluentd para recopilar registros de sus contenedores, puede seguir los pasos de Configuración de inicio rápido para Información de contenedores en Amazon EKS y Kubernetes o puede seguir los pasos de esta sección. En los pasos que se describen a continuación, va a configurar Fluentd como DaemonSet para enviar registros a registros de CloudWatch. Cuando se complete este paso, Fluentd creará los siguientes grupos de registros si no existen ya.

Nombre de grupo de registro	Fuente de registros
`/aws/containerinsights/` *`Cluster_Name`* `/application`	Todos los archivos de registros de `/var/log/containers`
`/aws/containerinsights/` *`Cluster_Name`* `/host`	Archivos de registros de `/var/log/dmesg` , `/var/log/secure` y `/var/log/messages`
`/aws/containerinsights/` *`Cluster_Name`* `/dataplane`	Los registros en `/var/log/journal` para `kubelet.service` , `kubeproxy.service` y `docker.service` .

Paso 1: Cree un espacio de nombres para CloudWatch

Siga el paso que se describe a continuación para crear un espacio de nombres denominado `amazon-cloudwatch` para CloudWatch. Puede omitir este paso si ya ha creado este espacio de nombres.

Para crear un espacio de nombres para CloudWatch

- Escriba el siguiente comando.

```
kubectl apply -f https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-
container-insights/latest/k8s-deployment-manifest-templates/deployment-mode/
daemonset/container-insights-monitoring/cloudwatch-namespace.yaml
```

Paso 2: Instalar Fluentd

Inicie este proceso descargando Fluentd. Cuando termine estos pasos, la implementación creará los siguientes recursos en el clúster:

- Una cuenta de servicio denominada `fluentd` en el espacio de nombres `amazon-cloudwatch`. Esta cuenta de servicio se utiliza para ejecutar el DaemonSet de Fluentd. Para obtener más información, consulte Managing Service Accounts en la documentación de Kubernetes.
- Un rol de clúster denominado `fluentd` en el espacio de nombres `amazon-cloudwatch`. Este rol de clúster concede los permisos `get`, `list` y `watch` para los registros de los pods a la cuenta de servicio `fluentd`. Para obtener información, consulte API Overview en la documentación de Kubernetes.
- Un ConfigMap denominado `fluentd-config` en el espacio de nombres `amazon-cloudwatch`. Este ConfigMap contiene la configuración que va a utilizar Fluentd. Para obtener más información, consulte Configure a Pod to Use a ConfigMap en la documentación de tareas de Kubernetes.

Para instalar Fluentd

1. Cree un ConfigMap denominado `cluster-info` con el nombre del clúster y la región de AWS a la que se enviarán los registros. Ejecute el siguiente comando sustituyendo los marcadores de posición de los nombres de clúster y de región.

   ```
   kubectl create configmap cluster-info \
   --from-literal=cluster.name=cluster_name \
   --from-literal=logs.region=region_name -n amazon-cloudwatch
   ```

2. Descargue e implemente el DaemonSet de Fluentd en el clúster ejecutando el siguiente comando. Asegúrese de que esté utilizando la imagen de contenedor con la arquitectura correcta. El manifiesto de ejemplo solo funciona en instancias x86 e ingresará `CrashLoopBackOff` si cuenta con instancias de Advanced RISC Machine (ARM) en el clúster. El daemonSet de Fluentd no tiene una imagen de Docker oficial de varias arquitecturas que le permita utilizar una etiqueta para varias imágenes subyacentes y dejar que el tiempo de

ejecución del contenedor extraiga la correcta. La imagen ARM de Fluentd utiliza una etiqueta diferente con un sufijo `arm64`.

```
kubectl apply -f https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-insights/latest/k8s-deployment-manifest-templates/deployment-mode/daemonset/container-insights-monitoring/fluentd/fluentd.yaml
```

> **Note**
>
> Debido a un cambio reciente para optimizar la configuración de Fluentd y minimizar el impacto de las solicitudes de la API de Fluentd en los puntos de enlace de la API de Kubernetes, la opción “Mirar” para los filtros de Kubernetes se ha desactivado de forma predeterminada. Para obtener más detalles, consulte fluent-plugin-kubernetes_metadata_filter.

3. Valide la implementación ejecutando el siguiente comando. Cada nodo debe tener un pod denominado `fluentd-cloudwatch-*`.

```
kubectl get pods -n amazon-cloudwatch
```

Paso 3: Verifique la configuración de Fluentd

Para comprobar la configuración de Fluentd, siga los pasos a continuación.

Cómo verificar la configuración de Fluentd para Información de contenedores

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, seleccione Grupos de registro. Asegúrese de que está en la región donde ha implementado Fluentd en los contenedores.

   En la lista de grupos de registros de la región, debería ver lo siguiente:

   - `/aws/containerinsights/Cluster_Name/application`
   - `/aws/containerinsights/Cluster_Name/host`
   - `/aws/containerinsights/Cluster_Name/dataplane`

   Si ve estos grupos de registros, la configuración de Fluentd es correcta.

Compatibilidad con registros de varias líneas

El 19 de agosto de 2019, añadimos compatibilidad con registros de varias líneas para los registros recopilados por Fluentd.

De forma predeterminada, el iniciador de entrada de registro de varias líneas es cualquier carácter sin espacios en blanco. Esto significa que todas las líneas de registro que comienzan con un carácter que no tiene espacios en blanco se consideran una nueva entrada de registro de varias líneas.

Si sus propios registros de aplicación utilizan un iniciador de varias líneas diferente, puede conseguir que se admitan realizando dos cambios en el archivo `fluentd.yaml`.

En primer lugar, exclúyalos de la compatibilidad de varias líneas predeterminada añadiendo los nombres de ruta de los archivos de registro a un campo `exclude_path` de la sección `containers` de `fluentd.yaml`. A continuación, se muestra un ejemplo.

```
<source>
  @type tail
  @id in_tail_container_logs
  @label @containers
  path /var/log/containers/*.log
  exclude_path ["full_pathname_of_log_file*", "full_pathname_of_log_file2*"]
```

A continuación, añada un bloque de archivos de registro al archivo `fluentd.yaml`. El siguiente ejemplo se utiliza para el archivo de registro del agente de CloudWatch, que utiliza una expresión regular de marca temporal como iniciador de varias líneas. Puede copiar este bloque y añadirlo a `fluentd.yaml`. Cambie las líneas indicadas para reflejar el nombre del archivo de registro de la aplicación y el iniciador de varias líneas que desee utilizar.

```
<source>
  @type tail
  @id in_tail_cwagent_logs
  @label @cwagentlogs
  path /var/log/containers/cloudwatch-agent*
  pos_file /var/log/cloudwatch-agent.log.pos
  tag *
  read_from_head true
 <parse>
    @type json
```

```
    time_format %Y-%m-%dT%H:%M:%S.%NZ
 </parse>
</source>
```

```
<label @cwagentlogs>
  <filter **>
    @type kubernetes_metadata
    @id filter_kube_metadata_cwagent
  </filter>

  <filter **>
    @type record_transformer
    @id filter_cwagent_stream_transformer
    <record>
      stream_name ${tag_parts[3]}
    </record>
 </filter>

  <filter **>
    @type concat
    key log
    multiline_start_regexp /^\d{4}[-/]\d{1,2}[-/]\d{1,2}/
    separator ""
    flush_interval 5
    timeout_label @NORMAL
 </filter>

 <match **>
    @type relabel
    @label @NORMAL
 </match>
</label>
```

(Opcional) Reducción del volumen de registro de Fluentd

De forma predeterminada, se envían los registros de aplicaciones de Fluentd y metadatos de Kubernetes a CloudWatch. Si desea reducir el volumen de datos que se envía a CloudWatch, puede detener el envío de uno o ambos de estos orígenes de datos.

Para detener los registros de aplicaciones Fluentd, elimine la siguiente sección del archivo `fluentd.yaml`.

```
<source>
  @type tail
  @id in_tail_fluentd_logs
  @label @fluentdlogs
  path /var/log/containers/fluentd*
  pos_file /var/log/fluentd.log.pos
  tag *
  read_from_head true
   <parse>
   @type json
   time_format %Y-%m-%dT%H:%M:%S.%NZ
  </parse>
</source>

<label @fluentdlogs>
  <filter **>
    @type kubernetes_metadata
    @id filter_kube_metadata_fluentd
  </filter>

  <filter **>
    @type record_transformer
    @id filter_fluentd_stream_transformer
    <record>
      stream_name ${tag_parts[3]}
    </record>
   </filter>

   <match **>
     @type relabel
     @label @NORMAL
   </match>
</label>
```

Para evitar que los metadatos de Kubernetes se añadan a los eventos de registro que se envían a CloudWatch, agregue una línea a la sección `record_transformer` en el archivo `fluentd.yaml`. Añada la siguiente línea en el origen de registros en el que desea eliminar esos metadatos.

```
remove_keys $.kubernetes.pod_id, $.kubernetes.master_url,
 $.kubernetes.container_image_id, $.kubernetes.namespace_id
```

Por ejemplo:

```
<filter **>
  @type record_transformer
  @id filter_containers_stream_transformer
  <record>
    stream_name ${tag_parts[3]}
  </record>
  remove_keys $.kubernetes.pod_id, $.kubernetes.master_url,
 $.kubernetes.container_image_id, $.kubernetes.namespace_id
</filter>
```

Resolución de problemas

Si no ve estos grupos de registros y está mirando en la región correcta, compruebe los registros de los pods del DaemonSet de Fluentd para buscar el error.

Ejecute el siguiente comando y asegúrese de que el estado es `Running`.

```
kubectl get pods -n amazon-cloudwatch
```

En los resultados del comando anterior, anote el nombre del pod que comienza por `fluentd-cloudwatch`. Utilice este nombre de pod en el siguiente comando.

```
kubectl logs pod_name -n amazon-cloudwatch
```

Si los registros tienen errores relacionados con los permisos de IAM, verifique el rol de IAM que está adjunto a los nodos del clúster. Para obtener más información sobre los permisos necesarios para ejecutar un clúster de Amazon EKS, consulte Amazon EKS IAM Policies, Roles, and Permissions (Permisos, roles y políticas de IAM de Amazon EKS) en la Guía del usuario de Amazon EKS.

Si el estado del pod es `CreateContainerConfigError`, obtenga el error exacto ejecutando el siguiente comando.

```
kubectl describe pod pod_name -n amazon-cloudwatch
```

Si el estado del pod es `CrashLoopBackOff`, asegúrese de que la arquitectura de la imagen de contenedor de Fluentd sea la misma que la del nodo cuando se instaló Fluentd. Si el clúster tiene

nodos x86 y ARM64, se puede usar una etiqueta kubernetes.io/arch para colocar las imágenes en el nodo correcto. Para obtener más información, consulte kuberntes.io/arch.

(Opcional) Configure el registro del plano de control de Amazon EKS

Si utiliza Amazon EKS, tiene la opción de habilitar el registro del plano de control de Amazon EKS, para proporcionar registros de auditoría y de diagnóstico directamente desde el plano de control de Amazon EKS a CloudWatch Logs. Para obtener más información, consulte Amazon EKS Control Plane Logging (Registro del plano de control de Amazon EKS).

(Opcional) Habilite los registros de acceso de App Mesh Envoy

Puede configurar Fluentd de Información de contenedores para enviar registros de acceso de App Mesh Envoy a registros de CloudWatch. Para obtener más información, consulte Logging (Registros).

Para que los registros de acceso de Envoy se envíen a CloudWatch Logs

1. Configure Fluentd en el clúster. Para obtener más información, consulte (Opcional) Configure Fluentd como DaemonSet para que envíe registros a registros de CloudWatch.
2. Configure los registros de acceso de Envoy en los nodos virtuales. Para obtener instrucciones, consulte Logging (Registros). No olvide configurar la ruta de acceso de registro como **`/dev/stdout`** en cada nodo virtual.

Cuando haya terminado, los registros de acceso de Envoy se enviarán al grupo de registros de `/aws/containerinsights/Cluster_Name/application`.

(Opcional) Habilitar la característica Use_Kubelet para clústeres grandes

De forma predeterminada, la característica Use_Kubelet está deshabilitada en el complemento FluentBit Kubernetes. Habilitar esta característica puede reducir el tráfico hacia el servidor de la API y mitigar el problema de que el servidor de la API sea un cuello de botella. Recomendamos que habilite esta característica para clústeres grandes.

Para habilitar Use_Kubelet, agregue primero los nodos y los permisos de nodos/proxy a la configuración de ClusterRole.

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: fluent-bit-role
```

```
rules:
  - nonResourceURLs:
      - /metrics
    verbs:
      - get
  - apiGroups: [""]
    resources:
      - namespaces
      - pods
      - pods/logs
      - nodes
      - nodes/proxy
    verbs: ["get", "list", "watch"]
```

En la configuración de Daemonset, esta característica necesita acceso a la red de host. La versión de la imagen de `amazon/aws-for-fluent-bit` debería ser 2.12.0 o posterior, o la versión de imagen de bits fluida debería ser 1.7.2 o posterior.

```
apiVersion: apps/v1
kind: DaemonSet
metadata:
  name: fluent-bit
  namespace: amazon-cloudwatch
  labels:
    k8s-app: fluent-bit
    version: v1
    kubernetes.io/cluster-service: "true"
spec:
  selector:
    matchLabels:
      k8s-app: fluent-bit
  template:
    metadata:
      labels:
        k8s-app: fluent-bit
        version: v1
        kubernetes.io/cluster-service: "true"
    spec:
      containers:
      - name: fluent-bit
        image: amazon/aws-for-fluent-bit:2.19.0
        imagePullPolicy: Always
        env:
```

```
            - name: AWS_REGION
              valueFrom:
                configMapKeyRef:
                  name: fluent-bit-cluster-info
                  key: logs.region
            - name: CLUSTER_NAME
              valueFrom:
                configMapKeyRef:
                  name: fluent-bit-cluster-info
                  key: cluster.name
            - name: HTTP_SERVER
              valueFrom:
                configMapKeyRef:
                  name: fluent-bit-cluster-info
                  key: http.server
            - name: HTTP_PORT
              valueFrom:
                configMapKeyRef:
                  name: fluent-bit-cluster-info
                  key: http.port
            - name: READ_FROM_HEAD
              valueFrom:
                configMapKeyRef:
                  name: fluent-bit-cluster-info
                  key: read.head
            - name: READ_FROM_TAIL
              valueFrom:
                configMapKeyRef:
                  name: fluent-bit-cluster-info
                  key: read.tail
            - name: HOST_NAME
              valueFrom:
                fieldRef:
                  fieldPath: spec.nodeName
            - name: HOSTNAME
              valueFrom:
                fieldRef:
                  apiVersion: v1
                  fieldPath: metadata.name
            - name: CI_VERSION
              value: "k8s/1.3.8"
        resources:
            limits:
              memory: 200Mi
```

```
            requests:
              cpu: 500m
              memory: 100Mi
        volumeMounts:
        # Please don't change below read-only permissions
        - name: fluentbitstate
          mountPath: /var/fluent-bit/state
        - name: varlog
          mountPath: /var/log
          readOnly: true
        - name: varlibdockercontainers
          mountPath: /var/lib/docker/containers
          readOnly: true
        - name: fluent-bit-config
          mountPath: /fluent-bit/etc/
        - name: runlogjournal
          mountPath: /run/log/journal
          readOnly: true
        - name: dmesg
          mountPath: /var/log/dmesg
          readOnly: true
      terminationGracePeriodSeconds: 10
      hostNetwork: true
      dnsPolicy: ClusterFirstWithHostNet
      volumes:
      - name: fluentbitstate
        hostPath:
          path: /var/fluent-bit/state
      - name: varlog
        hostPath:
          path: /var/log
      - name: varlibdockercontainers
        hostPath:
          path: /var/lib/docker/containers
      - name: fluent-bit-config
        configMap:
          name: fluent-bit-config
      - name: runlogjournal
        hostPath:
          path: /run/log/journal
      - name: dmesg
        hostPath:
          path: /var/log/dmesg
      serviceAccountName: fluent-bit
```

```
      tolerations:
      - key: node-role.kubernetes.io/master
        operator: Exists
        effect: NoSchedule
      - operator: "Exists"
        effect: "NoExecute"
      - operator: "Exists"
        effect: "NoSchedule"
```

La configuración del complemento de Kubernetes debería ser similar a la siguiente:

```
[FILTER]
        Name                kubernetes
        Match               application.*
        Kube_URL            https://kubernetes.default.svc:443
        Kube_Tag_Prefix     application.var.log.containers.
        Merge_Log           On
        Merge_Log_Key       log_processed
        K8S-Logging.Parser  On
        K8S-Logging.Exclude Off
        Labels              Off
        Annotations         Off
        Use_Kubelet         On
        Kubelet_Port        10250
        Buffer_Size         0
```

Actualización o eliminación de Información de contenedores en Amazon EKS y en Kubernetes

Siga los pasos de estas secciones para actualizar la imagen de contenedor del agente de CloudWatch o para eliminar Información de contenedores de un clúster de Amazon EKS o de Kubernetes.

Temas

- Actualización a Información de contenedores con observabilidad mejorada para Amazon EKS
- Actualización de la imagen del contenedor del agente de CloudWatch
- Eliminación del agente de CloudWatch y Fluen Bit para Información de contenedores

Actualización a Información de contenedores con observabilidad mejorada para Amazon EKS

> **Important**
>
> Si va a actualizar o instalar Información de contenedores en un clúster de Amazon EKS, recomendamos que utilice el complemento de EKS de observabilidad de Amazon CloudWatch para la instalación, en lugar de seguir las instrucciones de esta sección. Además, para recuperar las métricas de computación acelerada, debe usar el complemento de EKS de observabilidad de Amazon CloudWatch. Para obtener más información e instrucciones, consulte Instalación del complemento de observabilidad de EKS de Amazon CloudWatch.

Información de contenedores con observabilidad mejorada para Amazon EKS es la versión más reciente de Información de contenedores. Recopila métricas detalladas de los clústeres que ejecutan Amazon EKS y ofrece paneles seleccionados y de uso inmediato para analizar en detalle la telemetría de las aplicaciones y la infraestructura. Para obtener más información sobre Información de contenedores, consulte Información de contenedores con observabilidad mejorada para Amazon EKS.

Si instaló la versión original de Información de contenedores en un clúster de Amazon EKS y desea actualizarla a una versión más reciente con observabilidad mejorada, siga las instrucciones de esta sección.

> **Important**
>
> Antes de completar los pasos de esta sección, debe haber verificado los requisitos previos, incluido cert-manager. Para obtener más información, consulte Inicio rápido con el operador del agente de CloudWatch y Fluent Bit.

Para actualizar un clúster de Amazon EKS a Información de contenedores con observabilidad mejorada para Amazon EKS

1. Introduzca el siguiente comando para instalar el operador del agente de CloudWatch. En este comando, *my_cluster_name* es el nombre del clúster de Amazon EKS o el de Kubernetes y *my_cluster_region* es el nombre de la región en la que se publican los registros. Le

recomendamos que utilice la misma región en la que implemente el clúster para reducir los costos de transferencia de datos salientes de AWS.

```
ClusterName=my-cluster-name
RegionName=my-cluster-region
curl https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-
container-insights/main/k8s-quickstart/cwagent-operator-rendered.yaml | sed 's/
{{cluster_name}}/'${ClusterName}'/g;s/{{region_name}}/'${RegionName}'/g' | kubectl
 apply -f -
```

Si observa un error provocado por un conflicto de recursos, es probable que se deba a que ya tiene instalados en el clúster el agente de CloudWatch y Fluent Bit con sus componentes asociados, como ServiceAccount, ClusterRole y ClusterRoleBinding. Cuando el operador del agente de CloudWatch intenta instalar el agente de CloudWatch y sus componentes asociados, si detecta algún cambio en el contenido, por defecto no se realiza la instalación o la actualización para evitar sobrescribir el estado de los recursos del clúster. Le recomendamos que elimine cualquier agente de CloudWatch existente con la configuración de Información de contenedores que haya instalado anteriormente en el clúster y, a continuación, instale el operador del agente de CloudWatch.

2. (Opcional) Para aplicar una configuración de Fluent Bit personalizada existente, debe actualizar el mapa de configuración asociado al daemonset de Fluent Bit. El operador del agente de CloudWatch proporciona una configuración predeterminada para Fluent Bit y usted puede anular o modificar la configuración predeterminada según sea necesario. Para aplicar una configuración personalizada, siga estos pasos.

   a. Introduzca el siguiente comando para abrir la configuración existente.

   ```
   kubectl edit cm fluent-bit-config -n amazon-cloudwatch
   ```

   b. Realice los cambios en el archivo y, a continuación, introduzca `:wq` para guardarlo y salir del modo de edición.

   c. Introduzca el siguiente comando para reiniciar Fluent Bit.

   ```
   kubectl rollout restart fluent-bit -n amazon-cloudwatch
   ```

Actualización de la imagen del contenedor del agente de CloudWatch

> **Important**
>
> Si va a actualizar o instalar Información de contenedores en un clúster de Amazon EKS, recomendamos que utilice el complemento de EKS de observabilidad de Amazon CloudWatch para la instalación, en lugar de seguir las instrucciones de esta sección. Además, para recuperar las métricas de computación acelerada, debe usar el complemento de EKS de observabilidad de Amazon CloudWatch o el operador del agente de CloudWatch. Para obtener más información e instrucciones, consulte Instalación del complemento de observabilidad de EKS de Amazon CloudWatch.

Si necesita actualizar la imagen de contenedor a la versión más reciente, siga los pasos de esta sección.

Para actualizar la imagen de contenedor, realice el siguiente procedimiento:

1. Introduzca el siguiente comando para comprobar la definición del recurso personalizado (CRD) de `amazoncloudwatchagent`.

   ```
   kubectl get crds amazoncloudwatchagents.cloudwatch.aws.amazon.com -n amazon-cloudwatch
   ```

   Si este comando devuelve un error que indica que falta el CRD, el clúster no tiene Información de contenedores con observabilidad mejorada para Amazon EKS configurado con el operador del agente de CloudWatch. En este caso, consulte Actualización a Información de contenedores con observabilidad mejorada para Amazon EKS.

2. Especifique el siguiente comando para aplicar el archivo `cwagent-version.yaml` más reciente.

   ```
   curl https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-insights/main/k8s-quickstart/cwagent-version.yaml | kubectl apply -f -
   ```

Eliminación del agente de CloudWatch y Fluen Bit para Información de contenedores

Si instaló Información de contenedores mediante la instalación del complemento observabilidad de CloudWatch para Amazon EKS, puede eliminar Información de contenedores y el agente de CloudWatch introduciendo el siguiente comando:

> **Note**
>
> El complemento de Amazon EKS ahora es compatible con Información de contenedores en los nodos de trabajo de Windows. Si elimina el complemento de Amazon EKS, también se eliminará Información de contenedores para Windows.

```
aws eks delete-addon –cluster-name my-cluster –addon-name amazon-cloudwatch-
observability
```

Para eliminar todos los recursos relacionados con el agente de CloudWatch y Fluent Bit, introduzca el siguiente comando. En este comando, *My_Cluster_Name* es el nombre del clúster de Amazon EKS o el de Kubernetes y *My_Region* es el nombre de la región en la que se publican los registros.

```
ClusterName=My_Cluster_Name
RegionName=My-Region
curl https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-
container-insights/main/k8s-quickstart/cwagent-operator-rendered.yaml | sed 's/
{{cluster_name}}/'${ClusterName}'/g;s/{{region_name}}/'${RegionName}'/g' | kubectl
 delete -f -
curl https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
insights/main/k8s-quickstart/cwagent-custom-resource-definitions.yaml | kubectl delete
 -f -
```

Visualización de las métricas de Información de contenedores

Una vez que haya configurado Información de contenedores y esté recopilando las métricas, podrá ver esas métricas en la consola de CloudWatch.

Para que las métricas de Información de contenedores aparezcan en el panel, debe completar la configuración de Información de contenedores. Para obtener más información, consulte Configuración de Información de contenedores.

Este procedimiento explica cómo se pueden ver las métricas que Información de contenedores genera automáticamente a partir de los datos de registro recopilados. En el resto de esta sección se explica cómo profundizar en los datos y utilizar CloudWatch Logs Insights para ver más métricas en más niveles de granularidad.

Cómo ver las métricas de Información de contenedores

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Información y, luego, Información de contenedores.
3. En el cuadro desplegable situado debajo de Información de contenedores, seleccione Supervisión del rendimiento.
4. Utilice los cuadros desplegables situados junto a la parte superior para seleccionar el tipo de recurso que desea ver, así como el recurso específico.

Se puede establecer una alarma de CloudWatch en cualquiera de las métricas que recopila Información de contenedores. Para obtener más información, consulte Uso de las alarmas de Amazon CloudWatch

> **Note**
>
> Si ya ha configurado Información de aplicaciones de CloudWatch para supervisar las aplicaciones en contenedores, el panel de Información de aplicaciones aparece debajo del panel de Información de contenedores. Si aún no ha habilitado Información de aplicaciones, puede hacerlo si selecciona Configuración automática de Información de aplicaciones debajo de la vista de rendimiento en el panel de Información de contenedores.
> Para obtener más información acerca de Información de aplicaciones y aplicaciones en contenedores, consulte Habilitación del monitoreo de recursos de Información de aplicaciones para Amazon ECS y Amazon EKS.

Visualización de los colaboradores principales

Para algunas de las vistas de la supervisión del rendimiento de Información de contenedores, también puede ver los colaboradores principales por memoria o CPU, o los recursos activos más recientes. Está disponible cuando selecciona cualquiera de los siguientes paneles en el cuadro desplegable situado cerca de la parte superior de la página:

- Servicios de ECS
- Tareas de ECS
- Espacios de nombres de EKS
- Servicios de EKS
- Pods de EKS

Cuando está visualizando uno de estos tipos de recursos, la parte inferior de la página muestra una tabla ordenada inicialmente por el uso de la CPU. Puede cambiarlo para ordenarlo por uso de la memoria o por la actividad reciente. Para ver más detalles acerca de una de las filas de la tabla, puede seleccionar la casilla de verificación situada junto a esa fila y, a continuación, elegir Actions (Acciones) y elija una de las opciones en el menú Actions (Acciones).

Uso de Información de registros de CloudWatch para ver datos de Información de contenedores

Información de contenedores recopila métricas mediante los eventos de registro de rendimiento con el uso del embedded metric format (formato de métricas integradas). Los registros se almacenan en CloudWatch Logs. CloudWatch genera varias métricas automáticamente a partir de los registros que se pueden ver en la consola de CloudWatch. También puede realizarse un análisis más profundo de los datos de rendimiento que se recopilan mediante las consultas de CloudWatch Logs Insights.

Para obtener más información sobre CloudWatch Logs Insights, consulte Analyze Log Data with CloudWatch Logs Insights (Análisis de los datos de registro con CloudWatch Logs Insights). Para obtener más información sobre los campos de los registros puede utilizar en las consultas, consulte Eventos de registro de rendimiento de Información de contenedores para Amazon EKS y Kubernetes.

Para utilizar CloudWatch Logs Insights para consultar los datos de las métricas de contenedores

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Información.

   Cerca de la parte superior de la pantalla se encuentra el editor de consultas. La primera vez que se abre CloudWatch Logs Insights, este cuadro contiene una consulta predeterminada que muestra los 20 eventos del registro más recientes.

3. En el cuadro situado encima del editor de consultas, seleccione uno de los grupos de registros de Información de contenedores que desee consultar. Para que funcionen las siguientes consultas de ejemplo, el nombre del grupo de registros debe terminar por performance.

   Al seleccionar un grupo de registros, CloudWatch Logs Insights detecta automáticamente los campos de los datos en el grupo de registros y los muestra en Discovered fields (Campos detectados) en el panel derecho. También muestra un gráfico de barras de eventos de registro en este grupo de registro con el paso del tiempo. Este gráfico de barras muestra la distribución de los eventos en el grupo de registros que coincide con la consulta y el intervalo de tiempo, no solo los eventos que se muestran en la tabla.

4. En el editor de consultas, sustituya la consulta predeterminada por la consulta siguiente y elija Run query (Ejecutar consulta).

   ```
   STATS avg(node_cpu_utilization) as avg_node_cpu_utilization by NodeName
   | SORT avg_node_cpu_utilization DESC
   ```

   Esta consulta muestra una lista de nodos, ordenada por el promedio de utilización de la CPU del nodo.

5. Para probar otro ejemplo, sustituya esa consulta por otra y elija Run query (Ejecutar consulta). Se enumerarán más consultas de ejemplo más adelante en esta página.

   ```
   STATS avg(number_of_container_restarts) as avg_number_of_container_restarts by
    PodName
   | SORT avg_number_of_container_restarts DESC
   ```

   Esta consulta muestra una lista de pods ordenada por el número medio de reinicios del contenedor.

6. Si desea probar otra consulta, puede incluir campos de la lista situada a la derecha de la pantalla. Para obtener más información sobre la sintaxis de consulta, consulte CloudWatch Logs Insights Query Syntax (Sintaxis de consulta de CloudWatch Logs Insights).

Para ver las listas de recursos

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Recursos.

3. La vista predeterminada contiene los recursos que Información de contenedores supervisa y las alarmas que se han configurado en estos recursos. Para ver un mapa visual de los recursos, elija Vista de mapa.
4. En la vista de mapa, puede detener el puntero sobre cualquier recurso del mapa para ver sus métricas básicas. Elija cualquier recurso para ver gráficos más detallados sobre él.

Caso de uso: Visualización de métricas de nivel de tarea en contenedores de Amazon ECS

En el siguiente ejemplo se ilustra cómo se utiliza Información de registros de CloudWatch para profundizar en los registros de Información de contenedores. Para obtener más ejemplos, consulte el blog Presentación de Información de contenedores de Amazon CloudWatch para Amazon ECS).

Información de contenedores no genera automáticamente métricas en el nivel de granularidad de la tarea. La siguiente consulta muestra métricas de nivel de tarea para el uso de la CPU y de la memoria.

```
stats avg(CpuUtilized) as CPU, avg(MemoryUtilized) as Mem by TaskId, ContainerName
| sort Mem, CPU desc
```

Otras consultas de muestra para Información de contenedores

Lista de pods ordenada por el número medio de reinicios del contenedor

```
STATS avg(number_of_container_restarts) as avg_number_of_container_restarts by PodName
| SORT avg_number_of_container_restarts DESC
```

Pods solicitados frente a pods en ejecución

```
fields @timestamp, @message
| sort @timestamp desc
| filter Type="Pod"
| stats min(pod_number_of_containers) as requested,
 min(pod_number_of_running_containers) as running, ceil(avg(pod_number_of_containers-
pod_number_of_running_containers)) as pods_missing by kubernetes.pod_name
| sort pods_missing desc
```

Número de errores de nodos del clúster

```
stats avg(cluster_failed_node_count) as CountOfNodeFailures
| filter Type="Cluster"
| sort @timestamp desc
```

Errores de registro de aplicaciones por nombre de contenedor

```
stats count() as countoferrors by kubernetes.container_name
| filter stream="stderr"
| sort countoferrors desc
```

Métricas que Información de contenedores recopila

Información de contenedores recopila un conjunto de métricas para Amazon ECS y AWS Fargate en Amazon ECS y un conjunto diferente para Amazon EKS, AWS Fargate en Amazon EKS y Kubernetes.

Las métricas no están visibles hasta que las tareas del contenedor hayan estado ejecutándose durante algún tiempo.

Temas

- Métricas de Información de contenedores de Amazon ECS
- Métricas de Información de contenedores de Kubernetes y de Amazon EKS

Métricas de Información de contenedores de Amazon ECS

En la siguiente tabla, se muestran las métricas y dimensiones que recopila Información de contenedores para Amazon ECS. Estas métricas se encuentran en el espacio de nombres `ECS/ContainerInsights`. Para obtener más información, consulte Métricas.

Si no ve ninguna métrica de Información de contenedores en la consola, asegúrese de haber completado la configuración de Información de contenedores. Las métricas no aparecen antes de haber configurado por completo Información de contenedores. Para obtener más información, consulte Configuración de Información de contenedores.

Las siguientes métricas están disponibles al completar los pasos en Configuración de Información de contenedores en Amazon ECS para métricas de nivel de clúster y de nivel de servicio

Nombre de métrica	Dimensiones	Descripción
`ContainerInstanceCount`	ClusterName	El número de instancia s EC2 que ejecutan el agente de Amazon ECS que están registradas en un clúster. Esta métrica se recopila únicamente para las instancias de contenedo r que ejecutan tareas de Amazon ECS en el clúster. No se recopila para las instancias de contenedor vacías que no tienen ninguna tarea de Amazon ECS. Unidad: recuento

`CpuUtilized`	`TaskDefinitionFami ly` , `ClusterName`
	
	`ServiceName` , `ClusterName`
	
	`ClusterName`
		Las unidades de CPU utilizadas por tareas en el recurso especificado por el conjunto de dimension es que está utilizando.
		
		Esta métrica se recopila únicamente para las tareas que tienen una reserva de CPU definida en su definición de tarea.
		
		Unidad: ninguna

`CpuReserved`	`TaskDefinitionFami ly` , `ClusterName`	Las unidades de CPU reservadas por las tareas en el recurso especific

Nombre de métrica	Dimensiones	Descripción

	`ServiceName` , `ClusterName`
	
	`ClusterName`
		ado por el conjunto de dimensiones que está utilizando.
		
		Esta métrica se recopila únicamente para las tareas que tienen una reserva de CPU definida en su definición de tarea.
		
		Unidad: ninguna

`DeploymentCount`	`ServiceName` , `ClusterName`	El número de implement aciones en un servicio Amazon ECS. Unidad: recuento
`DesiredTaskCount`	`ServiceName` , `ClusterName`	El número deseado de tareas para un servicio Amazon ECS. Unidad: recuento

Nombre de métrica	Dimensiones	Descripción
EBSFilesystemSize	VolumeName , TaskDefinitionFamily , ClusterName TaskDefinitionFamily , ClusterName ServiceName , ClusterName	La cantidad total, en gigabytes (GB), del almacenamiento del sistema de archivos de Amazon EBS que se asigna a los recursos especificados por las dimensiones que está utilizando. Esta métrica solo está disponible para tareas que utilicen la infraestructura de Amazon ECS en Fargate mediante la versión de la plataforma a 1.4.0 o posterior o instancias de Amazon EC2 que utilicen la versión de agente de contenedor 1.79.0 o posterior. Unidad: Gigabytes (GB)

Nombre de métrica	Dimensiones	Descripción
EBSFilesystemUtilized	VolumeName , TaskDefinitionFamily , ClusterName TaskDefinitionFamily , ClusterName ServiceName , ClusterName	La cantidad total, en gigabytes (GB), del almacenamiento del sistema de archivos de Amazon EBS que utilizan los recursos especificados según las dimensiones que está utilizando. Esta métrica solo está disponible para tareas que utilicen la infraestructura de Amazon ECS en Fargate mediante la versión de la plataforma a 1.4.0 o posterior o instancias de Amazon EC2 que utilicen la versión de agente de contenedor 1.79.0 o posterior. Para las tareas que se ejecutan en Fargate, Fargate reserva espacio en el disco que solo utiliza Fargate. El espacio que usa Fargate no tiene ningún costo, pero verá este almacenamiento adicional con herramientas como df. Unidad: Gigabytes (GB)

Nombre de métrica	Dimensiones	Descripción
EphemeralStorageReserved [1]	TaskDefinitionFamily , ClusterName ServiceName , ClusterName ClusterName	El número de bytes reservados desde el almacenamiento efímero en el recurso que se especifica mediante las dimensiones que está utilizando. El almacenamiento efímero se utiliza para el sistema de archivos raíz del contenedor y para cualquier volumen host de montaje enlazado definido en la imagen del contenedor y la definición de la tarea. La cantidad de almacenamiento efímero no se puede cambiar en una tarea en ejecución. Esta métrica solo está disponible para tareas que utilicen la versión de la plataforma Fargate de Linux 1.4.0 o una posterior. Unidad: Gigabytes (GB)

Nombre de métrica	Dimensiones	Descripción

`EphemeralStorageUtilized` [1]	`TaskDefinitionFamily` , `ClusterName`
	
	`ServiceName` , `ClusterName`
	
	`ClusterName`
		El número de bytes que se usan desde el almacenamiento efímero en el recurso que se especifica mediante las dimensiones que está utilizando. El almacenamiento efímero se utiliza para el sistema de archivos raíz del contenedor y para cualquier volumen host de montaje enlazado definido en la imagen del contenedor y la definición de la tarea. La cantidad de almacenamiento efímero no se puede cambiar en una tarea en ejecución.
		
		Esta métrica solo está disponible para tareas que utilicen la versión de la plataforma Fargate de Linux 1.4.0 o una posterior.
		
		Unidad: Gigabytes (GB)

Nombre de métrica	Dimensiones	Descripción
MemoryUtilized	TaskDefinitionFamily , ClusterName ServiceName , ClusterName ClusterName	La memoria que están utilizando las tareas en el recurso especificado por el conjunto de dimensiones que está utilizando. Esta métrica se recopila únicamente para las tareas que tienen una reserva de memoria definida en su definición de tarea. Unidades: megabytes
MemoryReserved	TaskDefinitionFamily , ClusterName ServiceName , ClusterName ClusterName	La memoria reservada por las tareas en el recurso que se especifica mediante el conjunto de dimensiones que está utilizando. Esta métrica se recopila únicamente para las tareas que tienen una reserva de memoria definida en su definición de tarea. Unidades: megabytes

Nombre de métrica	Dimensiones	Descripción
NetworkRxBytes	TaskDefinitionFamily , ClusterName ServiceName , ClusterName ClusterName	El número de bytes recibidos por el recurso que se especifica mediante las dimensiones que está utilizando. Esta métrica se obtiene del tiempo de ejecución de Docker. Esta métrica solo está disponible para los contenedores en tareas que utilizan modos de redes awsvpc o bridge. Unidad: bytes/segundo
NetworkTxBytes	TaskDefinitionFamily , ClusterName ServiceName , ClusterName ClusterName	El número de bytes transmitidos por el recurso que se especifica mediante las dimensiones que está utilizando. Esta métrica se obtiene del tiempo de ejecución de Docker. Esta métrica solo está disponible para los contenedores en tareas que utilizan modos de redes awsvpc o bridge. Unidad: bytes/segundo

Nombre de métrica	Dimensiones	Descripción
PendingTaskCount	ServiceName , ClusterName	El número de tareas que actualmente tienen el estado PENDING. Unidad: recuento
RunningTaskCount	ServiceName , ClusterName	El número de tareas que actualmente tienen el estado RUNNING. Unidad: recuento
ServiceCount	ClusterName	El número de servicios en el clúster. Unidad: recuento
StorageReadBytes	TaskDefinitionFamily , ClusterName ServiceName , ClusterName ClusterName	La cantidad de bytes leídos desde el almacenamiento en la instancia en el recurso que se especifica mediante las dimensiones que está utilizando. Esto no incluye los bytes de lectura de sus dispositivos de almacenamiento. Esta métrica se obtiene del tiempo de ejecución de Docker. Unidades: bytes

Nombre de métrica	Dimensiones	Descripción

`StorageWriteBytes`	`TaskDefinitionFamily` , `ClusterName`
	
	`ServiceName` , `ClusterName`
	
	`ClusterName`
		El número de bytes escritos en almacenamiento en el recurso que se especifica mediante las dimensiones que está utilizando. Esta métrica se obtiene del tiempo de ejecución de Docker.
		
		Unidades: bytes

`TaskCount`	`ClusterName`	El número de tareas que se ejecutan en el clúster. Unidad: recuento
`TaskSetCount`	`ServiceName` , `ClusterName`	El número de conjuntos de tareas en el servicio. Unidad: recuento

> **Note**
>
> Las métricas `EphemeralStorageReserved` y `EphemeralStorageUtilized` solo están disponibles para tareas que se ejecuten en la versión 1.4.0 o una versión posterior de la plataforma Fargate de Linux.
> Fargate reserva espacio en el disco. Solo lo usa Fargate. No se cobra por esto. No se muestra en estas métricas. Sin embargo, puede ver este almacenamiento adicional en otras herramientas, como `df`.

Las siguientes métricas están disponibles al completar los pasos en Implementación del agente de CloudWatch para recopilar métricas de nivel de instancia EC2 en Amazon ECS

Nombre de métrica	Dimensiones	Descripción
instance_cpu_limit	ClusterName	El número máximo de unidades de CPU que se pueden asignar a una única instancia EC2 en el clúster. Unidad: ninguna
instance_cpu_reserved_capacity	ClusterName InstanceId , ContainerInstanceId , ClusterName	El porcentaje de CPU reservado actualmente en una única instancia EC2 en el clúster. Unidad: porcentaje
instance_cpu_usage_total	ClusterName	El número de unidades de CPU utilizadas en una única instancia EC2 en el clúster. Unidad: ninguna
instance_cpu_utilization	ClusterName InstanceId , ContainerInstanceId , ClusterName	El porcentaje total de unidades de CPU utilizadas en una única instancia EC2 en el clúster. Unidad: porcentaje
instance_filesystem_utilization	ClusterName InstanceId , ContainerInstanceId , ClusterName	El porcentaje total de capacidad del sistema de archivos utilizada en una única instancia EC2 en el clúster.

Nombre de métrica	Dimensiones	Descripción
		Unidad: porcentaje
instance_memory_limit	ClusterName	La cantidad máxima de memoria, en bytes, que se puede asignar a una única instancia EC2 en este clúster. Unidades: bytes
instance_memory_reserved_capacity	ClusterName InstanceId , ContainerInstanceId , ClusterName	El porcentaje de memoria reservada actualmente en una única instancia EC2 en el clúster. Unidad: porcentaje
instance_memory_utilization	ClusterName InstanceId , ContainerInstanceId , ClusterName	El porcentaje total de memoria utilizada en una única instancia EC2 en el clúster. Unidad: porcentaje
instance_memory_working_set	ClusterName	La cantidad de memoria, en bytes, utilizada en una única instancia EC2 en el clúster. Unidades: bytes

Nombre de métrica	Dimensiones	Descripción
`instance_network_total_bytes`	`ClusterName`	El número total de bytes por segundo transmitidos y recibidos a través de la red en una única instancia EC2 en el clúster. Unidad: bytes/segundo
`instance_number_of_running_tasks`	`ClusterName`	El número de tareas en ejecución en una única instancia EC2 en el clúster. Unidad: recuento

Métricas de Información de contenedores de Kubernetes y de Amazon EKS

En las siguientes tablas, se muestran las métricas y las dimensiones que recopila Información de contenedores para Amazon EKS y Kubernetes. Estas métricas se encuentran en el espacio de nombres `ContainerInsights`. Para obtener más información, consulte Métricas.

Si no ve ninguna métrica de Información de contenedores en la consola, asegúrese de haber completado la configuración de Información de contenedores. Las métricas no aparecen antes de haber configurado por completo Información de contenedores. Para obtener más información, consulte Configuración de Información de contenedores.

Si utiliza la versión 1.5.0 o posterior del complemento de Amazon EKS o la versión 1.300035.0 del agente CloudWatch, la mayoría de las métricas que se muestran en la siguiente tabla se recopilan para los nodos de Linux y Windows. Consulte la columna Nombre de métrica de la tabla para ver qué métricas no se recopilan para Windows.

Con la versión original de Información de contenedores, las métricas recopiladas se cobran como métricas personalizadas. Con Información de contenedores, con una observabilidad mejorada para

Amazon EKS, las métricas de Información de contenedores se cobran por observación en lugar de cobrarse por métrica almacenada o registro ingerido. Para obtener más información sobre los precios de CloudWatch, consulte Precios de Amazon CloudWatch.

> **Note**
>
> En Windows, las métricas de red como `pod_network_rx_bytes` y `pod_network_tx_bytes` no se recopilan para los contenedores de procesos del host.

Nombre de métrica	Dimensiones con cualquier versión de Información de contenedores	Dimensiones adicionales con Información de contenedores con observabilidad mejorada para Amazon EKS	Descripción
`cluster_failed_node_count`	`ClusterName`		El número de nodos de trabajo con errores en el clúster. Se considera que un nodo ha fallado si está sufriendo de cualquiera de las condiciones de nodo. Para obtener más información, consulte Conditions (Condiciones) en la documentación de Kubernetes.
`cluster_node_count`	`ClusterName`		El número total de nodos de trabajo en el clúster.

Nombre de métrica	Dimensiones con cualquier versión de Información de contenedores	Dimensiones adicionales con Información de contenedores con observabilidad mejorada para Amazon EKS	Descripción

`namespace_number_of_running_pods`	`Namespace` `ClusterName`
	
	`ClusterName`
			El número de pods que se ejecutan por espacio de nombres en el recurso que se especifica mediante las dimensiones que está utilizando.

`node_cpu_limit`	`ClusterName`	`ClusterName` , `InstanceId` , `NodeName`	El número máximo de unidades de CPU que se pueden asignar a un único nodo en este clúster.

Nombre de métrica	Dimensiones con cualquier versión de Información de contenedores	Dimensiones adicionales con Información de contenedores con observabilidad mejorada para Amazon EKS	Descripción

`node_cpu_reserved_capacity`	`NodeName`, `ClusterName` , `InstanceId`
	
	`ClusterName`
			El porcentaje de unidades de CPU que están reservadas para los componentes de nodos, como kubelet, kube-proxy y Docker.
			
			Fórmula: `node_cpu_request / node_cpu_limit`
			
			Note
			`node_cpu_request` no se informa directamente como una métrica, sino que es un campo en el registro de eventos del rendimiento. Para obtener más informaci ón, consulte Campos relevantes

Nombre de métrica	Dimensiones con cualquier versión de Información de contenedores	Dimensiones adicionales con Información de contenedores con observabilidad mejorada para Amazon EKS	Descripción
			en eventos de registro de rendimiento para Amazon EKS y Kubernetes.
`node_cpu_usage_total`	`ClusterName`	`ClusterName` , `InstanceId` , `NodeName`	El número de unidades de CPU que se utilizan en los nodos del clúster.

`node_cpu_utilization`	`NodeName`, `ClusterName` , `InstanceId`
	
	`ClusterName`
			El porcentaje total de unidades de CPU que se utilizan en los nodos del clúster.
			
			Fórmula: `node_cpu_usage_total / node_cpu_limit`

Nombre de métrica	Dimensiones con cualquier versión de Información de contenedores	Dimensiones adicionales con Información de contenedores con observabilidad mejorada para Amazon EKS	Descripción

`node_filesystem_utilization`	`NodeName`, `ClusterName` , `InstanceId`
	
	`ClusterName`
			El porcentaje total de capacidad de sistema de archivos que se utiliza en los nodos del clúster.
			
			Fórmula: `node_filesystem_usage / node_filesystem_capacity`
			
			Note
			`node_filesystem_usage` y `node_filesystem_capacity` no se informan directamente como métricas, sino que son campos en el registro de eventos del rendimiento.

Nombre de métrica	Dimensiones con cualquier versión de Información de contenedores	Dimensiones adicionales con Información de contenedores con observabilidad mejorada para Amazon EKS	Descripción
			Para obtener más información, consulte Campos relevantes en eventos de registro de rendimiento para Amazon EKS y Kubernetes.
`node_memory_limit`	`ClusterName`	`ClusterName` , `InstanceId` , `NodeName`	La cantidad máxima de memoria, en bytes, que se puede asignar a un único nodo en este clúster.

Nombre de métrica	Dimensiones con cualquier versión de Información de contenedores	Dimensiones adicionales con Información de contenedores con observabilidad mejorada para Amazon EKS	Descripción

`node_file system_inodes`
 	
Esta métrica solo está disponible con Información de contenedores con observabilidad mejorada para Amazon EKS. No está disponible en Windows.
		`ClusterName`
		
		`ClusterName` , `InstanceId` , `NodeName`
			La cantidad total de inodos (utilizados y no utilizados) en un nodo.

`node_file system_in odes_free`
 	
Esta métrica solo está disponible con Información de contenedores con observabilidad mejorada para Amazon EKS. No está disponible en Windows.
		`ClusterName`
		
		`ClusterName` , `InstanceId` , `NodeName`
			La cantidad total de inodos no utilizados en un nodo.

Nombre de métrica	Dimensiones con cualquier versión de Información de contenedores	Dimensiones adicionales con Información de contenedores con observabilidad mejorada para Amazon EKS	Descripción

`node_memory_reserved_capacity`	`NodeName`, `ClusterName` , `InstanceId`
	
	`ClusterName`
			El porcentaje de memoria que se utiliza actualmente en los nodos del clúster.
			
			Fórmula: `node_memory_request / node_memory_limit`
			
			Note
			`node_memory_request` no se informa directamente como una métrica, sino que es un campo en el registro de eventos del rendimiento. Para obtener más información, consulte Campos relevantes

Nombre de métrica	Dimensiones con cualquier versión de Información de contenedores	Dimensiones adicionales con Información de contenedores con observabilidad mejorada para Amazon EKS	Descripción
			en eventos de registro de rendimiento para Amazon EKS y Kubernetes.
node_memory_utilization	NodeName, ClusterName , InstanceId ClusterName		El porcentaje de memoria que utiliza actualmente el nodo o los nodos. Es el porcentaje de uso de memoria de nodo sobre la limitación de memoria de nodo. Fórmula: node_memory_working_set / node_memory_limit .
node_memory_working_set	ClusterName	ClusterName , InstanceId , NodeName	La cantidad de memoria, en bytes, que se utiliza en el conjunto de trabajo de los nodos del clúster.

Nombre de métrica	Dimensiones con cualquier versión de Información de contenedores	Dimensiones adicionales con Información de contenedores con observabilidad mejorada para Amazon EKS	Descripción

`node_network_total_bytes`	`NodeName`, `ClusterName` , `InstanceId`
	
	`ClusterName`
			El número total de bytes por segundo transmitidos y recibidos a través de la red por nodo en un clúster.
			
			Fórmula: `node_network_rx_bytes + node_network_tx_bytes`
			
			Note
			`node_network_rx_bytes` y `node_network_tx_bytes` no se informan directamente como métricas, sino que son campos en el registro de eventos del rendimiento.

Nombre de métrica	Dimensiones con cualquier versión de Información de contenedores	Dimensiones adicionales con Información de contenedores con observabilidad mejorada para Amazon EKS	Descripción
			Para obtener más información, consulte Campos relevantes en eventos de registro de rendimiento para Amazon EKS y Kubernetes.
node_number_of_running_containers	NodeName, ClusterName , InstanceId ClusterName		El número de contenedores en ejecución por nodo en un clúster.
node_number_of_running_pods	NodeName, ClusterName , InstanceId ClusterName		El número de pods en ejecución por nodo en un clúster.

Nombre de métrica	Dimensiones con cualquier versión de Información de contenedores	Dimensiones adicionales con Información de contenedores con observabilidad mejorada para Amazon EKS	Descripción

`node_status_allocatable_pods`
 	
Esta métrica solo está disponible con Información de contenedores con observabilidad mejorada para Amazon EKS
		`ClusterName`
		
		`ClusterName` , `InstanceId` , `NodeName`
			La cantidad de pods que se pueden asignar a un nodo en función de sus recursos asignables, que se define como el resto de la capacidad de un nodo después de tener en cuenta las reservas de daemons del sistema y los umbrales de expulsión forzoso.

`node_status_capacity_pods`
 	
Esta métrica solo está disponible con Información de contenedores con observabilidad mejorada para Amazon EKS
		`ClusterName`
		
		`ClusterName` , `InstanceId` , `NodeName`
			Cantidad de pods que se pueden asignar a un nodo en función de su capacidad.

Nombre de métrica	Dimensiones con cualquier versión de Información de contenedores	Dimensiones adicionales con Información de contenedores con observabilidad mejorada para Amazon EKS	Descripción

`node_status_condition_ready`
 	
Esta métrica solo está disponible con Información de contenedores con observabilidad mejorada para Amazon EKS
		`ClusterName`
		
		`ClusterName` , `InstanceId` , `NodeName`
			Indica si la condición de estado `Ready` del nodo es verdadera.

`node_status_condition_memory_pressure`
 	
Esta métrica solo está disponible con Información de contenedores con observabilidad mejorada para Amazon EKS
		`ClusterName`
		
		`ClusterName` , `InstanceId` , `NodeName`
			Indica si la condición de estado `MemoryPressure` del nodo es verdadera .

Nombre de métrica	Dimensiones con cualquier versión de Información de contenedores	Dimensiones adicionales con Información de contenedores con observabilidad mejorada para Amazon EKS	Descripción

`node_status_condition_pid_pressure`
 	
Esta métrica solo está disponible con Información de contenedores con observabilidad mejorada para Amazon EKS
		`ClusterName`
		
		`ClusterName` , `InstanceId` , `NodeName`
			Indica si la condición de estado `PIDPressure` del nodo es verdadera.

`node_status_condition_disk_pressure`
 	
Esta métrica solo está disponible con Información de contenedores con observabilidad mejorada para Amazon EKS
		`ClusterName`
		
		`ClusterName` , `InstanceId` , `NodeName`
			Indica si la condición de estado `OutOfDisk` del nodo es verdadera.

Nombre de métrica	Dimensiones con cualquier versión de Información de contenedores	Dimensiones adicionales con Información de contenedores con observabilidad mejorada para Amazon EKS	Descripción

`node_status_condition_unknown`
 	
Esta métrica solo está disponible con Información de contenedores con observabilidad mejorada para Amazon EKS
		`ClusterName`
		
		`ClusterName` , `InstanceId` , `NodeName`
			Indica si alguna de las condiciones de estado del nodo es Desconocida.

Nombre de métrica	Dimensiones con cualquier versión de Información de contenedores	Dimensiones adicionales con Información de contenedores con observabilidad mejorada para Amazon EKS	Descripción

`node_interface_network_rx_dropped`
 	
Esta métrica solo está disponible con Información de contenedores con observabilidad mejorada para Amazon EKS
 	
Esta métrica solo está disponible con Información de contenedores con observabilidad mejorada para Amazon EKS
		`ClusterName`
		
		`ClusterName` , `InstanceId` , `NodeName`
			La cantidad de paquetes que una interfaz de red del nodo recibió y luego descartó.

Nombre de métrica	Dimensiones con cualquier versión de Información de contenedores	Dimensiones adicionales con Información de contenedores con observabilidad mejorada para Amazon EKS	Descripción

`node_interface_network_tx_dropped`
 	
Esta métrica solo está disponible con Información de contenedores con observabilidad mejorada para Amazon EKS
		`ClusterName`
		
		`ClusterName` , `InstanceId` , `NodeName`
			La cantidad de paquetes que debían transmitirse pero que una interfaz de red del nodo descartó.

`node_diskio_io_service_bytes_total`
 	
Esta métrica solo está disponible con Información de contenedores con observabilidad mejorada para Amazon EKS. No está disponible en Windows.
		`ClusterName`
		
		`ClusterName` , `InstanceId` , `NodeName`
			La cantidad total de bytes transferidos por todas las operaciones de E/S del nodo.

Nombre de métrica	Dimensiones con cualquier versión de Información de contenedores	Dimensiones adicionales con Información de contenedores con observabilidad mejorada para Amazon EKS	Descripción

`node_diskio_io_serviced_total`
 	
Esta métrica solo está disponible con Información de contenedores con observabilidad mejorada para Amazon EKS. No está disponible en Windows.
		`ClusterName`
		
		`ClusterName` , `InstanceId` , `NodeName`
			La cantidad total de operaciones de E/S del nodo.

Nombre de métrica	Dimensiones con cualquier versión de Información de contenedores	Dimensiones adicionales con Información de contenedores con observabilidad mejorada para Amazon EKS	Descripción

`pod_cpu_reserved_capacity`	`PodName`, Espacio de nombres, `ClusterName`
	
	`ClusterName`
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`
		
		`ClusterName` , `Namespace` , `Service`
			La capacidad de la CPU reservada por pod en un clúster.
			
			Fórmula: `pod_cpu_request / node_cpu_limit`
			
			Note
			`pod_cpu_request` no se informa directamente como una métrica, sino que es un campo en el registro de eventos del rendimiento. Para obtener más información, consulte Campos relevantes en eventos de registro de rendimien

Nombre de métrica	Dimensiones con cualquier versión de Información de contenedores	Dimensiones adicionales con Información de contenedores con observabilidad mejorada para Amazon EKS	Descripción
			to para Amazon EKS y Kubernetes.

Nombre de métrica	Dimensiones con cualquier versión de Información de contenedores	Dimensiones adicionales con Información de contenedores con observabilidad mejorada para Amazon EKS	Descripción

`pod_cpu_utilization`	`PodName`, Espacio de nombres, `ClusterName`
	
	Espacio de nombres, `ClusterName`
	
	Servicios, Espacio de nombres, `ClusterName`
	
	`ClusterName`
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`	El porcentaje de unidades de CPU que utilizan los pods.
			
			Fórmula: `pod_cpu_usage_total / node_cpu_limit`
			
			Note
			`pod_cpu_usage_total` no se informa directamente como una métrica, sino que es un campo en el registro de eventos del rendimiento. Para obtener más información, consulte Campos relevantes en eventos de registro

Nombre de métrica	Dimensiones con cualquier versión de Información de contenedores	Dimensiones adicionales con Información de contenedores con observabilidad mejorada para Amazon EKS	Descripción
			de rendimiento para Amazon EKS y Kubernetes.

Nombre de métrica	Dimensiones con cualquier versión de Información de contenedores	Dimensiones adicionales con Información de contenedores con observabilidad mejorada para Amazon EKS	Descripción
`pod_cpu_utilization_over_pod_limit`	PodName, Espacio de nombres, `ClusterName` Espacio de nombres, `ClusterName` Servicios, Espacio de nombres, `ClusterName` `ClusterName`	`ClusterName` , `Namespace` , PodName, `FullPodName`	El porcentaje de unidades de CPU utilizadas por pods en relación con el límite de pods. Fórmula: `pod_cpu_usage_total / pod_cpu_limit` **Note** `pod_cpu_usage_total` y `pod_cpu_limit` no se informan directamente como métricas, sino que son campos en el registro de eventos del rendimiento. Para obtener más información, consulte

Nombre de métrica	Dimensiones con cualquier versión de Información de contenedores	Dimensiones adicionales con Información de contenedores con observabilidad mejorada para Amazon EKS	Descripción
			Campos relevantes en eventos de registro de rendimiento para Amazon EKS y Kubernetes.

Nombre de métrica	Dimensiones con cualquier versión de Información de contenedores	Dimensiones adicionales con Información de contenedores con observabilidad mejorada para Amazon EKS	Descripción

`pod_memory_reserved_capacity`	`PodName`, Espacio de nombres, `ClusterName`
	
	`ClusterName`
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`
		
		`ClusterName` , `Namespace` , `Service`
			El porcentaje de memoria reservada para los pods.
			
			Fórmula: `pod_memory_request / node_memory_limit`
			
			Note
			`pod_memory_request` no se informa directamente como una métrica, sino que es un campo en el registro de eventos del rendimiento. Para obtener más información, consulte Campos relevantes en eventos

Nombre de métrica	Dimensiones con cualquier versión de Información de contenedores	Dimensiones adicionales con Información de contenedores con observabilidad mejorada para Amazon EKS	Descripción
			de registro de rendimiento para Amazon EKS y Kubernetes.

Nombre de métrica	Dimensiones con cualquier versión de Información de contenedores	Dimensiones adicionales con Información de contenedores con observabilidad mejorada para Amazon EKS	Descripción

`pod_memory_utilization`	`PodName`, Espacio de nombres, `ClusterName`
	
	Espacio de nombres, `ClusterName`
	
	Servicios, Espacio de nombres, `ClusterName`
	
	`ClusterName`
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`	El porcentaje de memoria que utiliza actualmente el pod o los pods.
			
			Fórmula: `pod_memory_working_set / node_memory_limit`
			
			Note
			`pod_memory_working_set` no se informa directamente como una métrica, sino que es un campo en el registro de eventos del rendimiento. Para obtener más información, consulte Campos

Nombre de métrica	Dimensiones con cualquier versión de Información de contenedores	Dimensiones adicionales con Información de contenedores con observabilidad mejorada para Amazon EKS	Descripción
			relevantes en eventos de registro de rendimiento para Amazon EKS y Kubernetes.

Nombre de métrica	Dimensiones con cualquier versión de Información de contenedores	Dimensiones adicionales con Información de contenedores con observabilidad mejorada para Amazon EKS	Descripción

`pod_memory_utilization_over_pod_limit`	`PodName`, Espacio de nombres, `ClusterName`
	
	Espacio de nombres, `ClusterName`
	
	Servicios, Espacio de nombres, `ClusterName`
	
	`ClusterName`
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`	El porcentaje de memoria utilizada por los pods en relación con el límite de pods. Si algún contenedor del pod no tiene definido un límite de memoria, esta métrica no aparecerá.
			
			Fórmula: `pod_memory_working_set / pod_memory_limit`
			
			Note
			`pod_memory_working_set` no se informa directamente como una métrica, sino que es un campo en el registro de eventos del

Nombre de métrica	Dimensiones con cualquier versión de Información de contenedores	Dimensiones adicionales con Información de contenedores con observabilidad mejorada para Amazon EKS	Descripción
			rendimiento. Para obtener más información, consulte Campos relevantes en eventos de registro de rendimiento para Amazon EKS y Kubernetes.

Nombre de métrica	Dimensiones con cualquier versión de Información de contenedores	Dimensiones adicionales con Información de contenedores con observabilidad mejorada para Amazon EKS	Descripción

`pod_network_rx_bytes`	`PodName`, Espacio de nombres, `ClusterName`
	
	Espacio de nombres, `ClusterName`
	
	Servicios, Espacio de nombres, `ClusterName`
	
	`ClusterName`
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`	El número de bytes por segundo que se están recibiendo a través de la red por el pod.
			
			Fórmula: `sum(pod_interface_network_rx_bytes)`
			
			Note
			`pod_interface_network_rx_bytes` no se informa directamente como una métrica, sino que es un campo en el registro de eventos del rendimiento. Para obtener más informaci ón, consulte

Nombre de métrica	Dimensiones con cualquier versión de Información de contenedores	Dimensiones adicionales con Información de contenedores con observabilidad mejorada para Amazon EKS	Descripción
			Campos relevantes en eventos de registro de rendimiento para Amazon EKS y Kubernetes.

Nombre de métrica	Dimensiones con cualquier versión de Información de contenedores	Dimensiones adicionales con Información de contenedores con observabilidad mejorada para Amazon EKS	Descripción

`pod_network_tx_bytes`	`PodName`, Espacio de nombres, `ClusterName`
	
	Espacio de nombres, `ClusterName`
	
	Servicios, Espacio de nombres, `ClusterName`
	
	`ClusterName`
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`	El número de bytes por segundo que se están transmitiendo a través de la red por el pod.
			
			Fórmula: `sum(pod_interface_network_tx_bytes)`
			
			Note
			`pod_interface_network_tx_bytes` no se informa directamente como una métrica, sino que es un campo en el registro de eventos del rendimiento. Para obtener más informaci ón, consulte

Nombre de métrica	Dimensiones con cualquier versión de Información de contenedores	Dimensiones adicionales con Información de contenedores con observabilidad mejorada para Amazon EKS	Descripción
			Campos relevantes en eventos de registro de rendimiento para Amazon EKS y Kubernetes.

Nombre de métrica	Dimensiones con cualquier versión de Información de contenedores	Dimensiones adicionales con Información de contenedores con observabilidad mejorada para Amazon EKS	Descripción

`pod_cpu_request`
 	
Esta métrica solo está disponible con Información de contenedores con observabilidad mejorada para Amazon EKS
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName`
		
		`Namespace` , `ClusterName` , `Service`
		
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`
			Las solicitudes de la CPU para el pod.
			
			Fórmula: `sum(container_cpu_request)`
			
			Note
			`pod_cpu_request` no se informa directamente como una métrica, sino que es un campo en el registro de eventos del rendimiento. Para obtener más información, consulte Campos relevantes en eventos de registro de rendimiento para

Nombre de métrica	Dimensiones con cualquier versión de Información de contenedores	Dimensiones adicionales con Información de contenedores con observabilidad mejorada para Amazon EKS	Descripción
			Amazon EKS y Kubernetes.

Nombre de métrica	Dimensiones con cualquier versión de Información de contenedores	Dimensiones adicionales con Información de contenedores con observabilidad mejorada para Amazon EKS	Descripción

`pod_memory_request`
 	
Esta métrica solo está disponible con Información de contenedores con observabilidad mejorada para Amazon EKS
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName`
		
		`Namespace` , `ClusterName` , `Service`
		
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`
			Las solicitudes de memoria para el pod.
			
			Fórmula: `sum(container_memory_request)`
			
			Note
			`pod_memory_request` no se informa directamente como una métrica, sino que es un campo en el registro de eventos del rendimiento. Para obtener más información, consulte Campos relevantes en eventos de registro de rendimien

Nombre de métrica	Dimensiones con cualquier versión de Información de contenedores	Dimensiones adicionales con Información de contenedores con observabilidad mejorada para Amazon EKS	Descripción
			to para Amazon EKS y Kubernetes.

Nombre de métrica	Dimensiones con cualquier versión de Información de contenedores	Dimensiones adicionales con Información de contenedores con observabilidad mejorada para Amazon EKS	Descripción

`pod_cpu_limit`
 	
Esta métrica solo está disponible con Información de contenedores con observabilidad mejorada para Amazon EKS
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName`
		
		`Namespace` , `ClusterName` , `Service`
		
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`
			El límite de la CPU definido para los contenedores del pod. Si algún contenedor del pod no tiene definido un límite de la CPU, esta métrica no aparecerá.
			
			Fórmula: `sum(container_cpu_limit)`
			
			Note
			`pod_cpu_limit` no se informa directamente como una métrica, sino que es un campo en el registro de eventos del rendimiento. Para obtener más información, consulte

Nombre de métrica	Dimensiones con cualquier versión de Información de contenedores	Dimensiones adicionales con Información de contenedores con observabilidad mejorada para Amazon EKS	Descripción
			Campos relevantes en eventos de registro de rendimiento para Amazon EKS y Kubernetes.

Nombre de métrica	Dimensiones con cualquier versión de Información de contenedores	Dimensiones adicionales con Información de contenedores con observabilidad mejorada para Amazon EKS	Descripción

`pod_memory_limit`
 	
Esta métrica solo está disponible con Información de contenedores con observabilidad mejorada para Amazon EKS
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName`
		
		`Namespace` , `ClusterName` , `Service`
		
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`
			El límite de memoria definido para los contenedores del pod. Si algún contenedor del pod no tiene definido un límite de memoria, esta métrica no aparecerá.
			
			Fórmula: `sum(container_memory_limit)`
			
			Note
			`pod_cpu_limit` no se informa directamente como una métrica, sino que es un campo en el registro de eventos del rendimiento. Para obtener más información, consulte

Nombre de métrica	Dimensiones con cualquier versión de Información de contenedores	Dimensiones adicionales con Información de contenedores con observabilidad mejorada para Amazon EKS	Descripción
			Campos relevantes en eventos de registro de rendimiento para Amazon EKS y Kubernetes.

`pod_status_failed`
 	
Esta métrica solo está disponible con Información de contenedores con observabilidad mejorada para Amazon EKS
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName`
		
		`Namespace` , `ClusterName` , `Service`
		
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`
			Indica que todos los contenedores del pod terminaron y que al menos un contenedor terminó con un estado distinto de cero o lo canceló el sistema.

Nombre de métrica	Dimensiones con cualquier versión de Información de contenedores	Dimensiones adicionales con Información de contenedores con observabilidad mejorada para Amazon EKS	Descripción

`pod_status_ready`
 	
Esta métrica solo está disponible con Información de contenedores con observabilidad mejorada para Amazon EKS
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName`
		
		`Namespace` , `ClusterName` , `Service`
		
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`
			Indica que todos los contenedores del pod están listos y alcanzaron el estado de `Container Ready` .

`pod_status_running`
 	
Esta métrica solo está disponible con Información de contenedores con observabilidad mejorada para Amazon EKS
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName`
		
		`Namespace` , `ClusterName` , `Service`
		
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`
			Indica que todos los contenedores del pod están en ejecución.

Nombre de métrica	Dimensiones con cualquier versión de Información de contenedores	Dimensiones adicionales con Información de contenedores con observabilidad mejorada para Amazon EKS	Descripción

`pod_status_scheduled`
 	
Esta métrica solo está disponible con Información de contenedores con observabilidad mejorada para Amazon EKS
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName`
		
		`Namespace` , `ClusterName` , `Service`
		
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`
			Indica que el pod se programó para un nodo.

`pod_status_unknown`
 	
Esta métrica solo está disponible con Información de contenedores con observabilidad mejorada para Amazon EKS
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName`
		
		`Namespace` , `ClusterName` , `Service`
		
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`
			Indica que no se puede obtener el estado del pod.

Nombre de métrica	Dimensiones con cualquier versión de Información de contenedores	Dimensiones adicionales con Información de contenedores con observabilidad mejorada para Amazon EKS	Descripción

`pod_status_pending`
 	
Esta métrica solo está disponible con Información de contenedores con observabilidad mejorada para Amazon EKS
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName`
		
		`Namespace` , `ClusterName` , `Service`
		
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`
			Indica que el clúster aceptó el pod, pero que uno o más de los contenedores aún no están listos.

`pod_status_succeeded`
 	
Esta métrica solo está disponible con Información de contenedores con observabilidad mejorada para Amazon EKS
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName`
		
		`Namespace` , `ClusterName` , `Service`
		
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`
			Indica que todos los contenedores del pod terminaron correctamente y no se reiniciarán.

Nombre de métrica	Dimensiones con cualquier versión de Información de contenedores	Dimensiones adicionales con Información de contenedores con observabilidad mejorada para Amazon EKS	Descripción

`pod_number_of_containers`
 	
Esta métrica solo está disponible con Información de contenedores con observabilidad mejorada para Amazon EKS
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName`
		
		`Namespace` , `ClusterName` , `Service`
		
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`
			Indica la cantidad de contenedores definido en la especificación del pod.

`pod_number_of_running_containers`
 	
Esta métrica solo está disponible con Información de contenedores con observabilidad mejorada para Amazon EKS
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName`
		
		`Namespace` , `ClusterName` , `Service`
		
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`
			Indica la cantidad de contenedores del pod que se encuentran actualmente en el estado `Running`.

Nombre de métrica	Dimensiones con cualquier versión de Información de contenedores	Dimensiones adicionales con Información de contenedores con observabilidad mejorada para Amazon EKS	Descripción

`pod_container_status_terminated`
 	
Esta métrica solo está disponible con Información de contenedores con observabilidad mejorada para Amazon EKS
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName`
		
		`Namespace` , `ClusterName` , `Service`
		
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`
			Indica la cantidad de contenedores del pod que se encuentran en el estado `Terminated` .

`pod_container_status_running`
 	
Esta métrica solo está disponible con Información de contenedores con observabilidad mejorada para Amazon EKS
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName`
		
		`Namespace` , `ClusterName` , `Service`
		
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`
			Indica la cantidad de contenedores del pod que se encuentran en el estado `Running`.

Nombre de métrica	Dimensiones con cualquier versión de Información de contenedores	Dimensiones adicionales con Información de contenedores con observabilidad mejorada para Amazon EKS	Descripción

`pod_container_status_waiting`
 	
Esta métrica solo está disponible con Información de contenedores con observabilidad mejorada para Amazon EKS
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName`
		
		`Namespace` , `ClusterName` , `Service`
		
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`
			Indica la cantidad de contenedores del pod que se encuentran en el estado `Waiting`.

`pod_interface_network_rx_dropped`
 	
Esta métrica solo está disponible con Información de contenedores con observabilidad mejorada para Amazon EKS
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName`
		
		`Namespace` , `ClusterName` , `Service`
		
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`
			La cantidad de paquetes que esta interfaz de red recibió y luego descartó para el pod.

Nombre de métrica	Dimensiones con cualquier versión de Información de contenedores	Dimensiones adicionales con Información de contenedores con observabilidad mejorada para Amazon EKS	Descripción

`pod_interface_network_tx_dropped`
 	
Esta métrica solo está disponible con Información de contenedores con observabilidad mejorada para Amazon EKS
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName`
		
		`Namespace` , `ClusterName` , `Service`
		
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`
			La cantidad de paquetes que debían transmitirse pero que se descartaron para el pod.

Nombre de métrica	Dimensiones con cualquier versión de Información de contenedores	Dimensiones adicionales con Información de contenedores con observabilidad mejorada para Amazon EKS	Descripción

`container_cpu_utilization`
 	
Esta métrica solo está disponible con Información de contenedores con observabilidad mejorada para Amazon EKS
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName` , `ContainerName`
		
		`PodName`, `Namespace` , `ClusterName` , `ContainerName` , `FullPodName`
			El porcentaje de unidades de CPU que utiliza el contenedor.
			
			Fórmula: `container_cpu_usage_total / node_cpu_limit`
			
			Note
			`container_cpu_utilization` no se informa directamente como una métrica, sino que es un campo en el registro de eventos del rendimiento. Para obtener más información, consulte Campos relevantes en eventos

Nombre de métrica	Dimensiones con cualquier versión de Información de contenedores	Dimensiones adicionales con Información de contenedores con observabilidad mejorada para Amazon EKS	Descripción
			de registro de rendimiento para Amazon EKS y Kubernetes.

Nombre de métrica	Dimensiones con cualquier versión de Información de contenedores	Dimensiones adicionales con Información de contenedores con observabilidad mejorada para Amazon EKS	Descripción

`container_cpu_utilization_over_container_limit`
 	
Esta métrica solo está disponible con Información de contenedores con observabilidad mejorada para Amazon EKS
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName` , `ContainerName`
		
		`PodName`, `Namespace` , `ClusterName` , `ContainerName` , `FullPodName`
			El porcentaje de unidades de CPU que utiliza el contenedor en relación con el límite de pods. Si algún contenedor no tiene definido un límite de la CPU, esta métrica no aparecerá.
			
			Fórmula: `container_cpu_usage_total / container_cpu_limit`
			
			Note
			`container_cpu_utilization_over_container_limit` no se informa directamente como una métrica, sino que es un

Nombre de métrica	Dimensiones con cualquier versión de Información de contenedores	Dimensiones adicionales con Información de contenedores con observabilidad mejorada para Amazon EKS	Descripción
			campo en el registro de eventos del rendimiento. Para obtener más información, consulte Campos relevantes en eventos de registro de rendimiento para Amazon EKS y Kubernetes.

Nombre de métrica	Dimensiones con cualquier versión de Información de contenedores	Dimensiones adicionales con Información de contenedores con observabilidad mejorada para Amazon EKS	Descripción

`container_memory_utilization`
 	
Esta métrica solo está disponible con Información de contenedores con observabilidad mejorada para Amazon EKS
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName` , `ContainerName`
		
		`PodName`, `Namespace` , `ClusterName` , `ContainerName` , `FullPodName`
			El porcentaje de unidades de memoria que utiliza el contenedor.
			
			Fórmula: `container_memory_working_set / node_memory_limit`
			
			Note
			`container_memory_utilization` no se informa directamente como una métrica, sino que es un campo en el registro de eventos del rendimiento. Para obtener más información, consulte

Nombre de métrica	Dimensiones con cualquier versión de Información de contenedores	Dimensiones adicionales con Información de contenedores con observabilidad mejorada para Amazon EKS	Descripción
			Campos relevantes en eventos de registro de rendimiento para Amazon EKS y Kubernetes.

Nombre de métrica	Dimensiones con cualquier versión de Información de contenedores	Dimensiones adicionales con Información de contenedores con observabilidad mejorada para Amazon EKS	Descripción

`container_memory_utilization_over_container_limit`
 	
Esta métrica solo está disponible con Información de contenedores con observabilidad mejorada para Amazon EKS
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName` , `ContainerName`
		
		`PodName`, `Namespace` , `ClusterName` , `ContainerName` , `FullPodName`
			El porcentaje de unidades de memoria que utiliza el contenedor en relación con el límite del contenedor. Si algún contenedor no tiene definido un límite de memoria, esta métrica no aparecerá.
			
			Fórmula: `container_memory_working_set / container_memory_limit`
			
			Note
			`container_memory_utilization_over_container_limit` no se informa directamente como una

Nombre de métrica	Dimensiones con cualquier versión de Información de contenedores	Dimensiones adicionales con Información de contenedores con observabilidad mejorada para Amazon EKS	Descripción
			métrica, sino que es un campo en el registro de eventos del rendimiento. Para obtener más información, consulte Campos relevantes en eventos de registro de rendimiento para Amazon EKS y Kubernetes.

Nombre de métrica	Dimensiones con cualquier versión de Información de contenedores	Dimensiones adicionales con Información de contenedores con observabilidad mejorada para Amazon EKS	Descripción

`container_memory_failures_total`
 	
Esta métrica solo está disponible con Información de contenedores con observabilidad mejorada para Amazon EKS. No está disponible en Windows.
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName` , `ContainerName`
		
		`PodName`, `Namespace` , `ClusterName` , `ContainerName` , `FullPodName`
			La cantidad de errores de asignación de memoria que experimentó el contenedor.

`pod_number_of_container_restarts`	PodName, `Namespace` , `ClusterName`		El número total de reinicios del contenedor en un pod.
`service_number_of_running_pods`	Servicio, `Namespace` , `ClusterName` `ClusterName`		El número de pods que ejecutan el servicio o servicios en el clúster.

Nombre de métrica	Dimensiones con cualquier versión de Información de contenedores	Dimensiones adicionales con Información de contenedores con observabilidad mejorada para Amazon EKS	Descripción

`replicas_desired`
 	
Esta métrica solo está disponible con Información de contenedores con observabilidad mejorada para Amazon EKS
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName`
			La cantidad de pods deseada para una carga de trabajo, tal como se define en la especificación de la carga de trabajo.

`replicas_ready`
 	
Esta métrica solo está disponible con Información de contenedores con observabilidad mejorada para Amazon EKS
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName`
			La cantidad de pods de una carga de trabajo que alcanzó el estado listo.

Nombre de métrica	Dimensiones con cualquier versión de Información de contenedores	Dimensiones adicionales con Información de contenedores con observabilidad mejorada para Amazon EKS	Descripción

`status_replicas_available`
 	
Esta métrica solo está disponible con Información de contenedores con observabilidad mejorada para Amazon EKS
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName`
			La cantidad de pods disponibles para una carga de trabajo. Un pod está disponible cuando está listo para cumplir con los `minReadySeconds` definidos en la especificación de la carga de trabajo.

`status_replicas_unavailable`
 	
Esta métrica solo está disponible con Información de contenedores con observabilidad mejorada para Amazon EKS
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName`
			La cantidad de pods de una carga de trabajo que no están disponibles. Un pod está disponible cuando está listo para cumplir con los `minReadySeconds` definidos en la especificación de la carga de trabajo. Los pods no están disponibles si no cumplen este criterio.

Nombre de métrica	Dimensiones con cualquier versión de Información de contenedores	Dimensiones adicionales con Información de contenedores con observabilidad mejorada para Amazon EKS	Descripción

`apiserver_storage_objects`
 	
Esta métrica solo está disponible con Información de contenedores con observabilidad mejorada para Amazon EKS
		`ClusterName`
		
		`ClusterName`, `resource`
			La cantidad de objetos almacenados en etcd en el momento de la última comprobación.

`apiserver_request_total`
 	
Esta métrica solo está disponible con Información de contenedores con observabilidad mejorada para Amazon EKS
		`ClusterName`
		
		`ClusterName`, `code`, `verb`
			La cantidad total de solicitudes de la API al servidor de la API de Kubernetes.

Nombre de métrica	Dimensiones con cualquier versión de Información de contenedores	Dimensiones adicionales con Información de contenedores con observabilidad mejorada para Amazon EKS	Descripción

`apiserver_request_duration_seconds`
 	
Esta métrica solo está disponible con Información de contenedores con observabilidad mejorada para Amazon EKS
		`ClusterName`
		
		`ClusterName , verb`
			La latencia de respuesta a las solicitudes de la API al servidor de la API de Kubernetes.

`apiserver_admission_controller_admission_duration_seconds`
 	
Esta métrica solo está disponible con Información de contenedores con observabilidad mejorada para Amazon EKS
		`ClusterName`
		
		`ClusterName , operation`
			La latencia del controlador de admisión en segundos. Un controlador de admisión es un código que intercept a las solicitudes al servidor de la API de Kubernetes.

Nombre de métrica	Dimensiones con cualquier versión de Información de contenedores	Dimensiones adicionales con Información de contenedores con observabilidad mejorada para Amazon EKS	Descripción

`rest_client_request_duration_seconds`
 	
Esta métrica solo está disponible con Información de contenedores con observabilidad mejorada para Amazon EKS
		`ClusterName`
		
		`ClusterName` , `operation`
			La latencia de respuesta que experimentan los clientes que llaman al servidor de la API de Kubernetes. Esta métrica es experimental y puede cambiar en futuras versiones de Kubernetes.

`rest_client_requests_total`
 	
Esta métrica solo está disponible con Información de contenedores con observabilidad mejorada para Amazon EKS
		`ClusterName`
		
		`ClusterName` , `code`, `method`
			La cantidad total de solicitudes de la API al servidor de la API de Kubernetes que hacen los clientes. Esta métrica es experimental y puede cambiar en futuras versiones de Kubernetes.

Nombre de métrica	Dimensiones con cualquier versión de Información de contenedores	Dimensiones adicionales con Información de contenedores con observabilidad mejorada para Amazon EKS	Descripción

`etcd_request_duration_seconds`
 	
Esta métrica solo está disponible con Información de contenedores con observabilidad mejorada para Amazon EKS
		`ClusterName`
		
		`ClusterName , operation`
			La latencia de respuesta de las llamadas de la API a Etcd. Esta métrica es experimental y puede cambiar en futuras versiones de Kubernetes.

`apiserver_storage_size_bytes`
 	
Esta métrica solo está disponible con Información de contenedores con observabilidad mejorada para Amazon EKS
		`ClusterName`
		
		`ClusterName , endpoint`
			El tamaño del archivo de base de datos de almacenamiento asignado físicamente en bytes. Esta métrica es experimental y puede cambiar en futuras versiones de Kubernetes.

Nombre de métrica	Dimensiones con cualquier versión de Información de contenedores	Dimensiones adicionales con Información de contenedores con observabilidad mejorada para Amazon EKS	Descripción

`apiserver_longrunning_requests`
 	
Esta métrica solo está disponible con Información de contenedores con observabilidad mejorada para Amazon EKS
		`ClusterName`
		
		`ClusterName , resource`
			La cantidad de solicitudes activas de larga duración al servidor de la API de Kubernetes.

`apiserver_current_inflight_requests`
 	
Esta métrica solo está disponible con Información de contenedores con observabilidad mejorada para Amazon EKS
		`ClusterName`
		
		`ClusterName , request_kind`
			La cantidad de solicitudes que procesa el servidor de la API de Kubernetes.

Nombre de métrica	Dimensiones con cualquier versión de Información de contenedores	Dimensiones adicionales con Información de contenedores con observabilidad mejorada para Amazon EKS	Descripción

`apiserver_admission_webhook_admission_duration_seconds`
 	
Esta métrica solo está disponible con Información de contenedores con observabilidad mejorada para Amazon EKS
		`ClusterName`
		
		`ClusterName , name`
			La latencia del webhook de admisión en segundos. Los webhooks de admisión son devoluciones de llamadas HTTP que reciben las solicitudes de admisión y hacen algo con ellas.

`apiserver_admission_step_admission_duration_seconds`
 	
Esta métrica solo está disponible con Información de contenedores con observabilidad mejorada para Amazon EKS
		`ClusterName`
		
		`ClusterName , operation`
			La latencia del subpaso de admisión en segundos.

Nombre de métrica	Dimensiones con cualquier versión de Información de contenedores	Dimensiones adicionales con Información de contenedores con observabilidad mejorada para Amazon EKS	Descripción

`apiserver_requested_deprecated_apis`
 	
Esta métrica solo está disponible con Información de contenedores con observabilidad mejorada para Amazon EKS
		`ClusterName`
		
		`ClusterName`, `group`
			La cantidad de solicitudes a API obsoletas en el servidor de la API de Kubernetes.

`apiserver_request_total_5XX`
 	
Esta métrica solo está disponible con Información de contenedores con observabilidad mejorada para Amazon EKS
		`ClusterName`
		
		`ClusterName`, `code`, `verb`
			La cantidad de solicitudes al servidor de la API de Kubernetes a las que se respondió con un código de respuesta HTTP 5XX.

Nombre de métrica	Dimensiones con cualquier versión de Información de contenedores	Dimensiones adicionales con Información de contenedores con observabilidad mejorada para Amazon EKS	Descripción

`apiserver_storage_list_duration_seconds`
 	
Esta métrica solo está disponible con Información de contenedores con observabilidad mejorada para Amazon EKS
		`ClusterName`
		
		`ClusterName , resource`
			La latencia de respuesta de los objetos listados de Etcd. Esta métrica es experimental y puede cambiar en futuras versiones de Kubernetes.

`apiserver_current_inqueue_requests`
 	
Esta métrica solo está disponible con Información de contenedores con observabilidad mejorada para Amazon EKS
		`ClusterName`
		
		`ClusterName , request_kind`
			La cantidad de solicitudes en cola que colocó el servidor de la API de Kubernetes. Esta métrica es experimental y puede cambiar en futuras versiones de Kubernetes.

Nombre de métrica	Dimensiones con cualquier versión de Información de contenedores	Dimensiones adicionales con Información de contenedores con observabilidad mejorada para Amazon EKS	Descripción

`apiserver_flowcontrol_rejected_requests_total`
 	
Esta métrica solo está disponible con Información de contenedores con observabilidad mejorada para Amazon EKS
		`ClusterName`
		
		`ClusterName` , `reason`
			La cantidad de solicitudes rechazadas por el Subsistema de prioridad y equidad de la API. Esta métrica es experimental y puede cambiar en futuras versiones de Kubernetes.

Métricas de GPU de NVIDIA

A partir de la versión `1.300034.0` del agente de CloudWatch, Información de contenedores con observabilidad mejorada para Amazon EKS recopila las métricas de GPU de NVIDIA de las cargas de trabajo de EKS de forma predeterminada. El agente de CloudWatch debe instalarse con la versión `v1.3.0-eksbuild.1` o posterior del complemento de EKS de observabilidad de Amazon CloudWatch. Para obtener más información, consulte Instalar el agente de CloudWatch mediante el complemento de EKS de observabilidad de Amazon CloudWatch. Estas métricas de GPU de NVIDIA que se recopilan se muestran en la tabla de esta sección.

Para que Información de contenedores recopile métricas de GPU de NVIDIA, debe cumplir los siguientes requisitos previos:

- Debe utilizar Información de contenedores con observabilidad mejorada para Amazon EKS, con la versión `v1.3.0-eksbuild.1` o posterior del complemento de EKS de observabilidad de Amazon CloudWatch.
- El complemento de dispositivo de NVIDIA para Kubernetes debe estar instalado en el clúster.
- El kit de herramientas de contenedor de NVIDIA debe estar instalado en los nodos del clúster. Por ejemplo, las AMI aceleradas optimizadas para Amazon EKS se crearon con los componentes necesarios.

Para dejar de recopilar métricas de GPU de NVIDIA, establezca la opción `accelerated_compute_metrics` del inicio del archivo de configuración del agente de CloudWatch como `false`. Para obtener más información y un ejemplo de configuración de desactivación, consulte Configuraciones adicionales (Opcional).

Nombre de métrica	Dimensiones	Descripción

`container_gpu_memory_total`	`ClusterName`
	
	`ClusterName` , `Namespace` , `PodName`, `ContainerName`
	
	`ClusterName` , `Namespace` , `PodName`, `FullPodName` , `ContainerName`
	
	`ClusterName` , `Namespace` , `PodName`, `FullPodName` , `ContainerName` , `GpuDevice`
		El tamaño total del búfer del marco, en bytes, en las GPU asignadas al contenedor.

`container_gpu_memory_used`	`ClusterName`
	
	`ClusterName` , `Namespace` , `PodName`, `ContainerName`
	
	`ClusterName` , `Namespace` , `PodName`, `FullPodName` , `ContainerName`
		Los bytes del búfer del marco que se usan en las GPU asignadas al contenedor.

Nombre de métrica	Dimensiones	Descripción
	`ClusterName` , `Namespace` , `PodName`, `FullPodName` , `ContainerName` , `GpuDevice`	

`container_gpu_memory_utilization`	`ClusterName`
	
	`ClusterName` , `Namespace` , `PodName`, `ContainerName`
	
	`ClusterName` , `Namespace` , `PodName`, `FullPodName` , `ContainerName`
	
	`ClusterName` , `Namespace` , `PodName`, `FullPodName` , `ContainerName` , `GpuDevice`
		El porcentaje del búfer del marco que se usa en las GPU asignadas al contenedor.

`container_gpu_power_draw`	`ClusterName`
	
	`ClusterName` , `Namespace` , `PodName`, `ContainerName`
	
	`ClusterName` , `Namespace` , `PodName`, `FullPodName` , `ContainerName`
	
	`ClusterName` , `Namespace` , `PodName`, `FullPodName` , `ContainerName` , `GpuDevice`
		El consumo de energía en vatios de las GPU asignadas al contenedor.

Nombre de métrica	Dimensiones	Descripción
container_gpu_temperature	ClusterName ClusterName , Namespace , PodName, ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName , GpuDevice	La temperatura en grados centígrados de las GPU asignadas al contenedor.
container_gpu_utilization	ClusterName ClusterName , Namespace , PodName, ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName , GpuDevice	El porcentaje de uso de las GPU asignadas al contenedor.
node_gpu_memory_total	ClusterName ClusterName , InstanceId , NodeName ClusterName , InstanceId , InstanceType , NodeName, GpuDevice	El tamaño total del búfer del marco, en bytes, en las GPU asignadas al nodo.

Nombre de métrica	Dimensiones	Descripción
node_gpu_memory_used	ClusterName ClusterName , InstanceId , NodeName ClusterName , InstanceId , InstanceType , NodeName, GpuDevice	Los bytes del búfer del marco que se usan en las GPU asignadas al nodo.
node_gpu_memory_utilization	ClusterName ClusterName , InstanceId , NodeName ClusterName , InstanceId , InstanceType , NodeName, GpuDevice	El porcentaje del búfer del marco que se usa en las GPU asignadas al nodo.
node_gpu_power_draw	ClusterName ClusterName , InstanceId , NodeName ClusterName , InstanceId , InstanceType , NodeName, GpuDevice	El consumo de energía en vatios de las GPU asignadas al nodo.
node_gpu_temperature	ClusterName ClusterName , InstanceId , NodeName ClusterName , InstanceId , InstanceType , NodeName, GpuDevice	La temperatura en grados centígrados de las GPU asignadas al nodo.

Nombre de métrica	Dimensiones	Descripción
node_gpu_utilization	ClusterName ClusterName , InstanceId , NodeName ClusterName , InstanceId , InstanceType , NodeName, GpuDevice	El porcentaje de uso de las GPU asignadas al nodo.
pod_gpu_memory_total	ClusterName ClusterName , Namespace ClusterName , Namespace , Service ClusterName , Namespace , PodName ClusterName , Namespace , PodName, FullPodName ClusterName , Namespace , PodName, FullPodName . GpuDevice	El tamaño total del búfer del marco, en bytes, en las GPU asignadas al pod.

Nombre de métrica	Dimensiones	Descripción
pod_gpu_memory_used	ClusterName ClusterName , Namespace ClusterName , Namespace , Service ClusterName , Namespace , PodName ClusterName , Namespace , PodName, FullPodName ClusterName , Namespace , PodName, FullPodName . GpuDevice	Los bytes del búfer del marco que se usan en las GPU asignadas al pod.
pod_gpu_memory_utilization	ClusterName ClusterName , Namespace ClusterName , Namespace , Service ClusterName , Namespace , PodName ClusterName , Namespace , PodName, FullPodName ClusterName , Namespace , PodName, FullPodName . GpuDevice	El porcentaje del búfer del marco que se usa en las GPU asignadas al pod.

Nombre de métrica	Dimensiones	Descripción
pod_gpu_power_draw	ClusterName ClusterName , Namespace ClusterName , Namespace , Service ClusterName , Namespace , PodName ClusterName , Namespace , PodName, FullPodName ClusterName , Namespace , PodName, FullPodName . GpuDevice	El consumo de energía en vatios de las GPU asignadas al pod.
pod_gpu_temperature	ClusterName ClusterName , Namespace ClusterName , Namespace , Service ClusterName , Namespace , PodName ClusterName , Namespace , PodName, FullPodName ClusterName , Namespace , PodName, FullPodName . GpuDevice	La temperatura en grados centígrados de las GPU asignadas al pod.

Nombre de métrica	Dimensiones	Descripción
pod_gpu_utilization	ClusterName ClusterName , Namespace , PodName, ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName , GpuDevice	El porcentaje de uso de las GPU asignadas al pod.

Métricas de AWS Neuron para AWS Trainium y AWS Inferentia

A partir de la versión `1.300036.0` del agente de CloudWatch, Información de contenedores con observabilidad mejorada para Amazon EKS recopila métricas de computación aceleradas de los aceleradores de AWS Trainium y AWS Inferentia de forma predeterminada. El agente de CloudWatch debe instalarse con la versión `v1.5.0-eksbuild.1` o posterior del complemento de EKS de observabilidad de Amazon CloudWatch. Para obtener más información acerca del complemento, consulte Instalar el agente de CloudWatch mediante el complemento de EKS de observabilidad de Amazon CloudWatch. Para obtener más información acerca de AWS Trainium, consulte AWS Trainium. Para obtener más información acerca de AWS Inferentia, consulte AWS Inferentia.

Para que Información de contenedores recopile métricas de AWS Neuron, debe cumplir los siguientes requisitos previos:

- Debe utilizar Información de contenedores con observabilidad mejorada para Amazon EKS, con la versión `v1.5.0-eksbuild.1` o posterior del complemento de EKS de observabilidad de Amazon CloudWatch.
- El controlador Neuron debe estar instalado en los nodos del clúster.
- El complemento del dispositivo Neuron debe estar instalado en el clúster. Por ejemplo, las AMI aceleradas optimizadas para Amazon EKS se crearon con los componentes necesarios.

Las métricas que se recopilan se muestran en la tabla de esta sección. Las métricas se recopilan para AWS Trainium, AWS Inferentia y AWS Inferentia2.

El agente CloudWatch recopila estas métricas de Neuron Monitor y lleva a cabo la correlación de recursos de Kubernetes necesaria para entregar las métricas a nivel de pod y contenedor.

Nombre de métrica	Dimensiones	Descripción

`container_neuroncore_utilization`	`ClusterName`
	
	`ClusterName`, `Namespace`, `PodName`, `ContainerName`
	
	`ClusterName`, `Namespace`, `PodName`, `FullPodName`, `ContainerName`
	
	`ClusterName`, `Namespace`, `PodName`, `FullPodName`, `ContainerName`, `NeuronDevice`, `NeuronCore`
		Utilización de NeuronCore, durante el período de captura del NeuronCore asignado al contenedor.
		
		Unidad: porcentaje

`container_neuroncore_memory_usage_constants`	`ClusterName`
	
	`ClusterName`, `Namespace`, `PodName`, `ContainerName`
	
	`ClusterName`, `Namespace`, `PodName`, `FullPodName`, `ContainerName`
	
	`ClusterName`, `Namespace`, `PodName`, `FullPodName`, `ContainerName`, `NeuronDevice`, `NeuronCore`
		La cantidad de memoria del dispositivo que NeuronCore utiliza para las constantes durante el entrenamiento y que se asigna al contenedor (o a los pesos durante la inferencia).
		
		Unidades: bytes

`container_neuroncore_memory`	`ClusterName`	La cantidad de memoria del dispositivo que NeuronCore utiliza para el

Nombre de métrica	Dimensiones	Descripción

`_usage_model_code`	`ClusterName` , `Namespace` , `PodName`, `ContainerName`
	
	`ClusterName` , `Namespace` , `PodName`, `FullPodName` , `ContainerName`
	
	`ClusterName` , `Namespace` , `PodName`, `FullPodName` , `ContainerName` , `NeuronDevice` , `NeuronCore`
		código ejecutable de los modelos y que se asigna al contenedor.
		
		Unidades: bytes

`container_neuroncore_memory_usage_model_shared_scratchpad`	`ClusterName`
	
	`ClusterName` , `Namespace` , `PodName`, `ContainerName`
	
	`ClusterName` , `Namespace` , `PodName`, `FullPodName` , `ContainerName`
	
	`ClusterName` , `Namespace` , `PodName`, `FullPodName` , `ContainerName` , `NeuronDevice` , `NeuronCore`
		La cantidad de memoria del dispositivo que NeuronCore utiliza para el scratchpad compartido de los modelos y que se asigna al contenedor. Esta región de memoria está reservada para los modelos.
		
		Unidades: bytes

Nombre de métrica	Dimensiones	Descripción
container_neuroncore_memory_usage_runtime_memory	ClusterName ClusterName , Namespace , PodName, ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName , NeuronDevice , NeuronCore	La cantidad de memoria del dispositivo que NeuronCore utiliza para el tiempo de ejecución de Neuron y que se asigna al contenedor. Unidades: bytes
container_neuroncore_memory_usage_tensors	ClusterName ClusterName , Namespace , PodName, ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName , NeuronDevice , NeuronCore	La cantidad de memoria del dispositivo que NeuronCore utiliza para los tensores y que se asigna al contenedor. Unidades: bytes

Nombre de métrica	Dimensiones	Descripción
container_neuroncore_memory_usage_total	ClusterName ClusterName , Namespace , PodName, ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName , NeuronDevice , NeuronCore	La cantidad total de memoria que NeuronCore utiliza y que se asigna al contenedor. Unidades: bytes
container_neurondevice_hw_ecc_events_total	ClusterName ClusterName , Namespace , PodName, ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName , NeuronDevice	El número de eventos de ECC corregidos y no corregidos para la SRAM integrada en el chip y la memoria del dispositivo Neuron del nodo. Unidad: recuento

Nombre de métrica	Dimensiones	Descripción
pod_neuroncore_utilization	ClusterName ClusterName , Namespace ClusterName , Namespace , Service ClusterName , Namespace , PodName, FullPodName ClusterName , Namespace , PodName, FullPodName , NeuronDevice , NeuronCore	La utilización de NeuronCore durante el período de captura de NeuronCore asignada al pod. Unidad: porcentaje
pod_neuroncore_memory_usage_constants	ClusterName ClusterName , Namespace ClusterName , Namespace , Service ClusterName , Namespace , PodName, FullPodName ClusterName , Namespace , PodName, FullPodName , NeuronDevice , NeuronCore	La cantidad de memoria del dispositivo que NeuronCore utiliza para las constantes durante el entrenamiento y que se asigna al pod (o a los pesos durante la inferencia). Unidades: bytes

Nombre de métrica	Dimensiones	Descripción
pod_neuroncore_memory_usage_model_code	ClusterName ClusterName , Namespace ClusterName , Namespace , Service ClusterName , Namespace , PodName, FullPodName ClusterName , Namespace , PodName, FullPodName , NeuronDevice , NeuronCore	La cantidad de memoria del dispositivo que NeuronCore utiliza para el código ejecutable de los modelos y que se asigna al pod. Unidades: bytes
pod_neuroncore_memory_usage_model_shared_scratchpad	ClusterName ClusterName , Namespace ClusterName , Namespace , Service ClusterName , Namespace , PodName, FullPodName ClusterName , Namespace , PodName, FullPodName , NeuronDevice , NeuronCore	La cantidad de memoria del dispositivo que NeuronCore utiliza para el scratchpad compartido de los modelos y que se asigna al pod. Esta región de memoria está reservada para los modelos. Unidades: bytes

Nombre de métrica	Dimensiones	Descripción
pod_neuroncore_memory_usage_runtime_memory	ClusterName ClusterName , Namespace ClusterName , Namespace , Service ClusterName , Namespace , PodName, FullPodName ClusterName , Namespace , PodName, FullPodName , NeuronDevice , NeuronCore	La cantidad de memoria del dispositivo que NeuronCore utiliza para el tiempo de ejecución de Neuron y que se asigna al pod. Unidades: bytes
pod_neuroncore_memory_usage_tensors	ClusterName ClusterName , Namespace ClusterName , Namespace , Service ClusterName , Namespace , PodName, FullPodName ClusterName , Namespace , PodName, FullPodName , NeuronDevice , NeuronCore	La cantidad de memoria del dispositivo que NeuronCore utiliza para los tensores y que se asigna al pod. Unidades: bytes

Nombre de métrica	Dimensiones	Descripción
pod_neuroncore_memory_usage_total	ClusterName ClusterName , Namespace ClusterName , Namespace , Service ClusterName , Namespace , PodName, FullPodName ClusterName , Namespace , PodName, FullPodName , NeuronDevice , NeuronCore	La cantidad total de memoria que NeuronCore utiliza y que se asigna al pod. Unidades: bytes
pod_neurondevice_hw_ecc_events_total	ClusterName ClusterName , Namespace ClusterName , Namespace , Service ClusterName , Namespace , PodName ClusterName , Namespace , PodName, FullPodName ClusterName , Namespace , PodName, FullPodName , NeuronDevice	La cantidad de eventos de ECC corregidos y no corregidos para la SRAM en el chip y la memoria del dispositivo Neuron asignada a un pod. Unidades: bytes

Nombre de métrica	Dimensiones	Descripción
node_neuroncore_utilization	ClusterName ClusterName , InstanceId , NodeName ClusterName , InstanceType , InstanceId , NodeName, NeuronDevice , NeuronCore	La utilización de NeuronCore durante el período de captura de NeuronCore asignada al nodo. Unidad: porcentaje
node_neuroncore_memory_usage_constants	ClusterName ClusterName , InstanceId , NodeName ClusterName , InstanceType , InstanceId , NodeName, NeuronDevice , NeuronCore	La cantidad de memoria del dispositivo que NeuronCore utiliza para las constantes durante el entrenamiento y que se asigna al nodo (o a los pesos durante la inferencia). Unidades: bytes
node_neuroncore_memory_usage_model_code	ClusterName ClusterName , InstanceId , NodeName ClusterName , InstanceType , InstanceId , NodeName, NeuronDevice , NeuronCore	La cantidad de memoria del dispositivo que NeuronCore utiliza para el código ejecutable de los modelos y que se asigna al nodo. Unidades: bytes
node_neuroncore_memory_usage_model_shared_scratchpad	ClusterName ClusterName , InstanceId , NodeName ClusterName , InstanceType , InstanceId , NodeName, NeuronDevice , NeuronCore	La cantidad de memoria del dispositivo que NeuronCore utiliza para el scratchpad compartido de los modelos y que se asigna al nodo. Esta es una región de memoria reservada para los modelos. Unidades: bytes

Nombre de métrica	Dimensiones	Descripción
node_neuroncore_memory_usage_runtime_memory	ClusterName ClusterName , InstanceId , NodeName ClusterName , InstanceType , InstanceId , NodeName, NeuronDevice , NeuronCore	La cantidad de memoria del dispositivo que NeuronCore utiliza para el tiempo de ejecución de Neuron y que se asigna al nodo. Unidades: bytes
node_neuroncore_memory_usage_tensors	ClusterName ClusterName , InstanceId , NodeName ClusterName , InstanceType , InstanceId , NodeName, NeuronDevice , NeuronCore	La cantidad de memoria del dispositivo que NeuronCore utiliza para los tensores y que se asigna al nodo. Unidades: bytes
node_neuroncore_memory_usage_total	ClusterName ClusterName , InstanceId , NodeName ClusterName , InstanceType , InstanceId , NodeName, NeuronDevice , NeuronCore	La cantidad total de memoria que NeuronCore utiliza y que se asigna al nodo. Unidades: bytes
node_neuron_execution_errors_total	ClusterName ClusterName , InstanceId , NodeName	El número total de errores de ejecución del nodo. El agente de CloudWatch lo calcula agregando los errores de los siguientes tipos: generic, numerical , transient , model, runtime y hardware Unidad: recuento

Nombre de métrica	Dimensiones	Descripción
node_neuron_device_runtime_memory_used_bytes	ClusterName ClusterName , InstanceId , NodeName	El uso total de memoria del dispositivo Neuron en bytes en el nodo. Unidades: bytes
node_neuron_execution_latency	ClusterName ClusterName , InstanceId , NodeName	En segundos, la latencia para una ejecución en el nodo medida por el tiempo de ejecución de Neuron. Unidad: segundos
node_neuron_device_hw_ecc_events_total	ClusterName ClusterName , InstanceId , NodeName ClusterName , InstanceId , NodeName, NeuronDevice	El número de eventos de ECC corregidos y no corregidos para la SRAM integrada en el chip y la memoria del dispositivo Neuron del nodo. Unidad: recuento

Métricas de AWS Elastic Fabric Adapter (EFA)

A partir de la versión `1.300037.0` del agente de CloudWatch, Información de contenedores con observabilidad mejorada para Amazon EKS recopila métricas de AWS Elastic Fabric Adapter (EFA) de clústeres de Amazon EKS en instancias de Linux. El agente de CloudWatch debe instalarse con la versión `v1.5.2-eksbuild.1` o posterior del complemento de EKS de observabilidad de Amazon CloudWatch. Para obtener más información acerca del complemento, consulte Instalar el agente de CloudWatch mediante el complemento de EKS de observabilidad de Amazon CloudWatch. Para obtener más información sobre AWS Elastic Fabric Adapter (EFA), consulte Elastic Fabric Adapter.

Para que Información de contenedores recopile métricas de AWS Elastic Fabric Adapter, debe cumplir los siguientes requisitos previos:

- Debe utilizar Información de contenedores con observabilidad mejorada para Amazon EKS, con la versión `v1.5.2-eksbuild.1` o posterior del complemento de EKS de observabilidad de Amazon CloudWatch.
- El complemento del dispositivo EFA debe estar instalado en el clúster. Para obtener más información, consulte aws-efa-k8s-device-plugin en GitHub.

Las métricas que se recopilan se enumeran en la siguiente tabla.

Nombre de métrica	Dimensiones	Descripción

`container_efa_rx_bytes`	`ClusterName`
	
	`ClusterName` , `Namespace` , `PodName`, `ContainerName`
	
	`ClusterName` , `Namespace` , `PodName`, `FullPodName` , `ContainerName`
	
	`ClusterName` , `Namespace` , `PodName`, `FullPodName` , `ContainerName` , `EfaDevice`
		El número de bytes por segundo que han recibido los dispositivos EFA asignados al contenedor.
		
		Unidad: bytes/segundo

`container_efa_tx_bytes`	`ClusterName`
	
	`ClusterName` , `Namespace` , `PodName`, `ContainerName`
	
	`ClusterName` , `Namespace` , `PodName`, `FullPodName` , `ContainerName`
	
	`ClusterName` , `Namespace` , `PodName`, `FullPodName` , `ContainerName` , `EfaDevice`
		El número de bytes por segundo que han transmitido los dispositivos EFA asignados al contenedor.
		
		Unidad: bytes/segundo

Nombre de métrica	Dimensiones	Descripción
container_efa_rx_dropped	ClusterName ClusterName , Namespace , PodName, ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName , EfaDevice	El número de paquetes que han recibido y luego descartado los dispositivos EFA asignados al contenedor. Unidad: recuento/segundo
container_efa_rdma_read_bytes	ClusterName ClusterName , Namespace , PodName, ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName , EfaDevice	El número de bytes por segundo que han recibido los dispositivos EFA asignados al contenedor mediante operaciones de lectura de acceso remoto directo a la memoria. Unidad: bytes/segundo

Nombre de métrica	Dimensiones	Descripción
container_efa_rdma_write_bytes	ClusterName ClusterName , Namespace , PodName, ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName , EfaDevice	El número de bytes por segundo que han transmitido los dispositivos EFA asignados al contenedor mediante operaciones de lectura de acceso remoto directo a la memoria. Unidad: bytes/segundo
container_efa_rdma_write_recv_bytes	ClusterName ClusterName , Namespace , PodName, ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName , EfaDevice	El número de bytes por segundo que han recibido los dispositivos EFA asignados al contenedor durante operaciones de escritura de acceso remoto directo a la memoria. Unidad: bytes/segundo

Nombre de métrica	Dimensiones	Descripción
pod_efa_rx_bytes	ClusterName ClusterName , Namespace ClusterName , Namespace , Service ClusterName , Namespace , PodName ClusterName , Namespace , PodName, FullPodName ClusterName , Namespace , PodName, FullPodName , EfaDevice	El número de bytes por segundo que han recibido los dispositivos EFA asignados al pod. Unidad: bytes/segundo
pod_efa_tx_bytes	ClusterName ClusterName , Namespace ClusterName , Namespace , Service ClusterName , Namespace , PodName ClusterName , Namespace , PodName, FullPodName ClusterName , Namespace , PodName, FullPodName , EfaDevice	El número de bytes por segundo que han transmitido los dispositivos EFA asignados al pod. Unidad: bytes/segundo

Nombre de métrica	Dimensiones	Descripción
pod_efa_rx_dropped	ClusterName ClusterName , Namespace ClusterName , Namespace , Service ClusterName , Namespace , PodName ClusterName , Namespace , PodName, FullPodName ClusterName , Namespace , PodName, FullPodName , EfaDevice	El número de paquetes que han recibido y luego descartado los dispositivos EFA asignados al pod. Unidad: recuento/segundo
pod_efa_rdma_read_bytes	ClusterName ClusterName , Namespace ClusterName , Namespace , Service ClusterName , Namespace , PodName ClusterName , Namespace , PodName, FullPodName ClusterName , Namespace , PodName, FullPodName , EfaDevice	El número de bytes por segundo que han recibido los dispositivos EFA asignados al pod mediante operaciones de lectura de acceso remoto directo a la memoria. Unidad: bytes/segundo

Nombre de métrica	Dimensiones	Descripción
pod_efa_rdma_write_bytes	ClusterName ClusterName , Namespace ClusterName , Namespace , Service ClusterName , Namespace , PodName ClusterName , Namespace , PodName, FullPodName ClusterName , Namespace , PodName, FullPodName , EfaDevice	El número de bytes por segundo que han transmitido los dispositivos EFA asignados al pod mediante operaciones de lectura de acceso remoto directo a la memoria. Unidad: bytes/segundo
pod_efa_rdma_write_recv_bytes	ClusterName ClusterName , Namespace ClusterName , Namespace , Service ClusterName , Namespace , PodName ClusterName , Namespace , PodName, FullPodName ClusterName , Namespace , PodName, FullPodName , EfaDevice	El número de bytes por segundo que han recibido los dispositivos EFA asignados al pod durante operaciones de escritura de acceso remoto directo a la memoria. Unidad: bytes/segundo

Nombre de métrica	Dimensiones	Descripción
node_efa_rx_bytes	ClusterName ClusterName , InstanceId , NodeName ClusterName , InstanceId , InstanceType , NodeName, EfaDevice	El número de bytes por segundo que han recibido los dispositivos EFA asignados al nodo. Unidad: bytes/segundo
node_efa_tx_bytes	ClusterName ClusterName , InstanceId , NodeName ClusterName , InstanceId , InstanceType , NodeName, EfaDevice	El número de bytes por segundo que han transmitido los dispositivos EFA asignados al nodo. Unidad: bytes/segundo
node_efa_rx_dropped	ClusterName ClusterName , InstanceId , NodeName ClusterName , InstanceId , InstanceType , NodeName, EfaDevice	El número de paquetes que han recibido y luego descartado los dispositivos EFA asignados al nodo. Unidad: recuento/segundo
node_efa_rdma_read_bytes	ClusterName ClusterName , InstanceId , NodeName ClusterName , InstanceId , InstanceType , NodeName, EfaDevice	El número de bytes por segundo que han recibido los dispositivos EFA asignados al nodo mediante operaciones de lectura de acceso remoto directo a la memoria. Unidad: bytes/segundo

Nombre de métrica	Dimensiones	Descripción
pod_efa_rdma_write_bytes	ClusterName ClusterName , InstanceId , NodeName ClusterName , InstanceId , InstanceType , NodeName, EfaDevice	El número de bytes por segundo que han transmitido los dispositivos EFA asignados al pod mediante operaciones de lectura de acceso remoto directo a la memoria. Unidad: bytes/segundo
node_efa_rdma_write_recv_bytes	ClusterName ClusterName , InstanceId , NodeName ClusterName , InstanceId , InstanceType , NodeName, EfaDevice	El número de bytes por segundo que han recibido los dispositivos EFA asignados al nodo durante operaciones de escritura de acceso remoto directo a la memoria. Unidad: bytes/segundo

Referencia de registros de rendimiento de Información de contenedores

Esta sección incluye información de referencia sobre cómo Información de contenedores utiliza los eventos de registro de rendimiento para recopilar métricas. Al implementar Información de contenedores, este crea automáticamente un grupo de registro para los eventos del registro de rendimiento. No es necesario que usted mismo cree este grupo de registro.

Temas

- Eventos de registro de rendimiento de Información de contenedores para Amazon ECS
- Eventos de registro de rendimiento de Información de contenedores para Amazon EKS y Kubernetes
- Campos relevantes en eventos de registro de rendimiento para Amazon EKS y Kubernetes

Eventos de registro de rendimiento de Información de contenedores para Amazon ECS

A continuación, se muestran ejemplos de los eventos de registro de rendimiento que Información de contenedores recopila de Amazon ECS.

Estos registros se encuentran en registros de CloudWatch, en un grupo de registro denominado `/aws/ecs/containerinsights/`*`CLUSTER_NAME`*`/performance`. Dentro de ese grupo de registro, cada instancia de contenedor tendrá un flujo de registro denominado `AgentTelemetry-`*`CONTAINER_INSTANCE_ID`*.

Puede consultar estos registros mediante consultas como, por ejemplo, `{ $.Type = "Container" }` para ver todos los eventos del registro del contenedor.

Tipo: Contenedor


```
{
 "Version":"0",
 "Type":"Container",
 "ContainerName":"sleep",
 "TaskId":"7ac4dfba69214411b4783a3b8189c9ba",
 "TaskDefinitionFamily":"sleep360",
 "TaskDefinitionRevision":"1",
 "ContainerInstanceId":"0d7650e6dec34c1a9200f72098071e8f",
 "EC2InstanceId":"i-0c470579dbcdbd2f3",
 "ClusterName":"MyCluster",
 "Image":"busybox",
 "ContainerKnownStatus":"RUNNING",
 "Timestamp":1623963900000,
 "CpuUtilized":0.0,
 "CpuReserved":10.0,
 "MemoryUtilized":0,
 "MemoryReserved":10,
 "StorageReadBytes":0,
 "StorageWriteBytes":0,
 "NetworkRxBytes":0,
 "NetworkRxDropped":0,
 "NetworkRxErrors":0,
 "NetworkRxPackets":14,
 "NetworkTxBytes":0,
 "NetworkTxDropped":0,
 "NetworkTxErrors":0,
```

```
 "NetworkTxPackets":0
}
```

Tipo: Tarea

```
{
    "Version": "0",
    "Type": "Task",
    "TaskId": "7ac4dfba69214411b4783a3b8189c9ba",
    "TaskDefinitionFamily": "sleep360",
    "TaskDefinitionRevision": "1",
    "ContainerInstanceId": "0d7650e6dec34c1a9200f72098071e8f",
    "EC2InstanceId": "i-0c470579dbcdbd2f3",
    "ClusterName": "MyCluster",
    "AccountID": "637146863587",
    "Region": "us-west-2",
    "AvailabilityZone": "us-west-2b",
    "KnownStatus": "RUNNING",
    "LaunchType": "EC2",
    "PullStartedAt": 1623963608201,
    "PullStoppedAt": 1623963610065,
    "CreatedAt": 1623963607094,
    "StartedAt": 1623963610382,
    "Timestamp": 1623963900000,
    "CpuUtilized": 0.0,
    "CpuReserved": 10.0,
    "MemoryUtilized": 0,
    "MemoryReserved": 10,
    "StorageReadBytes": 0,
    "StorageWriteBytes": 0,
    "NetworkRxBytes": 0,
    "NetworkRxDropped": 0,
    "NetworkRxErrors": 0,
    "NetworkRxPackets": 14,
    "NetworkTxBytes": 0,
    "NetworkTxDropped": 0,
    "NetworkTxErrors": 0,
    "NetworkTxPackets": 0,
    "EBSFilesystemUtilized": 10,
    "EBSFilesystemSize": 20,
    "CloudWatchMetrics": [
        {
            "Namespace": "ECS/ContainerInsights",
```

```
            "Metrics": [
                {
                    "Name": "CpuUtilized",
                    "Unit": "None"
                },
                {
                    "Name": "CpuReserved",
                    "Unit": "None"
                },
                {
                    "Name": "MemoryUtilized",
                    "Unit": "Megabytes"
                },
                {
                    "Name": "MemoryReserved",
                    "Unit": "Megabytes"
                },
                {
                    "Name": "StorageReadBytes",
                    "Unit": "Bytes/Second"
                },
                {
                    "Name": "StorageWriteBytes",
                    "Unit": "Bytes/Second"
                },
                {
                    "Name": "NetworkRxBytes",
                    "Unit": "Bytes/Second"
                },
                {
                    "Name": "NetworkTxBytes",
                    "Unit": "Bytes/Second"
                },
                {
                    "Name": "EBSFilesystemSize",
                    "Unit": "Gigabytes"
                },
                {
                    "Name": "EBSFilesystemUtilzed",
                    "Unit": "Gigabytes"
                }
            ],
            "Dimensions": [
                ["ClusterName"],
```

```
                [
                    "ClusterName",
                    "TaskDefinitionFamily"
                ]
            ]
        }
    ]
}
```

Tipo: Servicio

```
{
    "Version": "0",
    "Type": "Service",
    "ServiceName": "myCIService",
    "ClusterName": "myCICluster",
    "Timestamp": 1561586460000,
    "DesiredTaskCount": 2,
    "RunningTaskCount": 2,
    "PendingTaskCount": 0,
    "DeploymentCount": 1,
    "TaskSetCount": 0,
    "CloudWatchMetrics": [
        {
            "Namespace": "ECS/ContainerInsights",
            "Metrics": [
                {
                    "Name": "DesiredTaskCount",
                    "Unit": "Count"
                },
                {
                    "Name": "RunningTaskCount",
                    "Unit": "Count"
                },
                {
                    "Name": "PendingTaskCount",
                    "Unit": "Count"
                },
                {
                    "Name": "DeploymentCount",
                    "Unit": "Count"
                },
                {
```

```
                    "Name": "TaskSetCount",
                    "Unit": "Count"
                }
            ],
            "Dimensions": [
                [
                    "ServiceName",
                    "ClusterName"
                ]
            ]
        }
    ]
}
```

Tipo: volumen

```
{
    "Version": "0",
    "Type": "Volume",
    "TaskDefinitionFamily": "myCITaskDef",
    "TaskId": "7ac4dfba69214411b4783a3b8189c9ba",
    "ClusterName": "myCICluster",
    "ServiceName": "myCIService",
    "VolumeId": "vol-1233436545ff708cb",
    "InstanceId": "i-0c470579dbcdbd2f3",
    "LaunchType": "EC2",
    "VolumeName": "MyVolumeName",
    "EBSFilesystemUtilized": 10,
    "EBSFilesystemSize": 20,
    "CloudWatchMetrics": [
        {
            "Namespace": "ECS/ContainerInsights",
            "Metrics": [
                {
                    "Name": "EBSFilesystemSize",
                    "Unit": "Gigabytes"
                },
                {
                    "Name": "EBSFilesystemUtilzed",
                    "Unit": "Gigabytes"
                }
            ],
            "Dimensions": [
```

```
                ["ClusterName"],
                [
                    "VolumeName",
                    "TaskDefinitionFamily",
                    "ClusterName"
                ],
                [
                    "ServiceName",
                    "ClusterName"
                ]
            ]
        }
    ]
}
```

Tipo: Clúster

```
{
    "Version": "0",
    "Type": "Cluster",
    "ClusterName": "myCICluster",
    "Timestamp": 1561587300000,
    "TaskCount": 5,
    "ContainerInstanceCount": 5,
    "ServiceCount": 2,
    "CloudWatchMetrics": [
        {
            "Namespace": "ECS/ContainerInsights",
            "Metrics": [
                {
                    "Name": "TaskCount",
                    "Unit": "Count"
                },
                {
                    "Name": "ContainerInstanceCount",
                    "Unit": "Count"
                },
                {
                    "Name": "ServiceCount",
                    "Unit": "Count"
                }
            ],
            "Dimensions": [
```

```
                [
                    "ClusterName"
                ]
            ]
        }
    ]
}
```

Eventos de registro de rendimiento de Información de contenedores para Amazon EKS y Kubernetes

A continuación, se muestran ejemplos de los eventos de registro de rendimiento que Información de contenedores recopila de clústeres de Amazon EKS y de Kubernetes.

Tipo: Node

```
{
  "AutoScalingGroupName": "eksctl-myCICluster-nodegroup-standard-workers-
NodeGroup-1174PV2WHZAYU",
  "CloudWatchMetrics": [
    {
      "Metrics": [
        {
          "Unit": "Percent",
          "Name": "node_cpu_utilization"
        },
        {
          "Unit": "Percent",
          "Name": "node_memory_utilization"
        },
        {
          "Unit": "Bytes/Second",
          "Name": "node_network_total_bytes"
        },
        {
          "Unit": "Percent",
          "Name": "node_cpu_reserved_capacity"
        },
        {
          "Unit": "Percent",
          "Name": "node_memory_reserved_capacity"
        },
        {
```

```
          "Unit": "Count",
          "Name": "node_number_of_running_pods"
        },
        {
          "Unit": "Count",
          "Name": "node_number_of_running_containers"
        }
      ],
      "Dimensions": [
        [
          "NodeName",
          "InstanceId",
          "ClusterName"
        ]
      ],
      "Namespace": "ContainerInsights"
    },
    {
      "Metrics": [
        {
          "Unit": "Percent",
          "Name": "node_cpu_utilization"
        },
        {
          "Unit": "Percent",
          "Name": "node_memory_utilization"
        },
        {
          "Unit": "Bytes/Second",
          "Name": "node_network_total_bytes"
        },
        {
          "Unit": "Percent",
          "Name": "node_cpu_reserved_capacity"
        },
        {
          "Unit": "Percent",
          "Name": "node_memory_reserved_capacity"
        },
        {
          "Unit": "Count",
          "Name": "node_number_of_running_pods"
        },
        {
```

```
          "Unit": "Count",
          "Name": "node_number_of_running_containers"
        },
        {
          "Name": "node_cpu_usage_total"
        },
        {
          "Name": "node_cpu_limit"
        },
        {
          "Unit": "Bytes",
          "Name": "node_memory_working_set"
        },
        {
          "Unit": "Bytes",
          "Name": "node_memory_limit"
        }
      ],
      "Dimensions": [
        [
          "ClusterName"
        ]
      ],
      "Namespace": "ContainerInsights"
    }
  ],
  "ClusterName": "myCICluster",
  "InstanceId": "i-1234567890123456",
  "InstanceType": "t3.xlarge",
  "NodeName": "ip-192-0-2-0.us-west-2.compute.internal",
  "Sources": [
    "cadvisor",
    "/proc",
    "pod",
    "calculated"
  ],
  "Timestamp": "1567096682364",
  "Type": "Node",
  "Version": "0",
  "kubernetes": {
    "host": "ip-192-168-75-26.us-west-2.compute.internal"
  },
  "node_cpu_limit": 4000,
  "node_cpu_request": 1130,
```

```
  "node_cpu_reserved_capacity": 28.249999999999996,
  "node_cpu_usage_system": 33.794636630852764,
  "node_cpu_usage_total": 136.47852169244098,
  "node_cpu_usage_user": 71.67075111567326,
  "node_cpu_utilization": 3.4119630423110245,
  "node_memory_cache": 3103297536,
  "node_memory_failcnt": 0,
  "node_memory_hierarchical_pgfault": 0,
  "node_memory_hierarchical_pgmajfault": 0,
  "node_memory_limit": 16624865280,
  "node_memory_mapped_file": 406646784,
  "node_memory_max_usage": 4230746112,
  "node_memory_pgfault": 0,
  "node_memory_pgmajfault": 0,
  "node_memory_request": 1115684864,
  "node_memory_reserved_capacity": 6.7109407818311055,
  "node_memory_rss": 798146560,
  "node_memory_swap": 0,
  "node_memory_usage": 3901444096,
  "node_memory_utilization": 6.601302600149552,
  "node_memory_working_set": 1097457664,
  "node_network_rx_bytes": 35918.392817386324,
  "node_network_rx_dropped": 0,
  "node_network_rx_errors": 0,
  "node_network_rx_packets": 157.67565245448117,
  "node_network_total_bytes": 68264.20276554905,
  "node_network_tx_bytes": 32345.80994816272,
  "node_network_tx_dropped": 0,
  "node_network_tx_errors": 0,
  "node_network_tx_packets": 154.21455923431654,
  "node_number_of_running_containers": 16,
  "node_number_of_running_pods": 13
}
```

Tipo: NodeFS

```
{
  "AutoScalingGroupName": "eksctl-myCICluster-nodegroup-standard-workers-
NodeGroup-1174PV2WHZAYU",
  "CloudWatchMetrics": [
    {
      "Metrics": [
        {
```

```
          "Unit": "Percent",
          "Name": "node_filesystem_utilization"
        }
      ],
      "Dimensions": [
        [
          "NodeName",
          "InstanceId",
          "ClusterName"
        ],
        [
          "ClusterName"
        ]
      ],
      "Namespace": "ContainerInsights"
    }
  ],
  "ClusterName": "myCICluster",
  "EBSVolumeId": "aws://us-west-2b/vol-0a53108976d4a2fda",
  "InstanceId": "i-1234567890123456",
  "InstanceType": "t3.xlarge",
  "NodeName": "ip-192-0-2-0.us-west-2.compute.internal",
  "Sources": [
    "cadvisor",
    "calculated"
  ],
  "Timestamp": "1567097939726",
  "Type": "NodeFS",
  "Version": "0",
  "device": "/dev/nvme0n1p1",
  "fstype": "vfs",
  "kubernetes": {
    "host": "ip-192-168-75-26.us-west-2.compute.internal"
  },
  "node_filesystem_available": 17298395136,
  "node_filesystem_capacity": 21462233088,
  "node_filesystem_inodes": 10484720,
  "node_filesystem_inodes_free": 10367158,
  "node_filesystem_usage": 4163837952,
  "node_filesystem_utilization": 19.400767547940255
}
```


Tipo: NodeDiskIO

```
{
  "AutoScalingGroupName": "eksctl-myCICluster-nodegroup-standard-workers-
NodeGroup-1174PV2WHZAYU",
  "ClusterName": "myCICluster",
  "EBSVolumeId": "aws://us-west-2b/vol-0a53108976d4a2fda",
  "InstanceId": "i-1234567890123456",
  "InstanceType": "t3.xlarge",
  "NodeName": "ip-192-0-2-0.us-west-2.compute.internal",
  "Sources": [
    "cadvisor"
  ],
  "Timestamp": "1567096928131",
  "Type": "NodeDiskIO",
  "Version": "0",
  "device": "/dev/nvme0n1",
  "kubernetes": {
    "host": "ip-192-168-75-26.us-west-2.compute.internal"
  },
  "node_diskio_io_service_bytes_async": 9750.505814277016,
  "node_diskio_io_service_bytes_read": 0,
  "node_diskio_io_service_bytes_sync": 230.6174506688036,
  "node_diskio_io_service_bytes_total": 9981.123264945818,
  "node_diskio_io_service_bytes_write": 9981.123264945818,
  "node_diskio_io_serviced_async": 1.153087253344018,
  "node_diskio_io_serviced_read": 0,
  "node_diskio_io_serviced_sync": 0.03603397666700056,
  "node_diskio_io_serviced_total": 1.1891212300110185,
  "node_diskio_io_serviced_write": 1.1891212300110185
}
```


Tipo: NodeNet


```
{
  "AutoScalingGroupName": "eksctl-myCICluster-nodegroup-standard-workers-
NodeGroup-1174PV2WHZAYU",
  "ClusterName": "myCICluster",
  "InstanceId": "i-1234567890123456",
  "InstanceType": "t3.xlarge",
  "NodeName": "ip-192-0-2-0.us-west-2.compute.internal",
  "Sources": [
    "cadvisor",
    "calculated"
  ],
```

```
  "Timestamp": "1567096928131",
  "Type": "NodeNet",
  "Version": "0",
  "interface": "eni972f6bfa9a0",
  "kubernetes": {
    "host": "ip-192-168-75-26.us-west-2.compute.internal"
  },
  "node_interface_network_rx_bytes": 3163.008420864309,
  "node_interface_network_rx_dropped": 0,
  "node_interface_network_rx_errors": 0,
  "node_interface_network_rx_packets": 16.575629266820258,
  "node_interface_network_total_bytes": 3518.3935157426017,
  "node_interface_network_tx_bytes": 355.385094878293,
  "node_interface_network_tx_dropped": 0,
  "node_interface_network_tx_errors": 0,
  "node_interface_network_tx_packets": 3.9997714100370625
}
```

Tipo: Pod

```
{
  "AutoScalingGroupName": "eksctl-myCICluster-nodegroup-standard-workers-
NodeGroup-1174PV2WHZAYU",
  "CloudWatchMetrics": [
    {
      "Metrics": [
        {
          "Unit": "Percent",
          "Name": "pod_cpu_utilization"
        },
        {
          "Unit": "Percent",
          "Name": "pod_memory_utilization"
        },
        {
          "Unit": "Bytes/Second",
          "Name": "pod_network_rx_bytes"
        },
        {
          "Unit": "Bytes/Second",
          "Name": "pod_network_tx_bytes"
        },
        {
```

```
          "Unit": "Percent",
          "Name": "pod_cpu_utilization_over_pod_limit"
        },
        {
          "Unit": "Percent",
          "Name": "pod_memory_utilization_over_pod_limit"
        }
      ],
      "Dimensions": [
        [
          "PodName",
          "Namespace",
          "ClusterName"
        ],
        [
          "Service",
          "Namespace",
          "ClusterName"
        ],
        [
          "Namespace",
          "ClusterName"
        ],
        [
          "ClusterName"
        ]
      ],
      "Namespace": "ContainerInsights"
    },
    {
      "Metrics": [
        {
          "Unit": "Percent",
          "Name": "pod_cpu_reserved_capacity"
        },
        {
          "Unit": "Percent",
          "Name": "pod_memory_reserved_capacity"
        }
      ],
      "Dimensions": [
        [
          "PodName",
          "Namespace",
```

```
          "ClusterName"
        ],
        [
          "ClusterName"
        ]
      ],
      "Namespace": "ContainerInsights"
    },
    {
      "Metrics": [
        {
          "Unit": "Count",
          "Name": "pod_number_of_container_restarts"
        }
      ],
      "Dimensions": [
        [
          "PodName",
          "Namespace",
          "ClusterName"
        ]
      ],
      "Namespace": "ContainerInsights"
    }
  ],
  "ClusterName": "myCICluster",
  "InstanceId": "i-1234567890123456",
  "InstanceType": "t3.xlarge",
  "Namespace": "amazon-cloudwatch",
  "NodeName": "ip-192-0-2-0.us-west-2.compute.internal",
  "PodName": "cloudwatch-agent-statsd",
  "Service": "cloudwatch-agent-statsd",
  "Sources": [
    "cadvisor",
    "pod",
    "calculated"
  ],
  "Timestamp": "1567097351092",
  "Type": "Pod",
  "Version": "0",
  "kubernetes": {
    "host": "ip-192-168-75-26.us-west-2.compute.internal",
    "labels": {
      "app": "cloudwatch-agent-statsd",
```

```
      "pod-template-hash": "df44f855f"
    },
    "namespace_name": "amazon-cloudwatch",
    "pod_id": "2f4ff5ac-c813-11e9-a31d-06e9dde32928",
    "pod_name": "cloudwatch-agent-statsd-df44f855f-ts4q2",
    "pod_owners": [
      {
        "owner_kind": "Deployment",
        "owner_name": "cloudwatch-agent-statsd"
      }
    ],
    "service_name": "cloudwatch-agent-statsd"
  },
  "pod_cpu_limit": 200,
  "pod_cpu_request": 200,
  "pod_cpu_reserved_capacity": 5,
  "pod_cpu_usage_system": 1.4504841104992765,
  "pod_cpu_usage_total": 5.817016867430125,
  "pod_cpu_usage_user": 1.1281543081661038,
  "pod_cpu_utilization": 0.14542542168575312,
  "pod_cpu_utilization_over_pod_limit": 2.9085084337150624,
  "pod_memory_cache": 8192,
  "pod_memory_failcnt": 0,
  "pod_memory_hierarchical_pgfault": 0,
  "pod_memory_hierarchical_pgmajfault": 0,
  "pod_memory_limit": 104857600,
  "pod_memory_mapped_file": 0,
  "pod_memory_max_usage": 25268224,
  "pod_memory_pgfault": 0,
  "pod_memory_pgmajfault": 0,
  "pod_memory_request": 104857600,
  "pod_memory_reserved_capacity": 0.6307275170893897,
  "pod_memory_rss": 22777856,
  "pod_memory_swap": 0,
  "pod_memory_usage": 25141248,
  "pod_memory_utilization": 0.10988455961791709,
  "pod_memory_utilization_over_pod_limit": 17.421875,
  "pod_memory_working_set": 18268160,
  "pod_network_rx_bytes": 9880.697124714186,
  "pod_network_rx_dropped": 0,
  "pod_network_rx_errors": 0,
  "pod_network_rx_packets": 107.80005532263283,
  "pod_network_total_bytes": 10158.829201483635,
  "pod_network_tx_bytes": 278.13207676944796,
```

```
  "pod_network_tx_dropped": 0,
  "pod_network_tx_errors": 0,
  "pod_network_tx_packets": 1.146027574644318,
  "pod_number_of_container_restarts": 0,
  "pod_number_of_containers": 1,
  "pod_number_of_running_containers": 1,
  "pod_status": "Running"
}
```

Tipo: PodNet

```
{
  "AutoScalingGroupName": "eksctl-myCICluster-nodegroup-standard-workers-
NodeGroup-1174PV2WHZAYU",
  "ClusterName": "myCICluster",
  "InstanceId": "i-1234567890123456",
  "InstanceType": "t3.xlarge",
  "Namespace": "amazon-cloudwatch",
  "NodeName": "ip-192-0-2-0.us-west-2.compute.internal",
  "PodName": "cloudwatch-agent-statsd",
  "Service": "cloudwatch-agent-statsd",
  "Sources": [
    "cadvisor",
    "calculated"
  ],
  "Timestamp": "1567097351092",
  "Type": "PodNet",
  "Version": "0",
  "interface": "eth0",
  "kubernetes": {
    "host": "ip-192-168-75-26.us-west-2.compute.internal",
    "labels": {
      "app": "cloudwatch-agent-statsd",
      "pod-template-hash": "df44f855f"
    },
    "namespace_name": "amazon-cloudwatch",
    "pod_id": "2f4ff5ac-c813-11e9-a31d-06e9dde32928",
    "pod_name": "cloudwatch-agent-statsd-df44f855f-ts4q2",
    "pod_owners": [
      {
        "owner_kind": "Deployment",
        "owner_name": "cloudwatch-agent-statsd"
      }
```

```
    ],
    "service_name": "cloudwatch-agent-statsd"
  },
  "pod_interface_network_rx_bytes": 9880.697124714186,
  "pod_interface_network_rx_dropped": 0,
  "pod_interface_network_rx_errors": 0,
  "pod_interface_network_rx_packets": 107.80005532263283,
  "pod_interface_network_total_bytes": 10158.829201483635,
  "pod_interface_network_tx_bytes": 278.13207676944796,
  "pod_interface_network_tx_dropped": 0,
  "pod_interface_network_tx_errors": 0,
  "pod_interface_network_tx_packets": 1.146027574644318
}
```

Tipo: Contenedor

```
{
  "AutoScalingGroupName": "eksctl-myCICluster-nodegroup-standard-workers-NodeGroup-
sample",
  "ClusterName": "myCICluster",
  "InstanceId": "i-1234567890123456",
  "InstanceType": "t3.xlarge",
  "Namespace": "amazon-cloudwatch",
  "NodeName": "ip-192-0-2-0.us-west-2.compute.internal",
  "PodName": "cloudwatch-agent-statsd",
  "Service": "cloudwatch-agent-statsd",
  "Sources": [
    "cadvisor",
    "pod",
    "calculated"
  ],
  "Timestamp": "1567097399912",
  "Type": "Container",
  "Version": "0",
  "container_cpu_limit": 200,
  "container_cpu_request": 200,
  "container_cpu_usage_system": 1.87958283771964,
  "container_cpu_usage_total": 6.159993652997942,
  "container_cpu_usage_user": 1.6707403001952357,
  "container_cpu_utilization": 0.15399984132494854,
  "container_memory_cache": 8192,
  "container_memory_failcnt": 0,
  "container_memory_hierarchical_pgfault": 0,
```

```
  "container_memory_hierarchical_pgmajfault": 0,
  "container_memory_limit": 104857600,
  "container_memory_mapped_file": 0,
  "container_memory_max_usage": 24580096,
  "container_memory_pgfault": 0,
  "container_memory_pgmajfault": 0,
  "container_memory_request": 104857600,
  "container_memory_rss": 22736896,
  "container_memory_swap": 0,
  "container_memory_usage": 24453120,
  "container_memory_utilization": 0.10574541028701798,
  "container_memory_working_set": 17580032,
  "container_status": "Running",
  "kubernetes": {
    "container_name": "cloudwatch-agent",
    "docker": {
      "container_id":
 "8967b6b37da239dfad197c9fdea3e5dfd35a8a759ec86e2e4c3f7b401e232706"
    },
    "host": "ip-192-168-75-26.us-west-2.compute.internal",
    "labels": {
      "app": "cloudwatch-agent-statsd",
      "pod-template-hash": "df44f855f"
    },
    "namespace_name": "amazon-cloudwatch",
    "pod_id": "2f4ff5ac-c813-11e9-a31d-06e9dde32928",
    "pod_name": "cloudwatch-agent-statsd-df44f855f-ts4q2",
    "pod_owners": [
      {
        "owner_kind": "Deployment",
        "owner_name": "cloudwatch-agent-statsd"
      }
    ],
    "service_name": "cloudwatch-agent-statsd"
  },
  "number_of_container_restarts": 0
}
```

Tipo: ContainerFS

```
{
  "AutoScalingGroupName": "eksctl-myCICluster-nodegroup-standard-workers-
NodeGroup-1174PV2WHZAYU",
```

```
  "ClusterName": "myCICluster",
  "EBSVolumeId": "aws://us-west-2b/vol-0a53108976d4a2fda",
  "InstanceId": "i-1234567890123456",
  "InstanceType": "t3.xlarge",
  "Namespace": "amazon-cloudwatch",
  "NodeName": "ip-192-0-2-0.us-west-2.compute.internal",
  "PodName": "cloudwatch-agent-statsd",
  "Service": "cloudwatch-agent-statsd",
  "Sources": [
    "cadvisor",
    "calculated"
  ],
  "Timestamp": "1567097399912",
  "Type": "ContainerFS",
  "Version": "0",

  "device": "/dev/nvme0n1p1",
  "fstype": "vfs",
  "kubernetes": {
    "container_name": "cloudwatch-agent",
    "docker": {
      "container_id":
 "8967b6b37da239dfad197c9fdea3e5dfd35a8a759ec86e2e4c3f7b401e232706"
    },
    "host": "ip-192-168-75-26.us-west-2.compute.internal",
    "labels": {
      "app": "cloudwatch-agent-statsd",
      "pod-template-hash": "df44f855f"
    },
    "namespace_name": "amazon-cloudwatch",
    "pod_id": "2f4ff5ac-c813-11e9-a31d-06e9dde32928",
    "pod_name": "cloudwatch-agent-statsd-df44f855f-ts4q2",
    "pod_owners": [
      {
        "owner_kind": "Deployment",
        "owner_name": "cloudwatch-agent-statsd"
      }
    ],
    "service_name": "cloudwatch-agent-statsd"
  }
}
```


Tipo: Clúster

```
{
  "CloudWatchMetrics": [
    {
      "Metrics": [
        {
          "Unit": "Count",
          "Name": "cluster_node_count"
        },
        {
          "Unit": "Count",
          "Name": "cluster_failed_node_count"
        }
      ],
      "Dimensions": [
        [
          "ClusterName"
        ]
      ],
      "Namespace": "ContainerInsights"
    }
  ],
  "ClusterName": "myCICluster",
  "Sources": [
    "apiserver"
  ],
  "Timestamp": "1567097534160",
  "Type": "Cluster",
  "Version": "0",
  "cluster_failed_node_count": 0,
  "cluster_node_count": 3
}
```

Tipo: ClusterService

```
{
  "CloudWatchMetrics": [
    {
      "Metrics": [
        {
          "Unit": "Count",
          "Name": "service_number_of_running_pods"
        }
      ],
```

```
      "Dimensions": [
        [
          "Service",
          "Namespace",
          "ClusterName"
        ],
        [
          "ClusterName"
        ]
      ],
      "Namespace": "ContainerInsights"
    }
  ],
  "ClusterName": "myCICluster",
  "Namespace": "amazon-cloudwatch",
  "Service": "cloudwatch-agent-statsd",
  "Sources": [
    "apiserver"
  ],
  "Timestamp": "1567097534160",
  "Type": "ClusterService",
  "Version": "0",
  "kubernetes": {
    "namespace_name": "amazon-cloudwatch",
    "service_name": "cloudwatch-agent-statsd"
  },
  "service_number_of_running_pods": 1
}
```

Tipo: ClusterNamespace

```
{
  "CloudWatchMetrics": [
    {
      "Metrics": [
        {
          "Unit": "Count",
          "Name": "namespace_number_of_running_pods"
        }
      ],
      "Dimensions": [
        [
          "Namespace",
```

```
          "ClusterName"
        ],
        [
          "ClusterName"
        ]
      ],
      "Namespace": "ContainerInsights"
    }
  ],
  "ClusterName": "myCICluster",
  "Namespace": "amazon-cloudwatch",
  "Sources": [
    "apiserver"
  ],
  "Timestamp": "1567097594160",
  "Type": "ClusterNamespace",
  "Version": "0",
  "kubernetes": {
    "namespace_name": "amazon-cloudwatch"
  },
  "namespace_number_of_running_pods": 7
}
```

Campos relevantes en eventos de registro de rendimiento para Amazon EKS y Kubernetes

En Amazon EKS y Kubernetes, el agente de CloudWatch en contenedores emite los datos como eventos de registro de rendimiento. Esto permite a CloudWatch capturar y almacenar datos de alta cardinalidad. CloudWatch utiliza los datos en los eventos de registro de rendimiento para crear métricas de CloudWatch agregadas en el nivel de clúster, de nodo y de pod sin necesidad de perder detalles pormenorizados.

En la siguiente tabla, se muestran los campos de estos eventos de registro de rendimiento que son relevantes para la recopilación de datos de las métricas de Información de contenedores. Puede utilizar CloudWatch Logs Insights para consultar cualquiera de estos campos con el fin de recopilar datos o investigar problemas. Para obtener más información, consulte Analyze Log Data With CloudWatch Logs Insights (Análisis de datos de registro con CloudWatch Logs Insights).

Tipo	Campo de registro		Origen	Fórmula o notas
Pod	`pod_cpu_utilization`		Calculado	Fórmula: `pod_cpu_usage_total / node_cpu_limit`
Pod	`pod_cpu_usage_total` `pod_cpu_usage_total` se mide en milinúcleos.		cadvisor	
Pod	`pod_cpu_limit`		Calculado	Fórmula: `sum(container_cpu_limit)` `sum(container_cpu_limit)` incluye los pods ya completados. Si no se ha definido un límite de CPU para alguno de los contenedores del pod, este campo no aparece en el evento de registro. Esto incluye

Tipo	Campo de registro	Origen	Fórmula o notas
			los contenedores init.
Pod	`pod_cpu_request`	Calculado	Fórmula: `sum(container_cpu_request)` No se garantiza que `container_cpu_request` se vaya a establecer. En la suma solo se incluyen las que se establezcan.
Pod	`pod_cpu_utilization_over_pod_limit`	Calculado	Fórmula: `pod_cpu_usage_total / pod_cpu_limit`
Pod	`pod_cpu_reserved_capacity`	Calculado	Fórmula: `pod_cpu_request / node_cpu_limit`

Tipo	Campo de registro	Origen	Fórmula o notas
Pod	`pod_memory_utilization`	Calculado	Fórmula: `pod_memory_working_set / node_memory_limit` Es el porcentaje de uso de memoria de pod sobre la limitación de memoria de nodo.
Pod	`pod_memory_working_set`	cadvisor	

Tipo	Campo de registro	Origen	Fórmula o notas
Pod	`pod_memory_limit`	Calculado	Fórmula: `sum(container_memory_limit)` Si algún contenedor del pod no tiene definido un límite de memoria, este campo no aparece en el evento de registro. Esto incluye los contenedores init.

Tipo	Campo de registro	Origen	Fórmula o notas
Pod	`pod_memory_request`	Calculado	Fórmula: `sum(container_memory_request)` No se garantiza que `container_memory_request` se vaya a establecer. En la suma solo se incluyen las que se establezcan.

Tipo	Campo de registro	Origen	Fórmula o notas
Pod	`pod_memory_utilization_over_pod_limit`	Calculado	Fórmula: `pod_memory_working_set / pod_memory_limit` Si algún contenedor del pod no tiene definido un límite de memoria, este campo no aparece en el evento de registro. Esto incluye los contenedores init.
Pod	`pod_memory_reserved_capacity`	Calculado	Fórmula: `pod_memory_request / node_memory_limit`

Tipo	Campo de registro	Origen	Fórmula o notas
Pod	pod_network_tx_bytes	Calculado	Fórmula: sum(pod_interface_network_tx_bytes) Estos datos están disponibles para todas las interfaces de red de cada pod. El agente de CloudWatch calcula el total y agrega las reglas de extracción de las métricas.
Pod	pod_network_rx_bytes	Calculado	Fórmula: sum(pod_interface_network_rx_bytes)
Pod	pod_network_total_bytes	Calculado	Fórmula: pod_network_rx_bytes + pod_network_tx_bytes

Tipo	Campo de registro	Origen	Fórmula o notas
PodNet	`pod_interface_network_rx_bytes`	cadvisor	Este dato son los bytes rx de red por segundo de la interfaz de red de un pod.
PodNet	`pod_interface_network_tx_bytes`	cadvisor	Este dato son los bytes tx de red por segundo de la interfaz de red de un pod.
Contenedor	`container_cpu_usage_total`	cadvisor	
Contenedor	`container_cpu_limit`	cadvisor	No garantiza que se vaya a establecer. Si no se establece, no se emite.
Contenedor	`container_cpu_request`	cadvisor	No garantiza que se vaya a establecer. Si no se establece, no se emite.
Contenedor	`container_memory_working_set`	cadvisor	

Tipo	Campo de registro	Origen	Fórmula o notas
Contenedor	`container_memory_limit`	pod	No garantiza que se vaya a establecer. Si no se establece, no se emite.
Contenedor	`container_memory_request`	pod	No garantiza que se vaya a establecer. Si no se establece, no se emite.
Nodo	`node_cpu_utilization`	Calculado	Fórmula: `node_cpu_usage_total / node_cpu_limit`
Nodo	`node_cpu_usage_total`	cadvisor	
Nodo	`node_cpu_limit`	/proc	

Tipo	Campo de registro	Origen	Fórmula o notas
Nodo	`node_cpu_request`	Calculado	Fórmula: `sum(pod_cpu_request)` En el caso de los cronjobs, `node_cpu_request` también incluye las solicitudes de módulos completados. Esto puede generar un alto valor para `node_cpu_reserved_capacity` .
Nodo	`node_cpu_reserved_capacity`	Calculado	Fórmula: `node_cpu_request / node_cpu_limit`
Nodo	`node_memory_utilization`	Calculado	Fórmula: `node_memory_working_set / node_memory_limit`

Tipo	Campo de registro	Origen	Fórmula o notas
Nodo	node_memory_working_set	cadvisor	
Nodo	node_memory_limit	/proc	
Nodo	node_memory_request	Calculado	Fórmula: sum(pod_memory_request)
Nodo	node_memory_reserved_capacity	Calculado	Fórmula: node_memory_request / node_memory_limit
Nodo	node_network_rx_bytes	Calculado	Fórmula: sum(node_interface_network_rx_bytes)
Nodo	node_network_tx_bytes	Calculado	Fórmula: sum(node_interface_network_tx_bytes)
Nodo	node_network_total_bytes	Calculado	Fórmula: node_network_rx_bytes + node_network_tx_bytes

Tipo	Campo de registro	Origen	Fórmula o notas
Nodo	node_number_of_running_pods	Lista de pods	
Nodo	node_number_of_running_containers	Lista de pods	
NodeNet	node_interface_network_rx_bytes	cadvisor	Este dato son los bytes rx de red por segundo de la interfaz de red de un nodo de trabajo.
NodeNet	node_interface_network_tx_bytes	cadvisor	Este dato son los bytes tx de red por segundo de la interfaz de red de un nodo de trabajo.
NodeFS	node_filesystem_capacity	cadvisor	
NodeFS	node_filesystem_usage	cadvisor	

Tipo	Campo de registro	Origen	Fórmula o notas
NodeFS	`node_filesystem_utilization`	Calculado	Fórmula: `node_filesystem_usage / node_filesystem_capacity` Estos datos están disponibles para cada nombre de dispositivo.
Clúster	`cluster_failed_node_count`	Servidor de API	
Clúster	`cluster_node_count`	Servidor de API	
Servicio	`service_number_of_running_pods`	Servidor de API	
`Namespace`	`namespace_number_of_running_pods`	Servidor de API	

Ejemplos de cálculo de métricas

En esta sección, se incluyen ejemplos que muestran cómo se calculan algunos de los valores de la tabla anterior.

Suponga que tiene un clúster en el estado siguiente.

```
Node1
   node_cpu_limit = 4
   node_cpu_usage_total = 3
```

```
   Pod1
     pod_cpu_usage_total = 2

     Container1
        container_cpu_limit = 1
        container_cpu_request = 1
        container_cpu_usage_total = 0.8

     Container2
        container_cpu_limit = null
        container_cpu_request = null
        container_cpu_usage_total = 1.2

   Pod2
     pod_cpu_usage_total = 0.4

     Container3
        container_cpu_limit = 1
        container_cpu_request = 0.5
        container_cpu_usage_total = 0.4

Node2
   node_cpu_limit = 8
   node_cpu_usage_total = 1.5

   Pod3
     pod_cpu_usage_total = 1

     Container4
        container_cpu_limit = 2
        container_cpu_request = 2
        container_cpu_usage_total = 1
```

En la tabla siguiente, se muestra cómo se calculan las métricas de CPU de los pods utilizando estos datos.

Métrica	Fórmula	Pod1	Pod2	Pod3
`pod_cpu_utilization`	`pod_cpu_usage_total / node_cpu_limit`	2 / 4 = 50 %	0,4 / 4 = 10 %	1 / 8 = 12,5 %

Métrica	Fórmula	Pod1	Pod2	Pod3
`pod_cpu_utilization_over_pod_limit`	`pod_cpu_usage_total / sum(container_cpu_limit)`	N/A, porque no se ha definido el límite de CPU para `Container 2` .	0,4 / 1 = 40 %	1 / 2 = 50 %
`pod_cpu_reserved_capacity`	`sum(container_cpu_request) / node_cpu_limit`	(1 + 0) / 4 = 25 %	0,5 / 4 = 12,5 %	2 / 8 = 25 %

En la tabla siguiente, se muestra cómo se calculan las métricas de CPU de los nodos utilizando estos datos.

Métrica	Fórmula	Node1	Node2
`node_cpu_utilization`	`node_cpu_usage_total / node_cpu_limit`	3 / 4 = 75 %	1,5 / 8 = 18,75 %
`node_cpu_reserved_capacity`	`sum(pod_cpu_request) / node_cpu_limit`	1,5 / 4 = 37,5 %	2 / 8 = 25 %

Supervisión de métricas de Información de contenedores de Prometheus

La supervisión de Información de contenedores de CloudWatch para Prometheus automatiza la detección de métricas de Prometheus de cargas de trabajo y de sistemas en contenedores. Prometheus es un conjunto de herramientas de alerta y supervisión de sistemas de código abierto. Para obtener más información, consulte s Prometheus? en la documentación de Prometheus.

La detección de métricas de Prometheus es compatible con los clústeres de Amazon Elastic Container Service, Amazon Elastic Kubernetes Service y Kubernetes que se ejecutan en instancias de Amazon EC2. Se recopilan los tipos de métricas de contador, de medición y de resumen de

Prometheus. La compatibilidad con las métricas de histograma está prevista para una próxima versión.

Para los clústeres de Amazon ECS y de Amazon EKS, se admiten los tipos de lanzamiento de EC2 y Fargate. Información de contenedores recopila automáticamente métricas de varias cargas de trabajo, y puede configurarse para recopilar métricas de cualquier carga de trabajo.

Puede adoptar Prometheus como método de código abierto y estándar abierto para capturar métricas personalizadas en CloudWatch. El agente de CloudWatch compatible con Prometheus detecta y recopila métricas de Prometheus para supervisar, solucionar errores y crear alarmas con más rapidez cuando el rendimiento de las aplicaciones se ve degradado y existen errores. Esto reduce también el número de herramientas de supervisión necesarias para mejorar la capacidad de observación.

Información de contenedores de Prometheus permite utilizar un sistema de pago por uso con las métricas y registros, incluida su recopilación, almacenamiento y análisis. Para obtener más información, consulte Precios de Amazon CloudWatch.

Paneles estándar para algunas cargas de trabajo

La solución Información de contenedores de Prometheus incluye paneles estándar para las cargas de trabajo populares que se enumeran en esta sección. Para obtener configuraciones de ejemplo para estas cargas de trabajo, consulte (Opcional) Configure de cargas de trabajo en contenedores de Amazon ECS de muestra para realizar pruebas con las métricas de Prometheus y (Opcional) Configure las cargas de trabajo de muestra de Amazon EKS en contenedores para realizar pruebas con las métricas de Prometheus.

También puede configurar Información de contenedores para que recopile métricas de Prometheus de otras aplicaciones y servicios en contenedores mediante la edición del archivo de configuración del agente.

Las cargas de trabajo con paneles prediseñados para clústeres de Amazon EKS y de Kubernetes que se ejecutan en instancias de Amazon EC2:

- AWS App Mesh
- NGINX
- Memcached
- Java/JMX
- HAProxy

Cargas de trabajo con paneles prediseñados para clústeres de Amazon ECS:

- AWS App Mesh
- Java/JMX
- NGINX
- NGINX Plus

Instale y configure la recopilación de métricas de Prometheus en clústeres de Amazon ECS

Para recopilar métricas de Prometheus de clústeres de Amazon ECS, se puede utilizar el agente de CloudWatch como recopilador o utilizar el recopilador de AWS Distro para OpenTelemetry. Para obtener información sobre el uso del recopilador AWS Distro para OpenTelemetry, consulte https://aws-otel.github.io/docs/getting-started/container-insights/ecs-prometheus.

En las siguientes secciones se explica cómo se utiliza el agente de CloudWatch como recopilador para recuperar métricas de Prometheus. Se instala el agente CloudWatch con supervisión de Prometheus en clústeres que ejecutan Amazon ECS y, opcionalmente, se puede configurar el agente para que raspe destinos adicionales. Estas secciones también proporcionan tutoriales opcionales para configurar cargas de trabajo de muestra con el fin de utilizarlas en pruebas con supervisión de Prometheus.

Información de contenedores en Amazon ECS admite las siguientes combinaciones de tipo de lanzamiento y modo de redes para las métricas de Prometheus:

Tipo de lanzamiento de Amazon ECS	Modos de redes compatibles
EC2 (Linux)	puente, host y awsvpc
Fargate	awsvpc

Requisitos del grupo de seguridad de la VPC

Las reglas de entrada de los grupos de seguridad para las cargas de trabajo de Prometheus deben abrir los puertos de Prometheus al agente de CloudWatch para raspar las métricas de Prometheus por la IP privada.

Las reglas de salida del grupo de seguridad para el agente de CloudWatch deben permitir que el agente de CloudWatch se conecte al puerto de cargas de trabajo de Prometheus mediante la IP privada.

Temas

Instale el agente CloudWatch con la colección de métricas de Prometheus en clústeres de Amazon ECS

En esta sección se explica cómo se configura el agente de CloudWatch con la supervisión de Prometheus en un clúster que ejecute Amazon ECS. Después de hacerlo, el agente raspa e importa automáticamente métricas para las siguientes cargas de trabajo que se ejecutan en ese clúster.

- AWS App Mesh
- Java/JMX

También puede configurar el agente para que recopile e importe métricas de fuentes y cargas de trabajo adicionales de Prometheus.

Configuración de los roles de IAM

Se necesitan dos roles de IAM para la definición de la tarea del agente de CloudWatch. Si especifica **`CreateIAMRoles=True`** en la pila de AWS CloudFormation para que Información de contenedores cree estos roles por usted, los roles se crearán con los permisos correctos. Si desea crearlos o usar roles existentes, se requieren los siguientes roles y permisos.

- Rol de tarea de ECS del agente de CloudWatch: el contenedor del agente de CloudWatch utiliza este rol. Debe incluirse la política CloudWatchAgentServerPolicy y una política administrada por el cliente que contiene los siguientes permisos que son solo de lectura:
  - `ec2:DescribeInstances`
  - `ecs:ListTasks`
  - `ecs:ListServices`

  - `ecs:DescribeContainerInstances`
  - `ecs:DescribeServices`
  - `ecs:DescribeTasks`
  - `ecs:DescribeTaskDefinition`
- Rol de ejecución de tareas de ECS del agente de CloudWatch: es el rol que Amazon ECS requiere para lanzar y ejecutar los contenedores. Asegúrese de que el rol de ejecución de la tarea tenga adjuntas las políticas AmazonSSMReadOnlyAccess, AmazonECSTaskExecutionRolePolicy, y .CloudWatchAgentServerPolicy. Si desea almacenar más información confidencial para que Amazon ECS la use, consulte Specifying sensitive data (Especificación de información confidencial).

Instalación del agente de CloudWatch con la supervisión de Prometheus mediante AWS CloudFormation

Se utiliza AWS CloudFormation para instalar el agente CloudWatch con supervisión de Prometheus para clústeres de Amazon ECS. En la siguiente lista, se enumeran los parámetros que utilizará en la plantilla de AWS CloudFormation.

- ECSClusterName: especifica el clúster de Amazon ECS de destino.
- CreateIAMRoles: especifique **`True`** para crear roles nuevos para el rol de tareas de Amazon ECS y el rol de ejecución de tareas de Amazon ECS. Especifique **`False`** para reutilizar los roles existentes.
- TaskRoleName: si ha especificado **`True`** en CreateIAMRoles, esto especifica el nombre que debe usarse para el nuevo rol de tareas de Amazon ECS. Si ha especificado **`False`** en CreateIAMRoles, esto especifica el rol existente que se va a utilizar como función de tarea de Amazon ECS.
- ExecutionRoleName: si ha especificado **`True`** en CreateIAMRoles, esto especifica el nombre que debe usarse para el nuevo rol de ejecución de tareas de Amazon ECS. Si ha especificado **`False`** en CreateIAMRoles, esto especifica el rol existente que se va a utilizar como función de ejecución de tareas de Amazon ECS.
- ECSNetworkMode: si utiliza el tipo de lanzamiento EC2, especifique aquí el modo de redes. Debe ser **`bridge`** o **`host`**.
- ECSLaunchType: especifique **`fargate`** o **`EC2`**.
- SecurityGroupID: si el ECSNetworkMode es **`awsvpc`**, especifique aquí el ID del grupo de seguridad.

- SubnetID: si el ECSNetworkMode es **awsvpc**, especifique aquí el ID de la subred.

Ejemplos de comandos

En esta sección se incluyen ejemplos de comandos de AWS CloudFormation para instalar Información de contenedores con supervisión de Prometheus en varias situaciones.

Cree una pila de AWS CloudFormation para un clúster de Amazon ECS en modo de red bridge

```
export AWS_PROFILE=your_aws_config_profile_eg_default
export AWS_DEFAULT_REGION=your_aws_region_eg_ap-southeast-1
export ECS_CLUSTER_NAME=your_ec2_ecs_cluster_name
export ECS_NETWORK_MODE=bridge
export CREATE_IAM_ROLES=True
export ECS_TASK_ROLE_NAME=your_selected_ecs_task_role_name
export ECS_EXECUTION_ROLE_NAME=your_selected_ecs_execution_role_name

curl -O https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
insights/latest/ecs-task-definition-templates/deployment-mode/replica-service/cwagent-
prometheus/cloudformation-quickstart/cwagent-ecs-prometheus-metric-for-bridge-host.yaml

aws cloudformation create-stack --stack-name CWAgent-Prometheus-ECS-
${ECS_CLUSTER_NAME}-EC2-${ECS_NETWORK_MODE} \
    --template-body file://cwagent-ecs-prometheus-metric-for-bridge-host.yaml \
    --parameters ParameterKey=ECSClusterName,ParameterValue=${ECS_CLUSTER_NAME} \
                 ParameterKey=CreateIAMRoles,ParameterValue=${CREATE_IAM_ROLES} \
                 ParameterKey=ECSNetworkMode,ParameterValue=${ECS_NETWORK_MODE} \
                 ParameterKey=TaskRoleName,ParameterValue=${ECS_TASK_ROLE_NAME} \
                 ParameterKey=ExecutionRoleName,ParameterValue=
${ECS_EXECUTION_ROLE_NAME} \
    --capabilities CAPABILITY_NAMED_IAM \
    --region ${AWS_DEFAULT_REGION} \
    --profile ${AWS_PROFILE}
```

Cree una pila de AWS CloudFormation para un clúster de Amazon ECS en modo de red host

```
export AWS_PROFILE=your_aws_config_profile_eg_default
export AWS_DEFAULT_REGION=your_aws_region_eg_ap-southeast-1
export ECS_CLUSTER_NAME=your_ec2_ecs_cluster_name
export ECS_NETWORK_MODE=host
export CREATE_IAM_ROLES=True
export ECS_TASK_ROLE_NAME=your_selected_ecs_task_role_name
```

```
export ECS_EXECUTION_ROLE_NAME=your_selected_ecs_execution_role_name


curl -O https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
insights/latest/ecs-task-definition-templates/deployment-mode/replica-service/cwagent-
prometheus/cloudformation-quickstart/cwagent-ecs-prometheus-metric-for-bridge-host.yaml

aws cloudformation create-stack --stack-name CWAgent-Prometheus-ECS-
${ECS_CLUSTER_NAME}-EC2-${ECS_NETWORK_MODE} \
    --template-body file://cwagent-ecs-prometheus-metric-for-bridge-host.yaml \
    --parameters ParameterKey=ECSClusterName,ParameterValue=${ECS_CLUSTER_NAME} \
                 ParameterKey=CreateIAMRoles,ParameterValue=${CREATE_IAM_ROLES} \
                 ParameterKey=ECSNetworkMode,ParameterValue=${ECS_NETWORK_MODE} \
                 ParameterKey=TaskRoleName,ParameterValue=${ECS_TASK_ROLE_NAME} \
                 ParameterKey=ExecutionRoleName,ParameterValue=
${ECS_EXECUTION_ROLE_NAME} \
    --capabilities CAPABILITY_NAMED_IAM \
    --region ${AWS_DEFAULT_REGION} \
    --profile ${AWS_PROFILE}
```

Cree una pila de AWS CloudFormation para un clúster de Amazon ECS en modo de red awsvpc

```
export AWS_PROFILE=your_aws_config_profile_eg_default
export AWS_DEFAULT_REGION=your_aws_region_eg_ap-southeast-1
export ECS_CLUSTER_NAME=your_ec2_ecs_cluster_name
export ECS_LAUNCH_TYPE=EC2
export CREATE_IAM_ROLES=True
export ECS_CLUSTER_SECURITY_GROUP=your_security_group_eg_sg-xxxxxxxxxx
export ECS_CLUSTER_SUBNET=your_subnet_eg_subnet-xxxxxxxxxx
export ECS_TASK_ROLE_NAME=your_selected_ecs_task_role_name
export ECS_EXECUTION_ROLE_NAME=your_selected_ecs_execution_role_name

curl -O https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
insights/latest/ecs-task-definition-templates/deployment-mode/replica-service/cwagent-
prometheus/cloudformation-quickstart/cwagent-ecs-prometheus-metric-for-awsvpc.yaml

aws cloudformation create-stack --stack-name CWAgent-Prometheus-ECS-
${ECS_CLUSTER_NAME}-${ECS_LAUNCH_TYPE}-awsvpc \
    --template-body file://cwagent-ecs-prometheus-metric-for-awsvpc.yaml \
    --parameters ParameterKey=ECSClusterName,ParameterValue=${ECS_CLUSTER_NAME} \
                 ParameterKey=CreateIAMRoles,ParameterValue=${CREATE_IAM_ROLES} \
                 ParameterKey=ECSLaunchType,ParameterValue=${ECS_LAUNCH_TYPE} \
```

```
                ParameterKey=SecurityGroupID,ParameterValue=
${ECS_CLUSTER_SECURITY_GROUP} \
                ParameterKey=SubnetID,ParameterValue=${ECS_CLUSTER_SUBNET} \
                ParameterKey=TaskRoleName,ParameterValue=${ECS_TASK_ROLE_NAME} \
                ParameterKey=ExecutionRoleName,ParameterValue=
${ECS_EXECUTION_ROLE_NAME} \
    --capabilities CAPABILITY_NAMED_IAM \
    --region ${AWS_DEFAULT_REGION} \
    --profile ${AWS_PROFILE}
```

Creación de una pila de AWS CloudFormation para un clúster de Fargate en modo de red awsvpc

```
export AWS_PROFILE=your_aws_config_profile_eg_default
export AWS_DEFAULT_REGION=your_aws_region_eg_ap-southeast-1
export ECS_CLUSTER_NAME=your_ec2_ecs_cluster_name
export ECS_LAUNCH_TYPE=FARGATE
export CREATE_IAM_ROLES=True
export ECS_CLUSTER_SECURITY_GROUP=your_security_group_eg_sg-xxxxxxxxxx
export ECS_CLUSTER_SUBNET=your_subnet_eg_subnet-xxxxxxxxxx
export ECS_TASK_ROLE_NAME=your_selected_ecs_task_role_name
export ECS_EXECUTION_ROLE_NAME=your_selected_ecs_execution_role_name

curl -O https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
insights/latest/ecs-task-definition-templates/deployment-mode/replica-service/cwagent-
prometheus/cloudformation-quickstart/cwagent-ecs-prometheus-metric-for-awsvpc.yaml

aws cloudformation create-stack --stack-name CWAgent-Prometheus-ECS-
${ECS_CLUSTER_NAME}-${ECS_LAUNCH_TYPE}-awsvpc \
    --template-body file://cwagent-ecs-prometheus-metric-for-awsvpc.yaml \
    --parameters ParameterKey=ECSClusterName,ParameterValue=${ECS_CLUSTER_NAME} \
                ParameterKey=CreateIAMRoles,ParameterValue=${CREATE_IAM_ROLES} \
                ParameterKey=ECSLaunchType,ParameterValue=${ECS_LAUNCH_TYPE} \
                ParameterKey=SecurityGroupID,ParameterValue=
${ECS_CLUSTER_SECURITY_GROUP} \
                ParameterKey=SubnetID,ParameterValue=${ECS_CLUSTER_SUBNET} \
                ParameterKey=TaskRoleName,ParameterValue=${ECS_TASK_ROLE_NAME} \
                ParameterKey=ExecutionRoleName,ParameterValue=
${ECS_EXECUTION_ROLE_NAME} \
    --capabilities CAPABILITY_NAMED_IAM \
    --region ${AWS_DEFAULT_REGION} \
    --profile ${AWS_PROFILE}
```

Recursos de AWS que crea la pila de AWS CloudFormation

En la siguiente tabla se enumeran los recursos de AWS que se crean cuando se utiliza AWS CloudFormation para configurar Información de contenedores con supervisión de Prometheus en un clúster de Amazon ECS.

Tipo de recurso	Nombre del recurso	Comentarios
AWS::SSM::Parameter	AmazonCloudWatch-CWAgentConfig-$*ECS_CLUSTER_NAME* -$*ECS_LAUNCH_TYPE* -$*ECS_NETWORK_MODE*	Este es el agente de CloudWatch con la definición predeterminada de formato de métrica integrada de App Mesh y Java/JMX.
AWS::SSM::Parameter	AmazonCloudWatch-Prometheus ConfigName-$*ECS_CLUSTER_NAME* -$*ECS_LAUNCH_TYPE* -$*ECS_NETWORK_MODE*	Esta es la configuración de raspado de Prometheus.
AWS::IAM::Role	$ECS_TASK_ROLE_NAME.	Rol de tarea de Amazon ECS. Esto se crea solo si ha especificado **`True`** para `CREATE_IAM_ROLES` .
AWS::IAM::Role	${ECS_EXECUTION_ROLE_NAME}	Rol de ejecución de tareas de Amazon ECS. Esto se crea solo si ha especificado **`True`** para `CREATE_IAM_ROLES` .
AWS::ECS::TaskDefinition	cwagent-prometheus-$*ECS_CLUSTER_NAME* -$*ECS_LAUNCH_TYPE* -$*ECS_NETWORK_MODE*	
AWS::ECS::Service	cwagent-prometheus-replica-service-$*ECS_LAUNCH_TYPE* -$*ECS_NETWORK_MODE*	

Eliminación de la pila de AWS CloudFormation para el agente de CloudWatch con supervisión de Prometheus

Para eliminar el agente de CloudWatch de un clúster de Amazon ECS, ingrese estos comandos.

```
export AWS_PROFILE=your_aws_config_profile_eg_default
export AWS_DEFAULT_REGION=your_aws_region_eg_ap-southeast-1
export CLOUDFORMATION_STACK_NAME=your_cloudformation_stack_name

aws cloudformation delete-stack \
--stack-name ${CLOUDFORMATION_STACK_NAME} \
--region ${AWS_DEFAULT_REGION} \
--profile ${AWS_PROFILE}
```

Paspado de fuentes adicionales de Prometheus e importación de esas métricas

El agente CloudWatch con supervisión de Prometheus necesita dos configuraciones para raspar las métricas de Prometheus. Una de ellas es para las configuraciones estándar de Prometheus que como se documenta en <scrape_config> en la documentación de Prometheus. La otra configuración es para la configuración del agente de CloudWatch.

Para los clústeres de Amazon ECS, las configuraciones se integran con Parameter Store de AWS Systems Manager según los secretos de la definición de tareas de Amazon ECS:

- El secreto `PROMETHEUS_CONFIG_CONTENT` es para la configuración de raspado de Prometheus.
- El secreto `CW_CONFIG_CONTENT` es para la configuración del agente de CloudWatch.

Para raspar las fuentes adicionales de las métricas de Prometheus e importarlas a CloudWatch, debe modificar tanto la configuración de raspado de Prometheus como la configuración del agente de CloudWatch y, a continuación, debe volver a implementar el agente con la configuración actualizada.

Requisitos del grupo de seguridad de la VPC

Las reglas de entrada de los grupos de seguridad para las cargas de trabajo de Prometheus deben abrir los puertos de Prometheus al agente de CloudWatch para raspar las métricas de Prometheus por la IP privada.

Las reglas de salida del grupo de seguridad para el agente de CloudWatch deben permitir que el agente de CloudWatch se conecte al puerto de cargas de trabajo de Prometheus mediante la IP privada.

Configuración de raspado de Prometheus

El agente de CloudWatch es compatible con la configuración de raspado estándar de Prometheus como se describe en <scrape_config> en la documentación de Prometheus. Se puede editar esta sección para actualizar las configuraciones que ya están en este archivo y agregar destinos adicionales de raspado de Prometheus. De forma predeterminada, el archivo de configuración de muestra contiene las siguientes líneas de configuración global:

```
global:
  scrape_interval: 1m
  scrape_timeout: 10s
```

- scrape_interval: define la frecuencia con la que se deben raspar los destinos.
- scrape_timeout: define cuánto tiempo debe esperar antes de que se agote el tiempo de espera de una petición de raspado.

También puede definir valores diferentes para estos parámetros en el nivel de trabajo, para anular las configuraciones globales.

Trabajos de raspado de Prometheus

Los archivos YAML del agente de CloudWatch ya tienen algunos trabajos de raspado configurados de forma predeterminada. Por ejemplo, en los archivos YAML para Amazon ECS, como `cwagent-ecs-prometheus-metric-for-bridge-host.yaml`, los trabajos de raspado predeterminados se configuran en la sección `ecs_service_discovery`.

```
"ecs_service_discovery": {
                  "sd_frequency": "1m",
                  "sd_result_file": "/tmp/cwagent_ecs_auto_sd.yaml",
                  "docker_label": {
                  },
                  "task_definition_list": [
                    {
                      "sd_job_name": "ecs-appmesh-colors",
                      "sd_metrics_ports": "9901",
```

```
                    "sd_task_definition_arn_pattern": ".*:task-definition\/.*-
ColorTeller-(white):[0-9]+",
                    "sd_metrics_path": "/stats/prometheus"
                  },
                  {
                    "sd_job_name": "ecs-appmesh-gateway",
                    "sd_metrics_ports": "9901",
                    "sd_task_definition_arn_pattern": ".*:task-definition/.*-
ColorGateway:[0-9]+",
                    "sd_metrics_path": "/stats/prometheus"
                  }
                ]
              }
```

Cada uno de estos destinos predeterminados se raspan y las métricas se envían a CloudWatch en eventos de registro mediante un formato de métrica integrada. Para obtener más información, consulte Incrustar métricas en los registros.

Los eventos de registro de los clústeres de Amazon ECS se almacenan en el grupo de registros / aws/ecs/containerinsights/***cluster_name***/prometheus.

Cada trabajo de extracción está contenido en un flujo de registros diferente en este grupo de registros.

Para agregar un nuevo destino de raspado, se debe agregar una entrada nueva en la sección `task_definition_list` en la sección `ecs_service_discovery` del archivo YAML y reiniciar el agente. Para obtener un ejemplo de este proceso, consulte Tutorial para agregar un destino de raspado nuevo de Prometheus: métricas del servidor de la API de Prometheus.

Configuración del agente de CloudWatch para Prometheus

El archivo de configuración del agente de CloudWatch cuenta con una sección `prometheus` en `metrics_collected` para la configuración de raspado de Prometheus. Incluye las siguientes opciones de configuración:

- nombre_clúster: especifica el nombre del clúster que se va a agregar como etiqueta en el evento de registro. Este campo es opcional. Si lo omite, el agente puede detectar el nombre del clúster de Amazon ECS.
- log_group_name: especifica el nombre del grupo de registros para las métricas de Prometheus raspadas. Este campo es opcional. Si lo omite, CloudWatch usa /aws/ecs/containerinsights/***cluster_name***/prometheus para los registros de clústeres de Amazon ECS.

- prometheus_config_path: especifica la ruta del archivo de configuración de raspado de Prometheus. Si el valor de este campo comienza con `env:`, el contenido del archivo de configuración de raspado de Prometheus se recuperará de la variable de entorno del contenedor. No cambie este campo.
- ecs_service_discovery: es la sección para especificar las configuraciones de las funciones de detección automática de destino de Prometheus de Amazon ECS . Se admiten dos modos para detectar los destinos de Prometheus: detección basada en la etiqueta de docker del contenedor o detección basada en la expresión regular ARN de definición de tarea de Amazon ECS. Puede utilizar los dos modos juntos y el agente de CloudWatch desduplicará los destinos detectados en función de: {private_ip}:{port}/{metrics_path}.

  La sección `ecs_service_discovery` puede incluir los siguientes campos:

  - `sd_frequency` es la frecuencia para detectar a los exportadores de Prometheus. Especifique un número y un sufijo de unidad. Por ejemplo, `1m` para una vez por minuto o `30s` para una vez cada 30 segundos. Los sufijos de unidad válidos son `ns`, `us`, `ms`, `s`, `m` y `h`.

    Este campo es opcional. El valor predeterminado es 60 segundos (1 minuto).
  - `sd_target_cluster` es el nombre de clúster de Amazon ECS de destino para la detección automática. Este campo es opcional. El valor predeterminado es el nombre del clúster de Amazon ECS donde está instalado el agente de CloudWatch.
  - `sd_cluster_region` es la Región del clúster de Amazon ECS de destino. Este campo es opcional. El valor predeterminado es la Región del clúster de Amazon ECS donde está instalado el agente de CloudWatch.
  - `sd_result_file` es la ruta del archivo YAML para los resultados de destino de Prometheus. La configuración de raspado de Prometheus hará referencia a este archivo.
  - `docker_label` es una sección opcional que se puede utilizar para especificar la configuración para la detección de servicios basada en etiquetas docker. Si omite esta sección, no se utiliza la detección basada en etiquetas docker. Esta sección puede incluir los siguientes campos:
    - `sd_port_label` es el nombre de etiqueta docker del contenedor que especifica el puerto del contenedor para las métricas de Prometheus. El valor predeterminado es `ECS_PROMETHEUS_EXPORTER_PORT`. Si el contenedor no tiene esta etiqueta docker, el agente de CloudWatch lo omitirá.
    - `sd_metrics_path_label` es el nombre de etiqueta docker del contenedor que especifica la ruta de métricas de Prometheus. El valor predeterminado es

ECS_PROMETHEUS_METRICS_PATH. Si el contenedor no tiene esta etiqueta docker, el agente asume la ruta predeterminada /metrics.

- sd_job_name_label es el nombre de etiqueta docker del contenedor que especifica el nombre del trabajo de raspado de Prometheus. El valor predeterminado es job. Si el contenedor no tiene esta etiqueta docker, el agente de CloudWatch utiliza el nombre del trabajo en la configuración de raspado de Prometheus.

- task_definition_list es una sección opcional que se puede utilizar para especificar la configuración de la detección de servicios basada en definiciones de tareas. Si omite esta sección, no se utiliza la detección basada en definiciones de tareas. Esta sección puede incluir los siguientes campos:
  - sd_task_definition_arn_pattern es el patrón que se utiliza para especificar las definiciones de tareas de Amazon ECS que se van a detectar. Esta es una expresión regular.
  - sd_metrics_ports enumera el containerPort para las métricas de Prometheus. Separe los containerPorts con punto y coma.
  - sd_container_name_pattern especifica los nombres de contenedor de tareas de Amazon ECS. Esta es una expresión regular.
  - sd_metrics_path especifica la ruta de la métrica de Prometheus. Si omite esto, el agente asume la ruta de acceso predeterminada /metrics
  - sd_job_name especifica el nombre del trabajo de raspado de Prometheus. Si omite este campo, el agente de CloudWatch utilizará el nombre de trabajo en la configuración de borrado de Prometheus.
- service_name_list_for_tasks es una sección opcional que puede utilizar para especificar la configuración de la detección basada en nombres de servicio. Si omite esta sección, no se utiliza la detección basada en nombres de servicio. Esta sección puede incluir los siguientes campos:
  - sd_service_name_pattern es el patrón que se debe utilizar para especificar el servicio Amazon ECS en el que se van a detectar las tareas. Esta es una expresión regular.
  - sd_metrics_ports enumera el containerPort para ver las métricas de Prometheus. Separe varios containerPorts con punto y coma.
  - sd_container_name_pattern especifica los nombres de contenedor de tareas de Amazon ECS. Esta es una expresión regular.
  - sd_metrics_path especifica la ruta de las métricas de Prometheus. Si omite esto, el agente asume la ruta de acceso predeterminada /metrics.

    - `sd_job_name` especifica el nombre del trabajo de raspado de Prometheus. Si omite este campo, el agente de CloudWatch utilizará el nombre de trabajo en la configuración de raspado de Prometheus.

- metric_declaration: son secciones que especifican la matriz de registros con formato de métrica integrada que se van a generar. Hay secciones `metric_declaration` para cada fuente de Prometheus desde las que el agente de CloudWatch importa de forma predeterminada. Cada una de estas secciones incluye los siguientes campos:
    - `label_matcher` es una expresión regular que verifica el valor de las etiquetas que aparecen en `source_labels`. Las métricas que concuerdan se pueden incorporar al formato de métrica integrada que se envía a CloudWatch.

      Si tiene varias etiquetas especificadas en `source_labels`, se recomienda que evite el uso de los caracteres ^ o $ en la expresión regular para `label_matcher`.
    - `source_labels` especifica el valor de las etiquetas que se comprueban con `label_matcher`.
    - `label_separator` especifica el separador que se utilizará en la línea `label_matcher` si se especifican múltiples `source_labels`. El valor predeterminado es `;`. Puede ver este valor predeterminado utilizado en la línea `label_matcher` en el siguiente ejemplo.
    - `metric_selectors` es una expresión regular que especifica las métricas que se van a recopilar y enviar a CloudWatch.
    - `dimensions` es la lista de etiquetas que se van a utilizar como dimensiones de CloudWatch en cada métrica seleccionada.

Consulte el siguiente ejemplo, `metric_declaration`.

```
"metric_declaration": [
  {
     "source_labels":[ "Service", "Namespace"],
     "label_matcher":"(.*node-exporter.*|.*kube-dns.*);kube-system$",
     "dimensions":[
        ["Service", "Namespace"]
     ],
     "metric_selectors":[
        "^coredns_dns_request_type_count_total$"
     ]
  }
]
```

En este ejemplo se configura una sección de formato de métricas integradas para que se envíe como evento de registro si se cumplen las condiciones siguientes:

- El valor de `Service` contiene `node-exporter` o `kube-dns`.
- El valor de `Namespace` es `kube-system`.
- La métrica de Prometheus `coredns_dns_request_type_count_total` contiene ambas etiquetas, `Namespace` y `Service`.

El evento de registro que se envía incluye la siguiente sección resaltada:

```
{
   "CloudWatchMetrics":[
      {
         "Metrics":[
            {
               "Name":"coredns_dns_request_type_count_total"
            }
         ],
         "Dimensions":[
            [
               "Namespace",
               "Service"
            ]
         ],
         "Namespace":"ContainerInsights/Prometheus"
      }
   ],
   "Namespace":"kube-system",
   "Service":"kube-dns",
   "coredns_dns_request_type_count_total":2562,
   "eks_amazonaws_com_component":"kube-dns",
   "instance":"192.168.61.254:9153",
   "job":"kubernetes-service-endpoints",
   ...
}
```

Guía detallada para la detección automática en clústeres de Amazon ECS

Prometheus proporciona docenas de mecanismos dinámicos de detección de servicios, como se describe en <scrape_config>. Sin embargo, no existe una detección de servicios integrado para Amazon ECS. El agente CloudWatch agrega este mecanismo.

Cuando se habilita la detección de servicios de Amazon ECS Prometheus, el agente de CloudWatch realiza periódicamente las siguientes llamadas a la API a Amazon ECS y a los frontends de Amazon EC2 para recuperar los metadatos de las tareas ECS en ejecución en el clúster de ECS de destino.

```
EC2:DescribeInstances
ECS:ListTasks
ECS:ListServices
ECS:DescribeContainerInstances
ECS:DescribeServices
ECS:DescribeTasks
ECS:DescribeTaskDefinition
```

El agente de CloudWatch utiliza los metadatos para examinar los destinos de Prometheus dentro del clúster de ECS. El agente de CloudWatch admite tres modos de detección de servicio:

- Detección de servicio basada en etiquetas docker de contenedor
- Detección de servicio basada en expresiones regulares ARN de definición de tarea de ECS
- Detección de servicio basada en expresiones regulares de nombre de servicio de ECS

Todos los modos se pueden utilizar de forma conjunta. El agente de CloudWatch desduplica los destinos detectados en función de: `{private_ip}:{port}/{metrics_path}`.

Todos los destinos detectados se registran en un archivo de resultados que el campo de configuración `sd_result_file` especifica dentro del contenedor del agente de CloudWatch. A continuación se muestra un archivo de resultados de ejemplo.

```
- targets:
  - 10.6.1.95:32785
  labels:
    __metrics_path__: /metrics
    ECS_PROMETHEUS_EXPORTER_PORT: "9406"
    ECS_PROMETHEUS_JOB_NAME: demo-jar-ec2-bridge-dynamic
    ECS_PROMETHEUS_METRICS_PATH: /metrics
    InstanceType: t3.medium
    LaunchType: EC2
    SubnetId: subnet-123456789012
    TaskDefinitionFamily: demo-jar-ec2-bridge-dynamic-port
    TaskGroup: family:demo-jar-ec2-bridge-dynamic-port
    TaskRevision: "7"
    VpcId: vpc-01234567890
```

```
    container_name: demo-jar-ec2-bridge-dynamic-port
    job: demo-jar-ec2-bridge-dynamic
- targets:
  - 10.6.3.193:9404
  labels:
    __metrics_path__: /metrics
    ECS_PROMETHEUS_EXPORTER_PORT_SUBSET_B: "9404"
    ECS_PROMETHEUS_JOB_NAME: demo-tomcat-ec2-bridge-mapped-port
    ECS_PROMETHEUS_METRICS_PATH: /metrics
    InstanceType: t3.medium
    LaunchType: EC2
    SubnetId: subnet-123456789012
    TaskDefinitionFamily: demo-tomcat-ec2-bridge-mapped-port
    TaskGroup: family:demo-jar-tomcat-bridge-mapped-port
    TaskRevision: "12"
    VpcId: vpc-01234567890
    container_name: demo-tomcat-ec2-bridge-mapped-port
    job: demo-tomcat-ec2-bridge-mapped-port
```

Puede integrar directamente este archivo de resultados con la detección de servicios basada en archivos de Prometheus. Para obtener más información acerca de la detección de servicios basada en archivos de Prometheus, consulte <file_sd_config>.

Suponga que el archivo de resultados se registra en `/tmp/cwagent_ecs_auto_sd.yaml`, la siguiente configuración de raspado de Prometheus lo consumirá.

```
global:
  scrape_interval: 1m
  scrape_timeout: 10s
scrape_configs:
  - job_name: cwagent-ecs-file-sd-config
    sample_limit: 10000
    file_sd_configs:
      - files: [ "/tmp/cwagent_ecs_auto_sd.yaml" ]
```

El agente de CloudWatch también agrega las siguientes etiquetas adicionales para los destinos detectados.

- `container_name`
- `TaskDefinitionFamily`
- `TaskRevision`

- `TaskGroup`
- `StartedBy`
- `LaunchType`
- `job`
- `__metrics_path__`
- Etiquetas docker

Cuando el clúster tiene el tipo de lanzamiento EC2, se agregan las tres etiquetas siguientes.

- `InstanceType`
- `VpcId`
- `SubnetId`

> **Note**
>
> Las etiquetas docker que no concuerdan con la expresión regular `[a-zA-Z_][a-zA-Z0-9_]*` se filtran. Coincide con las convenciones de Prometheus enumeradas en `label_name` en Configuration file (Archivo de configuración) en la documentación de Prometheus.

Ejemplos de configuración de detección de servicios de ECS

En esta sección se incluyen ejemplos que demuestran la detección de servicios de ECS.

Ejemplo 1

```
"ecs_service_discovery": {
  "sd_frequency": "1m",
  "sd_result_file": "/tmp/cwagent_ecs_auto_sd.yaml",
  "docker_label": {
  }
}
```

En este ejemplo se habilita la detección de servicios basada en etiquetas docker El agente de CloudWatch consultará los metadatos de las tareas de ECS una vez por minuto y registrará los

destinos detectados en el archivo `/tmp/cwagent_ecs_auto_sd.yaml` dentro del contenedor del agente de CloudWatch.

El valor predeterminado de `sd_port_label` en la sección `docker_label` es `ECS_PROMETHEUS_EXPORTER_PORT`. Si algún contenedor en ejecución en las tareas de ECS tiene una etiqueta docker `ECS_PROMETHEUS_EXPORTER_PORT`, el agente CloudWatch utiliza el valor como `container port` para examinar todos los puertos expuestos del contenedor. Si hay una concordancia, el puerto del host mapeado más la IP privada del contenedor se utilizan para construir el destino del exportador de Prometheus en el siguiente formato: `private_ip:host_port`.

El valor predeterminado de `sd_metrics_path_label` en la sección `docker_label` es `ECS_PROMETHEUS_METRICS_PATH`. Si el contenedor tiene esta etiqueta docker, el valor se utilizará como la `__metrics_path__`. Si el contenedor no tiene esta etiqueta, se utiliza el valor predeterminado `/metrics`.

El valor predeterminado de `sd_job_name_label` en la sección `docker_label` es `job`. Si el contenedor tiene esta etiqueta docker, el valor se agregará como una de las etiquetas para que el destino reemplace el nombre de trabajo predeterminado que se especifica en la configuración de Prometheus. El valor de esta etiqueta docker se utiliza como nombre de flujo de registro en el grupo de registros de CloudWatch Logs.

Ejemplo 2

```
"ecs_service_discovery": {
  "sd_frequency": "15s",
  "sd_result_file": "/tmp/cwagent_ecs_auto_sd.yaml",
  "docker_label": {
    "sd_port_label": "ECS_PROMETHEUS_EXPORTER_PORT_SUBSET_A",
    "sd_job_name_label": "ECS_PROMETHEUS_JOB_NAME"
  }
}
```

En este ejemplo se habilita la detección de servicios basada en etiquetas docker. El agente de CloudWatch consultará los metadatos de las tareas de ECS cada 15 segundos y registrará los destinos detectados en el archivo `/tmp/cwagent_ecs_auto_sd.yaml` dentro del contenedor del agente de CloudWatch. Los contenedores con una etiqueta docker de `ECS_PROMETHEUS_EXPORTER_PORT_SUBSET_A` se examinarán. El valor de la etiqueta docker `ECS_PROMETHEUS_JOB_NAME` se utiliza como el nombre del trabajo.

Ejemplo 3

```
"ecs_service_discovery": {
  "sd_frequency": "5m",
  "sd_result_file": "/tmp/cwagent_ecs_auto_sd.yaml",
  "task_definition_list": [
    {
      "sd_job_name": "java-prometheus",
      "sd_metrics_path": "/metrics",
      "sd_metrics_ports": "9404; 9406",
      "sd_task_definition_arn_pattern": ".*:task-definition/.*javajmx.*:[0-9]+"
    },
    {
      "sd_job_name": "envoy-prometheus",
      "sd_metrics_path": "/stats/prometheus",
      "sd_container_name_pattern": "^envoy$",
      "sd_metrics_ports": "9901",
      "sd_task_definition_arn_pattern": ".*:task-definition/.*appmesh.*:23"
    }
  ]
}
```

Este ejemplo habilita la detección de servicios basada en expresiones ARN regulares de definición de tareas de ECS. El agente de CloudWatch consultará los metadatos de las tareas de ECS cada cinco minutos y registrará los destinos detectados en el archivo `/tmp/cwagent_ecs_auto_sd.yaml` dentro del contenedor del agente de CloudWatch.

Se definen dos secciones de expresión regular de ARN de definición de tarea:

- Para la primera sección, las tareas de ECS con `javajmx` en la definición de tarea de ECS de ARN se filtran para el análisis del puerto del contenedor. Si los contenedores dentro de estas tareas de ECS exponen el puerto del contenedor en 9404 o 9406, el puerto del host mapeado junto con la IP privada del contenedor se utilizan para crear los destinos del exportador de Prometheus. El valor de `sd_metrics_path` establece `__metrics_path__` a `/metrics`. Por lo tanto, el agente de CloudWatch raspará las métricas de Prometheus de `private_ip:host_port/metrics` y las métricas raspadas se enviarán al flujo de registro `java-prometheus` en CloudWatch Logs en el grupo de registros `/aws/ecs/containerinsights/cluster_name/prometheus`.
- Para la segunda sección, las tareas de ECS con `appmesh` en los ARN de definición de tareas de ECS y con `version` de `:23` se filtran para el análisis del puerto del contenedor. Para contenedores con un nombre de `envoy` que exponen el puerto del contenedor en `9901`, el puerto del host mapeado junto con la IP privada del contenedor se utilizan para crear los destinos del exportador de Prometheus. El valor dentro de estas tareas de ECS expone el puerto contenedor

en 9404 o 9406, el puerto del host mapeado junto con la IP privada del contenedor se utilizan para crear los destinos del exportador de Prometheus. El valor de `sd_metrics_path` establece `__metrics_path__` a `/stats/prometheus`. Por lo tanto, el agente de CloudWatch eliminará las métricas de Prometheus de `private_ip:host_port/stats/prometheus` y enviará las métricas raspadas al flujo de registros `envoy-prometheus` en CloudWatch Logs en el grupo de registros `/aws/ecs/containerinsights/cluster_name/prometheus`.

Ejemplo 4

```
"ecs_service_discovery": {
  "sd_frequency": "5m",
  "sd_result_file": "/tmp/cwagent_ecs_auto_sd.yaml",
  "service_name_list_for_tasks": [
    {
      "sd_job_name": "nginx-prometheus",
      "sd_metrics_path": "/metrics",
      "sd_metrics_ports": "9113",
      "sd_service_name_pattern": "^nginx-.*"
    },
    {
      "sd_job_name": "haproxy-prometheus",
      "sd_metrics_path": "/stats/metrics",
      "sd_container_name_pattern": "^haproxy$",
      "sd_metrics_ports": "8404",
      "sd_service_name_pattern": ".*haproxy-service.*"
    }
  ]
}
```

En este ejemplo se habilita la detección de servicios basada en expresiones regulares del nombre de servicio de ECS. El agente de CloudWatch consultará los metadatos de los servicios de ECS cada cinco minutos y registrará los destinos detectados en el archivo `/tmp/cwagent_ecs_auto_sd.yaml` dentro del contenedor del agente de CloudWatch.

Se definen dos secciones de expresiones regulares de nombre de servicio:

- Para la primera sección, las tareas de ECS asociadas con los servicios de ECS que tienen nombres que concuerdan con la expresión regular `^nginx-.*` se filtran para el análisis del puerto del contenedor. Si los contenedores dentro de estas tareas de ECS exponen el puerto del contenedor en 9113, el puerto del host mapeado junto con la IP privada del contenedor se utilizan

para crear los destinos del exportador de Prometheus. El valor de `sd_metrics_path` establece `__metrics_path__` a `/metrics`. Por lo tanto, el agente de CloudWatch raspará las métricas de Prometheus de `private_ip:host_port/metrics`, y las métricas raspadas se enviarán al flujo de registro `nginx-prometheus` en CloudWatch Logs en el grupo de registros `/aws/ecs/containerinsights/cluster_name/prometheus`.

- Para la segunda sección, las tareas de ECS asociadas con los servicios de ECS que tienen nombres que concuerdan con la expresión regular `.*haproxy-service.*` se filtran para el análisis del puerto del contenedor. En contenedores con un nombre de `haproxy` que exponen el puerto del contenedor en 8404, el puerto del host mapeado junto con la IP privada del contenedor se utilizan para crear los destinos del exportador de Prometheus. El valor de `sd_metrics_path` establece `__metrics_path__` a `/stats/metrics`. Por lo tanto, el agente de CloudWatch raspará las métricas de Prometheus de `private_ip:host_port/stats/metrics`, y las métricas de raspado se enviarán al flujo de registro `haproxy-prometheus` en CloudWatch Logs en el grupo de registros `/aws/ecs/containerinsights/cluster_name/prometheus`.

Ejemplo 5

```
"ecs_service_discovery": {
  "sd_frequency": "1m30s",
  "sd_result_file": "/tmp/cwagent_ecs_auto_sd.yaml",
  "docker_label": {
    "sd_port_label": "MY_PROMETHEUS_EXPORTER_PORT_LABEL",
    "sd_metrics_path_label": "MY_PROMETHEUS_METRICS_PATH_LABEL",
    "sd_job_name_label": "MY_PROMETHEUS_METRICS_NAME_LABEL"
  }
  "task_definition_list": [
    {
      "sd_metrics_ports": "9150",
      "sd_task_definition_arn_pattern": "*memcached.*"
    }
  ]
}
```

En este ejemplo se habilitan los dos modos de detección de servicios de ECS. El agente de CloudWatch consultará los metadatos de las tareas de ECS cada 90 segundos y registrará los destinos detectados en el archivo `/tmp/cwagent_ecs_auto_sd.yaml` dentro del contenedor del agente de CloudWatch.

Para la configuración de detección de servicios basada en docker:

- Las tareas de ECS con etiqueta docker `MY_PROMETHEUS_EXPORTER_PORT_LABEL` se filtrarán para el análisis del puerto de Prometheus. El puerto de destino del contenedor de Prometheus se especifica por el valor de la etiqueta `MY_PROMETHEUS_EXPORTER_PORT_LABEL`.
- El valor de la etiqueta docker `MY_PROMETHEUS_EXPORTER_PORT_LABEL` se utiliza para `__metrics_path__`. Si el contenedor no tiene esta etiqueta docker, se utiliza el valor predeterminado `/metrics`.
- El valor de la etiqueta docker `MY_PROMETHEUS_EXPORTER_PORT_LABEL` se utiliza como etiqueta de trabajo. Si el contenedor no tiene esta etiqueta docker, se utiliza el nombre del trabajo definido en la configuración de Prometheus.

Para la configuración de la detección de servicio basada en expresiones regulares ARN de definición de tareas de ECS:

- Las tareas de ECS con `memcached` en los ARN de definición de tareas de ECS se filtran para el análisis del puerto del contenedor. El puerto del contenedor del destino de Prometheus es 9150 de acuerdo a la definición de `sd_metrics_ports`. Se utiliza la ruta de métricas predeterminada `/metrics`. Se utiliza el nombre del trabajo definido en la configuración de Prometheus.

(Opcional) Configure de cargas de trabajo en contenedores de Amazon ECS de muestra para realizar pruebas con las métricas de Prometheus

Para probar la compatibilidad de las métricas de Prometheus en Información de contenedores de CloudWatch, se puede configurar una o varias de las siguientes cargas de trabajo en contenedores. El agente de CloudWatch compatible con Prometheus recopila automáticamente las métricas de cada una de estas cargas de trabajo. Para ver las métricas que se recopilan de forma predeterminada, consulte Métricas de Prometheus que el agente de CloudWatch recopila.

Temas

- Carga de trabajo de App Mesh de muestra para clústeres de Amazon ECS
- Carga de trabajo Java/JMX de muestra para clústeres de Amazon ECS
- Carga de trabajo NGINX de muestra para clústeres de Amazon ECS
- Carga de trabajo de muestra de NGINX Plus para clústeres de Amazon ECS
- Tutorial para añadir un nuevo destino de raspado de Prometheus: Memcached en Amazon ECS
- Tutorial para el raspado de métricas de Redis Prometheus en Amazon ECS Fargate

Carga de trabajo de App Mesh de muestra para clústeres de Amazon ECS

Para recopilar métricas de una carga de trabajo de Prometheus Amazon ECS de muestra para Amazon ECS, debe ejecutar Información de contenedores en el clúster. Para obtener más información sobre la instalación de Información de contenedores, consulte Configuración de Información de contenedores en Amazon ECS.

En primer lugar, siga esta walkthrough (explicación) para implementar la aplicación de color de muestra en el clúster de Amazon ECS. Una vez finalizado, tendrá las métricas de Prometheus de App Mesh expuestas en el puerto 9901.

A continuación, siga estos pasos para instalar el agente de CloudWatch con supervisión de Prometheus en el mismo clúster de Amazon ECS en el que instaló la aplicación de color. Los pasos descritos en esta sección instalan el agente de CloudWatch en modo de redes puente.

Las variables de entorno `ENVIRONMENT_NAME`, `AWS_PROFILE` y `AWS_DEFAULT_REGION` que establezca en la explicación también se utilizarán en los siguientes pasos.

Para instalar el agente de CloudWatch con supervisión de Prometheus para las pruebas

1. Descargue la plantilla de AWS CloudFormation con el siguiente comando.

```
curl -O https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
insights/latest/ecs-task-definition-templates/deployment-mode/replica-service/
cwagent-prometheus/cloudformation-quickstart/cwagent-ecs-prometheus-metric-for-
bridge-host.yaml
```

2. Configure el modo de red con los siguientes comandos.

```
export ECS_CLUSTER_NAME=${ENVIRONMENT_NAME}
export ECS_NETWORK_MODE=bridge
```

3. Cree la pila de AWS CloudFormation con los siguientes comandos.

```
aws cloudformation create-stack --stack-name CWAgent-Prometheus-ECS-
${ECS_CLUSTER_NAME}-EC2-${ECS_NETWORK_MODE} \
    --template-body file://cwagent-ecs-prometheus-metric-for-bridge-host.yaml \
    --parameters ParameterKey=ECSClusterName,ParameterValue=${ECS_CLUSTER_NAME} \
                 ParameterKey=CreateIAMRoles,ParameterValue=True \
                 ParameterKey=ECSNetworkMode,ParameterValue=${ECS_NETWORK_MODE} \
                 ParameterKey=TaskRoleName,ParameterValue=CWAgent-Prometheus-
TaskRole-${ECS_CLUSTER_NAME} \
```

```
                  ParameterKey=ExecutionRoleName,ParameterValue=CWAgent-Prometheus-
ExecutionRole-${ECS_CLUSTER_NAME} \
    --capabilities CAPABILITY_NAMED_IAM \
    --region ${AWS_DEFAULT_REGION} \
    --profile ${AWS_PROFILE}
```

4. (Opcional) Cuando se crea la pila de AWS CloudFormation, se observa un mensaje de `CREATE_COMPLETE`. Si desea verificar el estado antes de ver el mensaje, ingrese el siguiente comando.

```
aws cloudformation describe-stacks \
--stack-name CWAgent-Prometheus-ECS-${ECS_CLUSTER_NAME}-EC2-${ECS_NETWORK_MODE} \
--query 'Stacks[0].StackStatus' \
--region ${AWS_DEFAULT_REGION} \
--profile ${AWS_PROFILE}
```

Solución de problemas

En los pasos de la explicación se utiliza jq para analizar el resultado de salida deAWS CLI. Para obtener más información sobre la instalación de jq, consulte jq. Utilice el siguiente comando para establecer el formato de salida predeterminado de AWS CLI a formato JSON para que jq pueda analizarlo de forma correcta.

```
$ aws configure
```

Cuando la respuesta llegue a `Default output format`, ingrese **`json`**.

Desinstale el agente de CloudWatch con supervisión de Prometheus

Cuando termine de realizar la prueba, ingrese el siguiente comando para desinstalar el agente de CloudWatch mediante la eliminación de la pila de AWS CloudFormation.

```
aws cloudformation delete-stack \
--stack-name CWAgent-Prometheus-ECS-${ECS_CLUSTER_NAME}-EC2-${ECS_NETWORK_MODE} \
--region ${AWS_DEFAULT_REGION} \
--profile ${AWS_PROFILE}
```

Carga de trabajo Java/JMX de muestra para clústeres de Amazon ECS

JMX Exporter es un exportador oficial de Prometheus que puede extraer y exponer mBeans de JMX como métricas de Prometheus. Para obtener más información, consulte prometheus/jmx_exporter.

El agente de CloudWatch compatible con Prometheus raspa las métricas de Java/JMX Prometheus en función de la configuración de detección de servicios en el clúster de Amazon ECS. Puede configurar JMX Exporter para exponer las métricas en un puerto o ruta (metrics_path) diferente. Si cambia el puerto o la ruta, actualice la sección `ecs_service_discovery` predeterminada en la configuración del agente de CloudWatch.

Para recopilar métricas de una carga de trabajo de Prometheus de muestra para Amazon ECS, debe ejecutar Información de contenedores en el clúster. Para obtener más información sobre la instalación de Información de contenedores, consulte Configuración de Información de contenedores en Amazon ECS.

Para instalar la carga de trabajo de ejemplo de Java/JMX para clústeres de Amazon ECS

1. Siga los pasos descritos en estas secciones para crear las imágenes de Docker.
   - Ejemplo: Imagen de Docker de una aplicación Jar de Java con métricas de Prometheus
   - Ejemplo: Imagen de Apache Tomcat Docker con métricas de Prometheus
2. Especifique las dos etiquetas docker siguientes en el archivo de definición de tareas de Amazon ECS. A continuación, se puede ejecutar la definición de tarea como un Servicio ECS de Amazon o como una tarea de Amazon ECS en el clúster.
   - Establezca `ECS_PROMETHEUS_EXPORTER_PORT` para apuntar al ContainerPort donde están expuestas las métricas de Prometheus.
   - Establezca `Java_EMF_Metrics` en `true`. El agente de CloudWatch utiliza este indicador para generar el formato de métrica integrada en el evento de registro.

   A continuación, se muestra un ejemplo:

```
{
  "family": "workload-java-ec2-bridge",
  "taskRoleArn": "{{task-role-arn}}",
  "executionRoleArn": "{{execution-role-arn}}",
  "networkMode": "bridge",
  "containerDefinitions": [
```

```
    {
      "name": "tomcat-prometheus-workload-java-ec2-bridge-dynamic-port",
      "image": "your_docker_image_tag_for_tomcat_with_prometheus_metrics",
      "portMappings": [
        {
          "hostPort": 0,
          "protocol": "tcp",
          "containerPort": 9404
        }
      ],
      "dockerLabels": {
        "ECS_PROMETHEUS_EXPORTER_PORT": "9404",
        "Java_EMF_Metrics": "true"
      }
    }
  ],
  "requiresCompatibilities": [
    "EC2"  ],
  "cpu": "256",
  "memory": "512"
  }
```

La configuración predeterminada del agente de CloudWatch en la plantilla de AWS CloudFormation permite tanto la detección de servicios basados en etiquetas docker como la detección de servicios basada en ARN de definición de tareas. Para ver estas configuraciones predeterminadas, consulte la línea 65 del CloudWatch agent YAML configuration file (Archivo de configuración YAML del agente de CloudWatch). Los contenedores con etiquetas `ECS_PROMETHEUS_EXPORTER_PORT` se detectarán automáticamente en función del puerto de contenedor especificado para el raspado de Prometheus.

La configuración predeterminada del agente de CloudWatch también tiene la configuración `metric_declaration` para Java/JMX en la línea 112 del mismo archivo. Todas las etiquetas docker de los contenedores de destino se agregarán como etiquetas adicionales en las métricas de Prometheus y se enviarán a CloudWatch Logs. Para los contenedores Java/JMX con etiqueta docker `Java_EMF_Metrics="true"`, se generará el formato de métrica integrada.

Carga de trabajo NGINX de muestra para clústeres de Amazon ECS

El exportador NGINX de Prometheus puede raspar y exponer datos NGINX como métricas de Prometheus. En este ejemplo se utiliza el exportador junto con el servicio proxy inverso NGINX para Amazon ECS.

Para obtener más información sobre el exportador NGINX de Prometheus, consulte nginx-prometheus-exporter en Github. Para obtener más información sobre el proxy inverso NGINX, consulte ecs-nginx-reverse-proxy en Github.

El agente de CloudWatch compatible con Prometheus raspa las métricas de NGINX de Prometheus basadas en la configuración de detección de servicios en el clúster de Amazon ECS. Puede configurar el exportador NGINX de Prometheus para exponer las métricas en un puerto o una ruta diferente. Si cambia el puerto o la ruta, actualice la sección `ecs_service_discovery` en el archivo de configuración del agente de CloudWatch.

Instale la carga de trabajo de muestra del proxy inverso NGINX para clústeres de Amazon ECS

Siga los pasos para instalar la carga de trabajo de muestra del proxy inverso NGINX.

Cree las imágenes de Docker

Para crear las imágenes de Docker para la carga de trabajo de muestra del proxy inverso NGINX

1. Descargue la siguiente carpeta del repositorio proxy inverso NGINX: https://github.com/awslabs/ecs-nginx-reverse-proxy/tree/master/reverse-proxy/.
2. Busque el directorio de la `app` y cree una imagen desde ese directorio:

   ```
   docker build -t web-server-app ./path-to-app-directory
   ```
3. Cree una imagen personalizada para NGINX. Primero, cree un directorio con los dos siguientes archivos:

   - Un archivo Dockerfile de muestra:

     ```
     FROM nginx
     COPY nginx.conf /etc/nginx/nginx.conf
     ```
   - Un archivo `nginx.conf`, modificado desde https://github.com/awslabs/ecs-nginx-reverse-proxy/tree/master/reverse-proxy/:

     ```
     events {
       worker_connections 768;
     }

     http {
       # Nginx will handle gzip compression of responses from the app server
     ```

```
  gzip on;
  gzip_proxied any;
  gzip_types text/plain application/json;
  gzip_min_length 1000;

  server{
    listen 8080;
    location /stub_status {
        stub_status   on;
    }
  }

  server {
    listen 80;

    # Nginx will reject anything not matching /api
    location /api {
      # Reject requests with unsupported HTTP method
      if ($request_method !~ ^(GET|POST|HEAD|OPTIONS|PUT|DELETE)$) {
        return 405;
      }

      # Only requests matching the whitelist expectations will
      # get sent to the application server
      proxy_pass http://app:3000;
      proxy_http_version 1.1;
      proxy_set_header Upgrade $http_upgrade;
      proxy_set_header Connection 'upgrade';
      proxy_set_header Host $host;
      proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
      proxy_cache_bypass $http_upgrade;
    }
  }
}
```

> **Note**
>
> `stub_status` debe estar habilitado en el mismo puerto que `nginx-prometheus-exporter` está configurado para raspar métricas desde allí. En nuestra definición de tarea de ejemplo, `nginx-prometheus-exporter` está configurado para raspar métricas del puerto 8080.

4. Cree una imagen a partir de archivos en el directorio nuevo:

```
docker build -t nginx-reverse-proxy ./path-to-your-directory
```

5. Cargue las imágenes nuevas en un repositorio de imágenes para usarlas posteriormente.

Cree la definición de tarea para ejecutar NGINX y la aplicación de servidor web en Amazon ECS

A continuación, se debe configurar la definición de tarea.

Esta definición de tarea permite la recopilación y exportación de métricas de NGINX de Prometheus. El contenedor NGINX realiza un seguimiento de la entrada desde la aplicación y expone esos datos al puerto 8080, como se establece en `nginx.conf`. El contenedor exportador NGINX de Prometheus raspa estas métricas y las publica en el puerto 9113, para usarlas en CloudWatch.

Para configurar la definición de tarea para la carga de trabajo de muestra de NGINX de Amazon ECS

1. Cree un archivo de definición de tarea JSON con el siguiente contenido. Reemplace *your-customized-nginx-iamge* con el URI de imagen para la imagen NGINX personalizada y reemplace *your-web-server-app-image* con el URI de imagen para la imagen de la aplicación del servidor web.

```
{
  "containerDefinitions": [
    {
      "name": "nginx",
      "image": "your-customized-nginx-image",
      "memory": 256,
      "cpu": 256,
      "essential": true,
      "portMappings": [
        {
          "containerPort": 80,
          "protocol": "tcp"
        }
      ],
      "links": [
        "app"
      ]
    },
    {
      "name": "app",
```

```
      "image": "your-web-server-app-image",
      "memory": 256,
      "cpu": 256,
      "essential": true
    },
    {
      "name": "nginx-prometheus-exporter",
      "image": "docker.io/nginx/nginx-prometheus-exporter:0.8.0",
      "memory": 256,
      "cpu": 256,
      "essential": true,
      "command": [
        "-nginx.scrape-uri",
        "http://nginx:8080/stub_status"
    ],
    "links":[
      "nginx"
    ],
      "portMappings":[
        {
          "containerPort": 9113,
          "protocol": "tcp"
        }
      ]
    }
  ],
  "networkMode": "bridge",
  "placementConstraints": [],
  "family": "nginx-sample-stack"
}
```

2. Ingrese el siguiente comando para registrar la definición de tarea.

```
aws ecs register-task-definition --cli-input-json file://path-to-your-task-definition-json
```

3. Cree un servicio para ejecutar la tarea mediante el siguiente comando:

   Asegúrese de no cambiar el nombre del servicio. Se ejecutará un servicio del agente de CloudWatch con una configuración que busca tareas mediante los patrones de nombres de los servicios que las iniciaron. Por ejemplo, para que el agente de CloudWatch encuentre la tarea que este comando ha lanzado, puede especificar que el valor de `sd_service_name_pattern` sea `^nginx-service$`. En la siguiente sección se proporcionan más detalles.

```
aws ecs create-service \
 --cluster your-cluster-name \
 --service-name nginx-service \
 --task-definition nginx-sample-stack:1 \
 --desired-count 1
```

Configure el agente de CloudWatch para que realice el raspado de las métricas de NGINX Prometheus

El paso final es configurar el agente de CloudWatch para que realice el raspado de las métricas de NGINX. En este ejemplo, el agente de CloudWatch detecta la tarea mediante el patrón de nombre de servicio y el puerto 9113, donde el exportador expone las métricas de prometheus para NGINX. Con la tarea detectada y las métricas disponibles, el agente de CloudWatch comienza a publicar las métricas recopiladas en el flujo de registro nginx-prometheus-exporter.

Para configurar el agente de CloudWatch para que realice el raspado de las métricas de NGINX

1. Descargue la última versión del archivo YAML necesario con el siguiente comando.

```
curl -O https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
insights/latest/ecs-task-definition-templates/deployment-mode/replica-service/
cwagent-prometheus/cloudformation-quickstart/cwagent-ecs-prometheus-metric-for-
bridge-host.yaml
```

2. Abra el archivo con un editor de texto y busque la configuración completa del agente CloudWatch en la clave de `value` en la sección `resource:CWAgentConfigSSMParameter`. A continuación, en la sección `ecs_service_discovery`, agregue la siguiente sección `service_name_list_for_tasks`.

```
"service_name_list_for_tasks": [
  {
    "sd_job_name": "nginx-prometheus-exporter",
    "sd_metrics_path": "/metrics",
    "sd_metrics_ports": "9113",
    "sd_service_name_pattern": "^nginx-service$"
   }
],
```

3. En el mismo archivo, agregue la siguiente sección en la sección `metric_declaration` para permitir las métricas de NGINX. Asegúrese de seguir el patrón de sangría existente.

```
{
  "source_labels": ["job"],
  "label_matcher": ".*nginx.*",
  "dimensions": [["ClusterName", "TaskDefinitionFamily", "ServiceName"]],
  "metric_selectors": [
    "^nginx_.*$"
  ]
},
```

4. Si aún no tiene el agente de CloudWatch implementado en este clúster, diríjase directamente al paso 8.

   Si ya implementó el agente de CloudWatch en el clúster de Amazon ECS mediante AWS CloudFormation, puede crear un conjunto de cambios con los siguientes comandos:

```
ECS_CLUSTER_NAME=your_cluster_name
AWS_REGION=your_aws_region
ECS_NETWORK_MODE=bridge
CREATE_IAM_ROLES=True
ECS_TASK_ROLE_NAME=your_selected_ecs_task_role_name
ECS_EXECUTION_ROLE_NAME=your_selected_ecs_execution_role_name

aws cloudformation create-change-set --stack-name CWAgent-Prometheus-ECS-
${ECS_CLUSTER_NAME}-EC2-${ECS_NETWORK_MODE} \
    --template-body file://cwagent-ecs-prometheus-metric-for-bridge-host.yaml \
    --parameters ParameterKey=ECSClusterName,ParameterValue=$ECS_CLUSTER_NAME \
                 ParameterKey=CreateIAMRoles,ParameterValue=$CREATE_IAM_ROLES \
                 ParameterKey=ECSNetworkMode,ParameterValue=$ECS_NETWORK_MODE \
                 ParameterKey=TaskRoleName,ParameterValue=$ECS_TASK_ROLE_NAME \
                 ParameterKey=ExecutionRoleName,ParameterValue=
$ECS_EXECUTION_ROLE_NAME \
    --capabilities CAPABILITY_NAMED_IAM \
    --region $AWS_REGION \
    --change-set-name nginx-scraping-support
```

5. Abra la consola de AWS CloudFormation en https://console.aws.amazon.com/cloudformation.
6. Revise el conjunto de cambios recién creado nginx-scraping-support. Se debería ver un cambio aplicado al recurso CWAgentConfigSSMParameter. Ejecute el conjunto de cambios y reinicie la tarea del agente de CloudWatch con el siguiente comando:

```
aws ecs update-service --cluster $ECS_CLUSTER_NAME \
--desired-count 0 \
--service cwagent-prometheus-replica-service-EC2-$ECS_NETWORK_MODE \
--region $AWS_REGION
```

7. Espere aproximadamente 10 segundos y, a continuación, ingrese el siguiente comando.

```
aws ecs update-service --cluster $ECS_CLUSTER_NAME \
--desired-count 1 \
--service cwagent-prometheus-replica-service-EC2-$ECS_NETWORK_MODE \
--region $AWS_REGION
```

8. Si va a instalar el agente de CloudWatch con la recopilación de métricas de Prometheus en el clúster por primera vez, ingrese los siguientes comandos.

```
ECS_CLUSTER_NAME=your_cluster_name
AWS_REGION=your_aws_region
ECS_NETWORK_MODE=bridge
CREATE_IAM_ROLES=True
ECS_TASK_ROLE_NAME=your_selected_ecs_task_role_name
ECS_EXECUTION_ROLE_NAME=your_selected_ecs_execution_role_name

aws cloudformation create-stack --stack-name CWAgent-Prometheus-ECS-
${ECS_CLUSTER_NAME}-EC2-${ECS_NETWORK_MODE} \
    --template-body file://cwagent-ecs-prometheus-metric-for-bridge-host.yaml \
    --parameters ParameterKey=ECSClusterName,ParameterValue=$ECS_CLUSTER_NAME \
                 ParameterKey=CreateIAMRoles,ParameterValue=$CREATE_IAM_ROLES \
                 ParameterKey=ECSNetworkMode,ParameterValue=$ECS_NETWORK_MODE \
                 ParameterKey=TaskRoleName,ParameterValue=$ECS_TASK_ROLE_NAME \
                 ParameterKey=ExecutionRoleName,ParameterValue=
$ECS_EXECUTION_ROLE_NAME \
    --capabilities CAPABILITY_NAMED_IAM \
    --region $AWS_REGION
```

Visualización de las métricas y los registros de NGINX

Ahora puede ver las métricas de NGINX que se están recopilando.

Para visualizar las métricas de la carga de trabajo de muestra de NGINX

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.

2. En la Región en la que se está ejecutando el clúster, elija Metrics (Métricas) en el panel de navegación izquierdo. Busque el espacio de nombres ContainerInsights/Prometheus para ver las métricas.
3. Para ver los eventos de CloudWatch Logs, elija Log groups (Grupos de registros) en el panel de navegación. Los eventos se encuentran en el grupo de registros /aws/containerinsights/***your_cluster_name***/prometheus, en el flujo de registros *nginx-prometheus-exporter*.

Carga de trabajo de muestra de NGINX Plus para clústeres de Amazon ECS

NGINX Plus es la versión comercial de NGINX. Debe contar con una licencia para utilizarla. Para obtener más información, consulte NGINX Plus.

El exportador NGINX de Prometheus puede raspar y exponer datos de NGINX como las métricas de Prometheus. En este ejemplo se utiliza el exportador junto con el servicio proxy inverso NGINX Plus para Amazon ECS.

Para obtener más información sobre el exportador NGINX de Prometheus, consulte nginx-prometheus-exporter en Github. Para obtener más información sobre el proxy inverso NGINX, consulte ecs-nginx-reverse-proxy en Github.

El agente de CloudWatch compatible con Prometheus realiza el raspado de las métricas de NGINX Plus Prometheus basado en la configuración de detección de servicios en el clúster de Amazon ECS. Puede configurar el exportador NGINX de Prometheus para exponer las métricas en un puerto o una ruta diferente. Si cambia el puerto o la ruta, actualice la sección de `ecs_service_discovery` en el archivo de configuración del agente de CloudWatch.

Instale la carga de trabajo de muestra del proxy inverso NGINX Plus para clústeres de Amazon ECS

Siga los pasos para instalar la carga de trabajo de muestra del proxy inverso NGINX.

Cree las imágenes de Docker

Para crear las imágenes de Docker para la carga de trabajo de muestra del proxy inverso NGINX Plus

1. Descargue la siguiente carpeta del repositorio del proxy inverso NGINX: https://github.com/awslabs/ecs-nginx-reverse-proxy/tree/master/reverse-proxy/.
2. Busque el directorio de la `app` y cree una imagen desde ese directorio:

```
docker build -t web-server-app ./path-to-app-directory
```

3. Cree una imagen personalizada para NGINX Plus. Antes de poder crear la imagen para NGINX Plus, debe obtener la clave llamada `nginx-repo.key` y el certificado SSL `nginx-repo.crt` para la licencia de NGINX Plus. Cree un directorio y almacene en él la `nginx-repo.key` y los archivos `nginx-repo.crt`.

   En el directorio que acaba de crear, cree los siguientes dos archivos:

   - Una muestra de Dockerfile con el siguiente contenido. Este archivo docker se adopta a partir de un archivo de muestra que se proporciona en https://docs.nginx.com/nginx/admin-guide/installing-nginx/installing-nginx-docker/#docker_plus_image. El cambio importante que se realiza es que se carga un archivo por separado, llamado`nginx.conf`, que se creará en el siguiente paso.

     ```
     FROM debian:buster-slim

     LABEL maintainer="NGINX Docker Maintainers <docker-maint@nginx.com>“

     # Define NGINX versions for NGINX Plus and NGINX Plus modules
     # Uncomment this block and the versioned nginxPackages block in the main RUN
     # instruction to install a specific release
     # ENV NGINX_VERSION 21
     # ENV NJS_VERSION 0.3.9
     # ENV PKG_RELEASE 1~buster

     # Download certificate and key from the customer portal (https://cs.nginx.com
      (https://cs.nginx.com/))
     # and copy to the build context
     COPY nginx-repo.crt /etc/ssl/nginx/
     COPY nginx-repo.key /etc/ssl/nginx/
     # COPY nginx.conf /etc/ssl/nginx/nginx.conf

     RUN set -x \
     # Create nginx user/group first, to be consistent throughout Docker variants
     && addgroup --system --gid 101 nginx \
     && adduser --system --disabled-login --ingroup nginx --no-create-home --home /
     nonexistent --gecos "nginx user" --shell /bin/false --uid 101 nginx \
     && apt-get update \
     && apt-get install --no-install-recommends --no-install-suggests -y ca-
     certificates gnupg1 \
     ```

```
&& \
NGINX_GPGKEY=573BFD6B3D8FBC641079A6ABABF5BD827BD9BF62; \
found=''; \
for server in \
ha.pool.sks-keyservers.net (http://ha.pool.sks-keyservers.net/) \
hkp://keyserver.ubuntu.com:80 \
hkp://p80.pool.sks-keyservers.net:80 \
pgp.mit.edu (http://pgp.mit.edu/) \
; do \
echo "Fetching GPG key $NGINX_GPGKEY from $server"; \
apt-key adv --keyserver "$server" --keyserver-options timeout=10 --recv-keys
 "$NGINX_GPGKEY" && found=yes && break; \
done; \
test -z "$found" && echo >&2 "error: failed to fetch GPG key $NGINX_GPGKEY" &&
 exit 1; \
apt-get remove --purge --auto-remove -y gnupg1 && rm -rf /var/lib/apt/lists/* \
# Install the latest release of NGINX Plus and/or NGINX Plus modules
# Uncomment individual modules if necessary
# Use versioned packages over defaults to specify a release
&& nginxPackages=" \
nginx-plus \
# nginx-plus=${NGINX_VERSION}-${PKG_RELEASE} \
# nginx-plus-module-xslt \
# nginx-plus-module-xslt=${NGINX_VERSION}-${PKG_RELEASE} \
# nginx-plus-module-geoip \
# nginx-plus-module-geoip=${NGINX_VERSION}-${PKG_RELEASE} \
# nginx-plus-module-image-filter \
# nginx-plus-module-image-filter=${NGINX_VERSION}-${PKG_RELEASE} \
# nginx-plus-module-perl \
# nginx-plus-module-perl=${NGINX_VERSION}-${PKG_RELEASE} \
# nginx-plus-module-njs \
# nginx-plus-module-njs=${NGINX_VERSION}+${NJS_VERSION}-${PKG_RELEASE} \
" \
&& echo "Acquire::https::plus-pkgs.nginx.com::Verify-Peer \"true\";" >> /etc/apt/
apt.conf.d/90nginx \
&& echo "Acquire::https::plus-pkgs.nginx.com::Verify-Host \"true\";" >> /etc/apt/
apt.conf.d/90nginx \
&& echo "Acquire::https::plus-pkgs.nginx.com::SslCert \"/etc/ssl/nginx/nginx-
repo.crt\";" >> /etc/apt/apt.conf.d/90nginx \
&& echo "Acquire::https::plus-pkgs.nginx.com::SslKey \"/etc/ssl/nginx/nginx-
repo.key\";" >> /etc/apt/apt.conf.d/90nginx \
&& printf "deb https://plus-pkgs.nginx.com/debian buster nginx-plus\n" > /etc/
apt/sources.list.d/nginx-plus.list \
&& apt-get update \
```

```
&& apt-get install --no-install-recommends --no-install-suggests -y \
$nginxPackages \
gettext-base \
curl \
&& apt-get remove --purge --auto-remove -y && rm -rf /var/lib/apt/lists/* /etc/
apt/sources.list.d/nginx-plus.list \
&& rm -rf /etc/apt/apt.conf.d/90nginx /etc/ssl/nginx

# Forward request logs to Docker log collector
RUN ln -sf /dev/stdout /var/log/nginx/access.log \
&& ln -sf /dev/stderr /var/log/nginx/error.log

COPY nginx.conf /etc/nginx/nginx.conf

EXPOSE 80

STOPSIGNAL SIGTERM

CMD ["nginx", "-g", "daemon off;"]
```

- Un archivo `nginx.conf`, modificado desde https://github.com/awslabs/ecs-nginx-reverse-proxy/tree/master/reverse-proxy/nginx.

```
events {
  worker_connections 768;
}

http {
  # Nginx will handle gzip compression of responses from the app server
  gzip on;
  gzip_proxied any;
  gzip_types text/plain application/json;
  gzip_min_length 1000;

  upstream backend {
    zone name 10m;
    server app:3000    weight=2;
    server app2:3000    weight=1;
  }

  server{
    listen 8080;
    location /api {
```

```
      api write=on;
    }
  }

  match server_ok {
    status 100-599;
  }

  server {
    listen 80;
    status_zone zone;
    # Nginx will reject anything not matching /api
    location /api {
      # Reject requests with unsupported HTTP method
      if ($request_method !~ ^(GET|POST|HEAD|OPTIONS|PUT|DELETE)$) {
        return 405;
      }

      # Only requests matching the whitelist expectations will
      # get sent to the application server
      proxy_pass http://backend;
      health_check uri=/lorem-ipsum match=server_ok;
      proxy_http_version 1.1;
      proxy_set_header Upgrade $http_upgrade;
      proxy_set_header Connection 'upgrade';
      proxy_set_header Host $host;
      proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
      proxy_cache_bypass $http_upgrade;
    }
  }
}
```

4. Cree una imagen a partir de los archivos en el directorio nuevo:

```
docker build -t nginx-plus-reverse-proxy ./path-to-your-directory
```

5. Cargue las imágenes nuevas en un repositorio de imágenes para usarlas posteriormente.

Cree la definición de tarea para ejecutar NGINX Plus y la aplicación de servidor web en Amazon ECS

A continuación, se debe configurar la definición de tarea.

Esta definición de tarea permite la recopilación y exportación de métricas de NGINX Plus Prometheus. El contenedor NGINX realiza un seguimiento de la entrada desde la aplicación y expone esos datos al puerto 8080, como se establece en `nginx.conf`. El contenedor exportador NGINX de Prometheus raspa estas métricas y las publica en el puerto 9113, para usarlas en CloudWatch.

Para configurar la definición de tarea para la carga de trabajo de muestra de NGINX de Amazon ECS

1. Cree un archivo JSON de definición de tarea con el siguiente contenido. Reemplace *your-customized-nginx-plus-image* por el URI de imagen para la imagen personalizada de NGINX Plus, y reemplace *your-web-server-app-image* por el URI de imagen para la imagen de la aplicación del servidor web.

```
{
  "containerDefinitions": [
    {
      "name": "nginx",
      "image": "your-customized-nginx-plus-image",
      "memory": 256,
      "cpu": 256,
      "essential": true,
      "portMappings": [
        {
          "containerPort": 80,
          "protocol": "tcp"
        }
      ],
      "links": [
        "app",
        "app2"
      ]
    },
    {
      "name": "app",
      "image": "your-web-server-app-image",
      "memory": 256,
      "cpu": 128,
      "essential": true
    },
    {
      "name": "app2",
      "image": "your-web-server-app-image",
```

```
      "memory": 256,
      "cpu": 128,
      "essential": true
    },
    {
      "name": "nginx-prometheus-exporter",
      "image": "docker.io/nginx/nginx-prometheus-exporter:0.8.0",
      "memory": 256,
      "cpu": 256,
      "essential": true,
      "command": [
        "-nginx.plus",
        "-nginx.scrape-uri",
         "http://nginx:8080/api"
    ],
    "links":[
      "nginx"
    ],
      "portMappings":[
        {
          "containerPort": 9113,
          "protocol": "tcp"
        }
      ]
    }
  ],
  "networkMode": "bridge",
  "placementConstraints": [],
  "family": "nginx-plus-sample-stack"
}
```

2. Registre la definición de tarea:

```
aws ecs register-task-definition --cli-input-json file://path-to-your-task-definition-json
```

3. Cree un servicio para ejecutar la tarea con el siguiente comando:

```
aws ecs create-service \
 --cluster your-cluster-name \
 --service-name nginx-plus-service \
 --task-definition nginx-plus-sample-stack:1 \
 --desired-count 1
```

Asegúrese de no cambiar el nombre del servicio. Se ejecutará un servicio del agente de CloudWatch con una configuración que busca tareas mediante los patrones de nombres de los servicios que las iniciaron. Por ejemplo, para que el agente de CloudWatch encuentre la tarea que este comando ha lanzado, puede especificar que el valor de `sd_service_name_pattern` sea `^nginx-plus-service$`. En la siguiente sección se proporcionan más detalles.

Configure el agente de CloudWatch para que raspe las métricas de NGINX Plus de Prometheus

El paso final es configurar el agente de CloudWatch para que realice el raspado de las métricas de NGINX. En este ejemplo, el agente de CloudWatch detecta la tarea mediante el patrón de nombre de servicio y el puerto 9113, donde el exportador expone las métricas de prometheus para NGINX. Con la tarea detectada y las métricas disponibles, el agente de CloudWatch comienza a publicar las métricas recopiladas en el flujo de registro nginx-prometheus-exporter.

Para configurar el agente de CloudWatch para que realice el raspado de las métricas de NGINX

1. Descargue la última versión del archivo YAML necesario con el siguiente comando.

```
curl -O https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-insights/latest/ecs-task-definition-templates/deployment-mode/replica-service/cwagent-prometheus/cloudformation-quickstart/cwagent-ecs-prometheus-metric-for-bridge-host.yaml
```

2. Abra el archivo con un editor de texto y busque la configuración completa del agente CloudWatch en la clave de `value` en la sección `resource:CWAgentConfigSSMParameter`. A continuación, en la sección `ecs_service_discovery`, agregue la siguiente sección `service_name_list_for_tasks`.

```
"service_name_list_for_tasks": [
  {
    "sd_job_name": "nginx-plus-prometheus-exporter",
    "sd_metrics_path": "/metrics",
    "sd_metrics_ports": "9113",
    "sd_service_name_pattern": "^nginx-plus.*"
   }
],
```

3. En el mismo archivo, agregue la siguiente sección en la sección `metric_declaration` para permitir las métricas de NGINX Plus. Asegúrese de seguir el patrón de sangría existente.

```
{
  "source_labels": ["job"],
  "label_matcher": "^nginx-plus.*",
  "dimensions": [["ClusterName", "TaskDefinitionFamily", "ServiceName"]],
  "metric_selectors": [
    "^nginxplus_connections_accepted$",
    "^nginxplus_connections_active$",
    "^nginxplus_connections_dropped$",
    "^nginxplus_connections_idle$",
    "^nginxplus_http_requests_total$",
    "^nginxplus_ssl_handshakes$",
    "^nginxplus_ssl_handshakes_failed$",
    "^nginxplus_up$",
    "^nginxplus_upstream_server_health_checks_fails$"
  ]
},
{
  "source_labels": ["job"],
  "label_matcher": "^nginx-plus.*",
  "dimensions": [["ClusterName", "TaskDefinitionFamily", "ServiceName",
 "upstream"]],
  "metric_selectors": [
    "^nginxplus_upstream_server_response_time$"
  ]
},
{
  "source_labels": ["job"],
  "label_matcher": "^nginx-plus.*",
  "dimensions": [["ClusterName", "TaskDefinitionFamily", "ServiceName", "code"]],
  "metric_selectors": [
    "^nginxplus_upstream_server_responses$",
    "^nginxplus_server_zone_responses$"
  ]
},
```

4. Si aún no tiene el agente de CloudWatch implementado en este clúster, diríjase directamente al paso 8.

   Si ya implementó el agente de CloudWatch en el clúster de Amazon ECS mediante AWS CloudFormation, puede crear un conjunto de cambios con los siguientes comandos:

```
ECS_CLUSTER_NAME=your_cluster_name
```

```
AWS_REGION=your_aws_region
ECS_NETWORK_MODE=bridge
CREATE_IAM_ROLES=True
ECS_TASK_ROLE_NAME=your_selected_ecs_task_role_name
ECS_EXECUTION_ROLE_NAME=your_selected_ecs_execution_role_name

aws cloudformation create-change-set --stack-name CWAgent-Prometheus-ECS-
${ECS_CLUSTER_NAME}-EC2-${ECS_NETWORK_MODE} \
    --template-body file://cwagent-ecs-prometheus-metric-for-bridge-host.yaml \
    --parameters ParameterKey=ECSClusterName,ParameterValue=$ECS_CLUSTER_NAME \
                 ParameterKey=CreateIAMRoles,ParameterValue=$CREATE_IAM_ROLES \
                 ParameterKey=ECSNetworkMode,ParameterValue=$ECS_NETWORK_MODE \
                 ParameterKey=TaskRoleName,ParameterValue=$ECS_TASK_ROLE_NAME \
                 ParameterKey=ExecutionRoleName,ParameterValue=
$ECS_EXECUTION_ROLE_NAME \
    --capabilities CAPABILITY_NAMED_IAM \
    --region $AWS_REGION \
    --change-set-name nginx-plus-scraping-support
```

5. Abra la consola de AWS CloudFormation en https://console.aws.amazon.com/cloudformation.
6. Revise el conjunto de cambios recién creado nginx-plus-scraping-support. Se debería ver un cambio aplicado al recurso CWAgentConfigSSMParamete. Ejecute el conjunto de cambios y reinicie la tarea del agente de CloudWatch al ingresar el siguiente comando:

```
aws ecs update-service --cluster $ECS_CLUSTER_NAME \
--desired-count 0 \
--service cwagent-prometheus-replica-service-EC2-$ECS_NETWORK_MODE \
--region $AWS_REGION
```

7. Espere aproximadamente 10 segundos y, a continuación, ingrese el siguiente comando.

```
aws ecs update-service --cluster $ECS_CLUSTER_NAME \
--desired-count 1 \
--service cwagent-prometheus-replica-service-EC2-$ECS_NETWORK_MODE \
--region $AWS_REGION
```

8. Si va a instalar el agente de CloudWatch con la recopilación de métricas de Prometheus en el clúster por primera vez, ingrese los siguientes comandos.

```
ECS_CLUSTER_NAME=your_cluster_name
AWS_REGION=your_aws_region
ECS_NETWORK_MODE=bridge
```

```
CREATE_IAM_ROLES=True
ECS_TASK_ROLE_NAME=your_selected_ecs_task_role_name
ECS_EXECUTION_ROLE_NAME=your_selected_ecs_execution_role_name

aws cloudformation create-stack --stack-name CWAgent-Prometheus-ECS-
${ECS_CLUSTER_NAME}-EC2-${ECS_NETWORK_MODE} \
    --template-body file://cwagent-ecs-prometheus-metric-for-bridge-host.yaml \
    --parameters ParameterKey=ECSClusterName,ParameterValue=$ECS_CLUSTER_NAME \
                 ParameterKey=CreateIAMRoles,ParameterValue=$CREATE_IAM_ROLES \
                 ParameterKey=ECSNetworkMode,ParameterValue=$ECS_NETWORK_MODE \
                 ParameterKey=TaskRoleName,ParameterValue=$ECS_TASK_ROLE_NAME \
                 ParameterKey=ExecutionRoleName,ParameterValue=
$ECS_EXECUTION_ROLE_NAME \
    --capabilities CAPABILITY_NAMED_IAM \
    --region $AWS_REGION
```

Visualización de las métricas y los registros de NGINX Plus

Ahora se pueden visualizar las métricas de NGINX Plus que se están recopilando.

Para visualizar las métricas de la carga de trabajo de muestra de NGINX

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En la Región en la que se está ejecutando el clúster, elija Metrics (Métricas) en el panel de navegación izquierdo. Busque el espacio de nombres ContainerInsights/Prometheus para ver las métricas.
3. Para ver los eventos de CloudWatch Logs, elija Log groups (Grupos de registros) en el panel de navegación. Los eventos están en el grupo de registro /aws/containerinsights/***your_cluster_name***/prometheus, en el flujo de registros *nginx-plus-prometheus-exporter*.

Tutorial para añadir un nuevo destino de raspado de Prometheus: Memcached en Amazon ECS

Este tutorial proporciona una introducción práctica para raspar las métricas de Prometheus de una aplicación Memcached de muestra en un clúster de Amazon ECS con el tipo de lanzamiento de EC2. El agente CloudWatch detectará automáticamente el destino del exportador de Memcached de Prometheus mediante la detección de servicios basada en la definición de tareas de ECS.

Memcached es un sistema de almacenamiento en caché en memoria distribuida de uso general. A menudo se utiliza para acelerar los sitios web dirigidos por base de datos dinámicos mediante el almacenamiento en caché de datos y objetos en la RAM para reducir el número de veces que se debe leer una fuente de datos externa (como una base de datos o una API). Para obtener más información, consulte What is Memcached? (¿Qué es Memcached?)

memchached_exporter (Apache Licencia 2.0) es uno de los exportadores oficiales de Prometheus. De forma predeterminada, memcache_exporter sirve en el puerto 0.0.0.0:9150 en `/metrics`.

En este tutorial se utilizan las imágenes de Docker en los siguientes dos repositorios de Docker Hub:

- Memcached
- prom/memcached-exporter

Requisito previo

Para recopilar métricas de una carga de trabajo de muestra de Prometheus para Amazon ECS, se debe ejecutar Información de contenedores en el clúster. Para obtener más información sobre la instalación de Información de contenedores, consulte Configuración de Información de contenedores en Amazon ECS.

Temas

- Establezca las variables de entorno del clúster de EC2 de Amazon ECS
- Instale la carga de trabajo de muestra de Memcached
- Configure el agente de CloudWatch para que realice el raspado de las métricas de Memcached Prometheus
- Visualización de las métricas de Memcached

Establezca las variables de entorno del clúster de EC2 de Amazon ECS

Para establecer las variables de entorno del clúster de EC2 de Amazon ECS

1. Instale la CLI de Amazon ECS si aún no lo ha hecho. Para obtener más información, consulte Installing the Amazon ECS CLI (Instalación de la CLI de Amazon ECS).
2. Establezca el nuevo nombre del clúster de Amazon ECS y la Región. Por ejemplo:

```
ECS_CLUSTER_NAME=ecs-ec2-memcached-tutorial
```

```
AWS_DEFAULT_REGION=ca-central-1
```

3. (Opcional) Si aún no cuenta con un clúster de Amazon ECS con el tipo de lanzamiento de EC2 en el que desea instalar la carga de trabajo de muestra de Memcached y el agente de CloudWatch de ejemplo, puede crear uno con el siguiente comando.

```
ecs-cli up --capability-iam --size 1 \
--instance-type t3.medium \
--cluster $ECS_CLUSTER_NAME \
--region $AWS_REGION
```

El resultado esperado de este comando es el siguiente:

```
WARN[0000] You will not be able to SSH into your EC2 instances without a key pair.
INFO[0000] Using recommended Amazon Linux 2 AMI with ECS Agent 1.44.4 and Docker
 version 19.03.6-ce
INFO[0001] Created cluster                               cluster=ecs-ec2-memcached-
tutorial region=ca-central-1
INFO[0002] Waiting for your cluster resources to be created...
INFO[0002] Cloudformation stack status
 stackStatus=CREATE_IN_PROGRESS
INFO[0063] Cloudformation stack status
 stackStatus=CREATE_IN_PROGRESS
INFO[0124] Cloudformation stack status
 stackStatus=CREATE_IN_PROGRESS
VPC created: vpc-xxxxxxxxxxxxxxxxx
Security Group created: sg-xxxxxxxxxxxxxxxxx
Subnet created: subnet-xxxxxxxxxxxxxxxxx
Subnet created: subnet-xxxxxxxxxxxxxxxxx
Cluster creation succeeded.
```

Instale la carga de trabajo de muestra de Memcached

Para instalar la carga de trabajo de muestra de Memcached que expone las métricas de Prometheus

1. Descargue la plantilla de AWS CloudFormation de Memcached con el siguiente comando.

```
curl -O https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
insights/latest/ecs-task-definition-templates/deployment-mode/replica-service/
cwagent-prometheus/sample_traffic/memcached/memcached-traffic-sample.yaml
```

2. Establezca los nombres de rol de IAM que se crearán para Memcached con los siguientes comandos.

```
MEMCACHED_ECS_TASK_ROLE_NAME=memcached-prometheus-demo-ecs-task-role-name
MEMCACHED_ECS_EXECUTION_ROLE_NAME=memcached-prometheus-demo-ecs-execution-role-name
```

3. Instale la carga de trabajo de muestra de Memcached mediante el siguiente comando. Este ejemplo instala la carga de trabajo en el modo de red `host`.

```
MEMCACHED_ECS_NETWORK_MODE=host

aws cloudformation create-stack --stack-name Memcached-Prometheus-Demo-ECS-
$ECS_CLUSTER_NAME-EC2-$MEMCACHED_ECS_NETWORK_MODE \
    --template-body file://memcached-traffic-sample.yaml \
    --parameters ParameterKey=ECSClusterName,ParameterValue=$ECS_CLUSTER_NAME \
                 ParameterKey=ECSNetworkMode,ParameterValue=
$MEMCACHED_ECS_NETWORK_MODE \
                 ParameterKey=TaskRoleName,ParameterValue=
$MEMCACHED_ECS_TASK_ROLE_NAME \
                 ParameterKey=ExecutionRoleName,ParameterValue=
$MEMCACHED_ECS_EXECUTION_ROLE_NAME \
    --capabilities CAPABILITY_NAMED_IAM \
    --region $AWS_REGION
```

La pila de AWS CloudFormation crea cuatro recursos:

- Un rol de tarea de ECS
- Un rol de ejecución de tareas de ECS
- Una definición de tarea de Memcached
- Un servicio de Memcached

En la definición de tarea de Memcached, se definen dos contenedores:

- El contenedor principal ejecuta una aplicación de Memcached simple y abre el puerto 11 211 para el acceso.
- El otro contenedor ejecuta el proceso exportador de Redis para exponer las métricas de Prometheus en el puerto 9150. Este es el contenedor que el agente de CloudWatch detectará y raspará.

Configure el agente de CloudWatch para que realice el raspado de las métricas de Memcached Prometheus

Para configurar el agente de CloudWatch para que raspe las métricas de Memcached Prometheus

1. Descargue la última versión de `cwagent-ecs-prometheus-metric-for-awsvpc.yaml` mediante el siguiente comando.

```
curl -O https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-insights/latest/ecs-task-definition-templates/deployment-mode/replica-service/cwagent-prometheus/cloudformation-quickstart/cwagent-ecs-prometheus-metric-for-awsvpc.yaml
```

2. Abra el archivo con un editor de texto y busque la configuración completa del agente de CloudWatch detrás de la clave de `value` en la sección `resource:CWAgentConfigSSMParameter`.

   A continuación, en la sección `ecs_service_discovery`, agregue la siguiente configuración en la sección `task_definition_list`.

```
{
    "sd_job_name": "ecs-memcached",
    "sd_metrics_ports": "9150",
    "sd_task_definition_arn_pattern": ".*:task-definition/memcached-prometheus-demo.*:[0-9]+"
},
```

   Para la sección `metric_declaration`, la configuración predeterminada no permite ninguna métrica de Memcached. Agregue la siguiente sección para permitir métricas de Memcached. Asegúrese de seguir el patrón de sangría existente.

```
{
  "source_labels": ["container_name"],
  "label_matcher": "memcached-exporter-.*",
  "dimensions": [["ClusterName", "TaskDefinitionFamily"]],
  "metric_selectors": [
    "^memcached_current_(bytes|items|connections)$",
    "^memcached_items_(reclaimed|evicted)_total$",
    "^memcached_(written|read)_bytes_total$",
    "^memcached_limit_bytes$",
    "^memcached_commands_total$"
```

```
  ]
},
{
  "source_labels": ["container_name"],
  "label_matcher": "memcached-exporter-.*",
  "dimensions": [["ClusterName", "TaskDefinitionFamily","status","command"],
 ["ClusterName", "TaskDefinitionFamily","command"]],
  "metric_selectors": [
    "^memcached_commands_total$"
  ]
},
```

3. Si ya implementó el agente de CloudWatch en el clúster de Amazon ECS mediante AWS CloudFormation, puede crear un conjunto de cambios con los siguientes comandos.

```
ECS_NETWORK_MODE=bridge
CREATE_IAM_ROLES=True
ECS_TASK_ROLE_NAME=your_selected_ecs_task_role_name
ECS_EXECUTION_ROLE_NAME=your_selected_ecs_execution_role_name

aws cloudformation create-change-set --stack-name CWAgent-Prometheus-ECS-
${ECS_CLUSTER_NAME}-EC2-${ECS_NETWORK_MODE} \
    --template-body file://cwagent-ecs-prometheus-metric-for-bridge-host.yaml \
    --parameters ParameterKey=ECSClusterName,ParameterValue=$ECS_CLUSTER_NAME \
                 ParameterKey=CreateIAMRoles,ParameterValue=$CREATE_IAM_ROLES \
                 ParameterKey=ECSNetworkMode,ParameterValue=$ECS_NETWORK_MODE \
                 ParameterKey=TaskRoleName,ParameterValue=$ECS_TASK_ROLE_NAME \
                 ParameterKey=ExecutionRoleName,ParameterValue=
$ECS_EXECUTION_ROLE_NAME \
    --capabilities CAPABILITY_NAMED_IAM \
    --region $AWS_REGION \
    --change-set-name memcached-scraping-support
```

4. Abra la consola de AWS CloudFormation en https://console.aws.amazon.com/cloudformation.
5. Revise el conjunto de cambios recién creado `memcached-scraping-support`. Se debería ver un cambio aplicado al recurso `CWAgentConfigSSMParameter`. Ejecute el conjunto de cambios y reinicie la tarea del agente de CloudWatch mediante los siguientes comandos.

```
aws ecs update-service --cluster $ECS_CLUSTER_NAME \
--desired-count 0 \
--service cwagent-prometheus-replica-service-EC2-$ECS_NETWORK_MODE \
--region $AWS_REGION
```

6. Espere aproximadamente 10 segundos y, a continuación, ingrese el siguiente comando.

```
aws ecs update-service --cluster $ECS_CLUSTER_NAME \
--desired-count 1 \
--service cwagent-prometheus-replica-service-EC2-$ECS_NETWORK_MODE \
--region $AWS_REGION
```

7. Si va a instalar el agente de CloudWatch con la recopilación de métricas de Prometheus para el clúster por primera vez, ingrese los siguientes comandos:

```
ECS_NETWORK_MODEE=bridge
CREATE_IAM_ROLES=True
ECS_TASK_ROLE_NAME=your_selected_ecs_task_role_name
ECS_EXECUTION_ROLE_NAME=your_selected_ecs_execution_role_name

aws cloudformation create-stack --stack-name CWAgent-Prometheus-ECS-
${ECS_CLUSTER_NAME}-EC2-${ECS_NETWORK_MODE} \
    --template-body file://cwagent-ecs-prometheus-metric-for-bridge-host.yaml \
    --parameters ParameterKey=ECSClusterName,ParameterValue=$ECS_CLUSTER_NAME \
                 ParameterKey=CreateIAMRoles,ParameterValue=$CREATE_IAM_ROLES \
                 ParameterKey=ECSNetworkMode,ParameterValue=$ECS_NETWORK_MODE \
                 ParameterKey=TaskRoleName,ParameterValue=$ECS_TASK_ROLE_NAME \
                 ParameterKey=ExecutionRoleName,ParameterValue=
$ECS_EXECUTION_ROLE_NAME \
    --capabilities CAPABILITY_NAMED_IAM \
    --region $AWS_REGION
```

Visualización de las métricas de Memcached

En este tutorial se envían las siguientes métricas al espacio de nombres en CloudWatch ECS/ContainerInsights/Prometheus. Puede utilizar la consola de CloudWatch para ver las métricas de ese espacio de nombres.

Nombre de métrica	Dimensiones	
`memcached_current_items`	`ClusterName` , `TaskDefinitionFamily`	

Nombre de métrica	Dimensiones
memcached_current_connections	ClusterName , TaskDefinitionFamily
memcached_limit_bytes	ClusterName , TaskDefinitionFamily
memcached_current_bytes	ClusterName , TaskDefinitionFamily
memcached_written_bytes_total	ClusterName , TaskDefinitionFamily
memcached_read_bytes_total	ClusterName , TaskDefinitionFamily
memcached_items_evicted_total	ClusterName , TaskDefinitionFamily
memcached_items_reclaimed_total	ClusterName , TaskDefinitionFamily
memcached_commands_total	ClusterName , TaskDefinitionFamily ClusterName , TaskDefinitionFamily, comando ClusterName , TaskDefinitionFamily, estado, comando

> **Note**
>
> Los valores de la dimensión de command (comando) pueden ser: `delete`, `get`, `cas`, `set`, `decr`, `touch`, `incr` o `flush`.
> Los valores de la dimensión status (estado) pueden ser: `hit`, `miss` o `badval`.

También puede crear un panel de CloudWatch para las métricas de Memcached Prometheus.

Para crear un panel para las métricas de Memcached Prometheus

1. Cree variables de entorno mediante el reemplazo de los siguientes valores para que concuerden con la implementación.

```
DASHBOARD_NAME=your_memcached_cw_dashboard_name
ECS_TASK_DEF_FAMILY=memcached-prometheus-demo-$ECS_CLUSTER_NAME-EC2-
$MEMCACHED_ECS_NETWORK_MOD
```

2. Ingrese el siguiente comando para crear el panel.

```
curl https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-
container-insights/latest/ecs-task-definition-templates/deployment-mode/
replica-service/cwagent-prometheus/sample_cloudwatch_dashboards/memcached/
cw_dashboard_memcached.json \
| sed "s/{{YOUR_AWS_REGION}}/$AWS_REGION/g" \
| sed "s/{{YOUR_CLUSTER_NAME}}/$ECS_CLUSTER_NAME/g" \
| sed "s/{{YOUR_TASK_DEF_FAMILY}}/$ECS_TASK_DEF_FAMILY/g" \
| xargs -0 aws cloudwatch put-dashboard --dashboard-name ${DASHBOARD_NAME} --region
 $AWS_REGION --dashboard-body
```

Tutorial para el raspado de métricas de Redis Prometheus en Amazon ECS Fargate

En este tutorial encontrará una introducción práctica para raspar las métricas de Prometheus de una aplicación de Redis de muestra en un clúster de Amazon ECS Fargate. El agente de CloudWatch con compatibilidad con las métricas de Prometheus basado en las etiquetas docker del contenedor detectará automáticamente el destino del exportador de Redis Prometheus.

Redis (https://redis.io/) es un almacén de estructuras de datos en memoria de código abierto (con licencia BSD), utilizado como base de datos, caché y agente de mensajes. Para obtener más información, consulte redis.

redis_exporter (con licencia MIT) se utiliza para exponer las métricas de Redis prometheus en el puerto especificado (predeterminado: 0.0.0.0:9121). Para obtener más información, consulte redis_exporter.

En este tutorial se utilizan las imágenes de Docker en los siguientes dos repositorios de Docker Hub:

- redis
- redis_exporter

Requisito previo

Para recopilar métricas de una carga de trabajo de muestra de Prometheus para Amazon ECS, se debe ejecutar Información de contenedores en el clúster. Para obtener más información sobre la instalación de Información de contenedores, consulte Configuración de Información de contenedores en Amazon ECS.

Temas

- Cómo establecer la variable de entorno del clúster de Amazon ECS Fargate
- Establezca las variables de entorno de red para el clúster de Amazon ECS Fargate
- Instale la carga de trabajo de muestra de Redis
- Configure el agente de CloudWatch para que raspe las métricas de Redis Prometheus
- Visualización de las métricas de Redis

Cómo establecer la variable de entorno del clúster de Amazon ECS Fargate

Cómo establecer la variable de entorno del clúster de Amazon ECS Fargate

1. Instale la CLI de Amazon ECS si aún no lo ha hecho. Para obtener más información, consulte Installing the Amazon ECS CLI (Instalación de la CLI de Amazon ECS).
2. Establezca el nuevo nombre del clúster de Amazon ECS y la Región. Por ejemplo:

```
ECS_CLUSTER_NAME=ecs-fargate-redis-tutorial
AWS_DEFAULT_REGION=ca-central-1
```

3. (Opcional) Si aún no cuenta con un clúster de Amazon ECS Fargate en el que desea instalar la carga de trabajo de muestra de Redis y el agente de CloudWatch, puede crear uno con el siguiente comando.

```
ecs-cli up --capability-iam \
--cluster $ECS_CLUSTER_NAME \
--launch-type FARGATE \
--region $AWS_DEFAULT_REGION
```

El resultado esperado de este comando es el siguiente:

```
INFO[0000] Created cluster   cluster=ecs-fargate-redis-tutorial region=ca-central-1
INFO[0001] Waiting for your cluster resources to be created...
INFO[0001] Cloudformation stack status   stackStatus=CREATE_IN_PROGRESS
VPC created: vpc-xxxxxxxxxxxxxxxxx
Subnet created: subnet-xxxxxxxxxxxxxxxxx
Subnet created: subnet-xxxxxxxxxxxxxxxxx
Cluster creation succeeded.
```

Establezca las variables de entorno de red para el clúster de Amazon ECS Fargate

Para establecer las variables de entorno de red para el clúster de Amazon ECS Fargate

1. Establezca la VPC y el ID de subred del clúster de Amazon ECS. Si creó un clúster nuevo en el procedimiento anterior, verá estos valores en el resultado del comando final. De lo contrario, utilice los ID del clúster existente que va a utilizar con Redis.

   ```
   ECS_CLUSTER_VPC=vpc-xxxxxxxxxxxxxxxxx
   ECS_CLUSTER_SUBNET_1=subnet-xxxxxxxxxxxxxxxxx
   ECS_CLUSTER_SUBNET_2=subnet-xxxxxxxxxxxxxxxxx
   ```

2. En este tutorial, vamos a instalar la aplicación Redis y el agente de CloudWatch en el grupo de seguridad predeterminado de la VPC del clúster de Amazon ECS. El grupo de seguridad predeterminado permite todas las conexiones de red dentro del mismo grupo de seguridad para que el agente de CloudWatch pueda raspar las métricas de Prometheus expuestas en los contenedores de Redis. En un entorno de producción real, es posible que desee crear grupos de seguridad dedicados para la aplicación Redis y el agente de CloudWatch y establecer permisos personalizados para ellos.

   Para obtener el ID del grupo de seguridad predeterminado, ingrese el siguiente comando.

   ```
   aws ec2 describe-security-groups \
   --filters Name=vpc-id,Values=$ECS_CLUSTER_VPC  \
   ```

```
--region $AWS_DEFAULT_REGION
```

A continuación, establezca la variable de grupo de seguridad predeterminado del clúster de Fargate mediante el siguiente comando; debe reemplazar *my-default-security-group* por el valor que encontró del comando anterior.

```
ECS_CLUSTER_SECURITY_GROUP=my-default-security-group
```

Instale la carga de trabajo de muestra de Redis

Para instalar la carga de trabajo de muestra de Redis que expone las métricas de Prometheus

1. Descargue la plantilla de AWS CloudFormation de Redis con el siguiente comando.

```
curl -O https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
insights/latest/ecs-task-definition-templates/deployment-mode/replica-service/
cwagent-prometheus/sample_traffic/redis/redis-traffic-sample.yaml
```

2. Establezca los nombres de roles de IAM que se crearán para Redis al ingresar los siguientes comandos.

```
REDIS_ECS_TASK_ROLE_NAME=redis-prometheus-demo-ecs-task-role-name
REDIS_ECS_EXECUTION_ROLE_NAME=redis-prometheus-demo-ecs-execution-role-name
```

3. Instale la carga de trabajo de muestra de Redis con el siguiente comando.

```
aws cloudformation create-stack --stack-name Redis-Prometheus-Demo-ECS-
$ECS_CLUSTER_NAME-fargate-awsvpc \
    --template-body file://redis-traffic-sample.yaml \
    --parameters ParameterKey=ECSClusterName,ParameterValue=$ECS_CLUSTER_NAME \
                 ParameterKey=SecurityGroupID,ParameterValue=
$ECS_CLUSTER_SECURITY_GROUP \
                 ParameterKey=SubnetID,ParameterValue=$ECS_CLUSTER_SUBNET_1 \
                 ParameterKey=TaskRoleName,ParameterValue=$REDIS_ECS_TASK_ROLE_NAME
 \
                 ParameterKey=ExecutionRoleName,ParameterValue=
$REDIS_ECS_EXECUTION_ROLE_NAME \
    --capabilities CAPABILITY_NAMED_IAM \
    --region $AWS_DEFAULT_REGION
```

La pila de AWS CloudFormation crea cuatro recursos:

- Un rol de tarea de ECS
- Un rol de ejecución de tareas de ECS
- Una definición de tarea de Redis
- Un servicio de Redis

En la definición de tarea de Redis, se definen dos contenedores:

- El contenedor principal ejecuta una aplicación simple de Redis y abre el puerto 6379 para el acceso.
- El otro contenedor ejecuta el proceso exportador de Redis para exponer las métricas de Prometheus en el puerto 9121. Este es el contenedor que el agente de CloudWatch detectará y raspará. La siguiente etiqueta docker se define para que el agente de CloudWatch pueda detectar este contenedor mediante ella.

```
ECS_PROMETHEUS_EXPORTER_PORT: 9121
```

Configure el agente de CloudWatch para que raspe las métricas de Redis Prometheus

Para configurar el agente de CloudWatch para que raspe las métricas de Redis Prometheus

1. Descargue la última versión de `cwagent-ecs-prometheus-metric-for-awsvpc.yaml` con el siguiente comando.

   ```
   curl -O https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-insights/latest/ecs-task-definition-templates/deployment-mode/replica-service/cwagent-prometheus/cloudformation-quickstart/cwagent-ecs-prometheus-metric-for-awsvpc.yaml
   ```

2. Abra el archivo con un editor de texto y busque la configuración completa del agente de CloudWatch detrás de la clave de `value` en la sección `resource:CWAgentConfigSSMParameter`.

   A continuación, en la sección `ecs_service_discovery` que se muestra aquí, la detección del servicio basada en `docker_label` se habilita con la configuración predeterminada que se basa en `ECS_PROMETHEUS_EXPORTER_PORT`, que coincide con la etiqueta docker que se definió en

la definición de tarea de ECS de Redis. Por lo tanto, no es necesario hacer ningún cambio en esta sección:

```
ecs_service_discovery": {
  "sd_frequency": "1m",
  "sd_result_file": "/tmp/cwagent_ecs_auto_sd.yaml",
*  "docker_label": {
  },*
  ...
```

Para la sección `metric_declaration`, la configuración predeterminada no permite ninguna métrica de Redis. Agregue la siguiente sección para permitir las métricas de Redis. Asegúrese de seguir el patrón de sangría existente.

```
{
  "source_labels": ["container_name"],
  "label_matcher": "^redis-exporter-.*$",
  "dimensions": [["ClusterName","TaskDefinitionFamily"]],
  "metric_selectors": [
    "^redis_net_(in|out)put_bytes_total$",
    "^redis_(expired|evicted)_keys_total$",
    "^redis_keyspace_(hits|misses)_total$",
    "^redis_memory_used_bytes$",
    "^redis_connected_clients$"
  ]
},
{
  "source_labels": ["container_name"],
  "label_matcher": "^redis-exporter-.*$",
  "dimensions": [["ClusterName","TaskDefinitionFamily","cmd"]],
  "metric_selectors": [
    "^redis_commands_total$"
  ]
},
{
  "source_labels": ["container_name"],
  "label_matcher": "^redis-exporter-.*$",
  "dimensions": [["ClusterName","TaskDefinitionFamily","db"]],
  "metric_selectors": [
    "^redis_db_keys$"
  ]
},
```

3. Si ya implementó el agente de CloudWatch en el clúster de Amazon ECS mediante AWS CloudFormation, puede crear un conjunto de cambios con los siguientes comandos.

```
ECS_LAUNCH_TYPE=FARGATE
CREATE_IAM_ROLES=True
ECS_CLUSTER_SUBNET=$ECS_CLUSTER_SUBNET_1
ECS_TASK_ROLE_NAME=your_selected_ecs_task_role_name
ECS_EXECUTION_ROLE_NAME=your_selected_ecs_execution_role_name

aws cloudformation create-change-set --stack-name CWAgent-Prometheus-ECS-
$ECS_CLUSTER_NAME-$ECS_LAUNCH_TYPE-awsvpc \
    --template-body file://cwagent-ecs-prometheus-metric-for-awsvpc.yaml \
    --parameters ParameterKey=ECSClusterName,ParameterValue=$ECS_CLUSTER_NAME \
                 ParameterKey=CreateIAMRoles,ParameterValue=$CREATE_IAM_ROLES \
                 ParameterKey=ECSLaunchType,ParameterValue=$ECS_LAUNCH_TYPE \
                 ParameterKey=SecurityGroupID,ParameterValue=
$ECS_CLUSTER_SECURITY_GROUP \
                 ParameterKey=SubnetID,ParameterValue=$ECS_CLUSTER_SUBNET \
                 ParameterKey=TaskRoleName,ParameterValue=$ECS_TASK_ROLE_NAME \
                 ParameterKey=ExecutionRoleName,ParameterValue=
$ECS_EXECUTION_ROLE_NAME \
    --capabilities CAPABILITY_NAMED_IAM \
    --region ${AWS_DEFAULT_REGION} \
    --change-set-name redis-scraping-support
```

4. Abra la consola de AWS CloudFormation en https://console.aws.amazon.com/cloudformation.

5. Revise el conjunto de cambios recién creado `redis-scraping-support`. Se debería ver un cambio aplicado al recurso `CWAgentConfigSSMParameter`. Ejecute el conjunto de cambios y reinicie la tarea del agente de CloudWatch mediante los siguientes comandos.

```
aws ecs update-service --cluster $ECS_CLUSTER_NAME \
--desired-count 0 \
--service cwagent-prometheus-replica-service-$ECS_LAUNCH_TYPE-awsvpc \
--region ${AWS_DEFAULT_REGION}
```

6. Espere aproximadamente 10 segundos y, a continuación, ingrese el siguiente comando.

```
aws ecs update-service --cluster $ECS_CLUSTER_NAME \
--desired-count 1 \
--service cwagent-prometheus-replica-service-$ECS_LAUNCH_TYPE-awsvpc \
--region ${AWS_DEFAULT_REGION}
```

7. Si va a instalar el agente de CloudWatch con la recopilación de métricas de Prometheus para el clúster por primera vez, ingrese los siguientes comandos:

```
ECS_LAUNCH_TYPE=FARGATE
CREATE_IAM_ROLES=True
ECS_CLUSTER_SUBNET=$ECS_CLUSTER_SUBNET_1
ECS_TASK_ROLE_NAME=your_selected_ecs_task_role_name
ECS_EXECUTION_ROLE_NAME=your_selected_ecs_execution_role_name

aws cloudformation create-stack --stack-name CWAgent-Prometheus-ECS-
$ECS_CLUSTER_NAME-$ECS_LAUNCH_TYPE-awsvpc \
    --template-body file://cwagent-ecs-prometheus-metric-for-awsvpc.yaml \
    --parameters ParameterKey=ECSClusterName,ParameterValue=$ECS_CLUSTER_NAME \
                 ParameterKey=CreateIAMRoles,ParameterValue=$CREATE_IAM_ROLES \
                 ParameterKey=ECSLaunchType,ParameterValue=$ECS_LAUNCH_TYPE \
                 ParameterKey=SecurityGroupID,ParameterValue=
$ECS_CLUSTER_SECURITY_GROUP \
                 ParameterKey=SubnetID,ParameterValue=$ECS_CLUSTER_SUBNET \
                 ParameterKey=TaskRoleName,ParameterValue=$ECS_TASK_ROLE_NAME \
                 ParameterKey=ExecutionRoleName,ParameterValue=
$ECS_EXECUTION_ROLE_NAME \
    --capabilities CAPABILITY_NAMED_IAM \
    --region ${AWS_DEFAULT_REGION}
```

Visualización de las métricas de Redis

En este tutorial se envían las siguientes métricas al espacio de nombres en CloudWatch ECS/ContainerInsights/Prometheus. Puede utilizar la consola de CloudWatch para ver las métricas de ese espacio de nombres.

Nombre de métrica	Dimensiones	
`redis_net_input_bytes_total`	ClusterName, `TaskDefinitionFamily`	
`redis_net_output_bytes_total`	ClusterName, `TaskDefinitionFamily`	

Nombre de métrica	Dimensiones
redis_expired_keys_total	ClusterName, TaskDefinitionFamily
redis_evicted_keys_total	ClusterName, TaskDefinitionFamily
redis_keyspace_hits_total	ClusterName, TaskDefinitionFamily
redis_keyspace_misses_total	ClusterName, TaskDefinitionFamily
redis_memory_used_bytes	ClusterName, TaskDefinitionFamily
redis_connected_clients	ClusterName, TaskDefinitionFamily
redis_commands_total	ClusterName , TaskDefinitionFamily , cmd
redis_db_keys	ClusterName , TaskDefinitionFamily , db

> **Note**
>
> Los valores de la dimensión cmd pueden ser: append, client, command, config, dbsize, flushall, get, incr, info, latency o slowlog.
> Los valores de la dimensión db pueden ser db0 o db15.

También puede crear un panel de CloudWatch para las métricas de Redis Prometheus.

Para crear un panel para las métricas de Redis Prometheus

1. Cree variables de entorno al reemplazar los siguientes valores para que concuerden con la implementación.

```
DASHBOARD_NAME=your_cw_dashboard_name
ECS_TASK_DEF_FAMILY=redis-prometheus-demo-$ECS_CLUSTER_NAME-fargate-awsvpc
```

2. Ingrese el siguiente comando para crear el panel.

```
curl https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
insights/latest/k8s-deployment-manifest-templates/deployment-mode/service/cwagent-
prometheus/sample_cloudwatch_dashboards/redis/cw_dashboard_redis.json \
| sed "s/{{YOUR_AWS_REGION}}/${REGION_NAME}/g" \
| sed "s/{{YOUR_CLUSTER_NAME}}/${CLUSTER_NAME}/g" \
| sed "s/{{YOUR_NAMESPACE}}/${NAMESPACE}/g" \
```

Establezca y configure la recopilación de métricas de Prometheus en clústeres de Amazon EKS y de Kubernetes

Para recopilar métricas de Prometheus de clústeres que ejecutan Amazon EKS o Kubernetes, puede utilizar el agente de CloudWatch como el recopilador de AWS Distro para OpenTelemetry. Para obtener información sobre el uso del recopilador de AWS Distro para OpenTelemetry, consulte https://aws-otel.github.io/docs/getting-started/container-insights/eks-prometheus.

En las siguientes secciones se explica cómo se recopilan las métricas de Prometheus con el agente de CloudWatch. Explican cómo se instala el agente CloudWatch con supervisión de Prometheus en clústeres que ejecutan Amazon EKS o Kubernetes y cómo se configura el agente para que raspe los destinos adicionales. También proporcionan tutoriales opcionales para configurar cargas de trabajo de muestra para utilizarlas en pruebas con supervisión de Prometheus.

Temas

- Instale el agente de CloudWatch con la colección de métricas de Prometheus en clústeres de Amazon EKS y de Kubernetes

Instale el agente de CloudWatch con la colección de métricas de Prometheus en clústeres de Amazon EKS y de Kubernetes

En esta sección se explica cómo se configura el agente de CloudWatch con supervisión de Prometheus en un clúster que ejecute Amazon EKS o Kubernetes. Después de hacerlo, el agente raspa e importa automáticamente las métricas para las siguientes cargas de trabajo que se ejecutan en ese clúster.

- AWS App Mesh
- NGINX
- Memcached
- Java/JMX
- HAProxy
- Fluent Bit

También puede configurar el agente para que raspe e importe cargas de trabajo y fuentes de Prometheus adicionales.

Antes de seguir estos pasos e instalar el agente de CloudWatch para recopilar métricas de Prometheus, debe contar con un clúster que se ejecute en Amazon EKS o un clúster de Kubernetes que se ejecute en una instancia de Amazon EC2.

Requisitos del grupo de seguridad de la VPC

Las reglas de entrada de los grupos de seguridad para las cargas de trabajo de Prometheus deben abrir los puertos de Prometheus al agente de CloudWatch para raspar las métricas de Prometheus por la IP privada.

Las reglas de salida del grupo de seguridad para el agente de CloudWatch deben permitir que el agente de CloudWatch se conecte al puerto de cargas de trabajo de Prometheus mediante la IP privada.

Temas

- (Opcional) Configure las cargas de trabajo de muestra de Amazon EKS en contenedores para realizar pruebas con las métricas de Prometheus

Instale el agente de CloudWatch con la obtención de métricas de Prometheus en clústeres de Amazon EKS y de Kubernetes

En esta sección se explica cómo se configura el agente de CloudWatch con supervisión de Prometheus en un clúster que ejecute Amazon EKS o Kubernetes. Después de hacerlo, el agente raspa e importa automáticamente las métricas para las siguientes cargas de trabajo que se ejecutan en ese clúster.

- AWS App Mesh
- NGINX
- Memcached
- Java/JMX
- HAProxy
- Fluent Bit

También puede configurar el agente para que raspe e importe cargas de trabajo y fuentes de Prometheus adicionales.

Antes de seguir estos pasos e instalar el agente de CloudWatch para recopilar métricas de Prometheus, debe contar con un clúster que se ejecute en Amazon EKS o un clúster de Kubernetes que se ejecute en una instancia de Amazon EC2.

Requisitos del grupo de seguridad de la VPC

Las reglas de entrada de los grupos de seguridad para las cargas de trabajo de Prometheus deben abrir los puertos de Prometheus al agente de CloudWatch para raspar las métricas de Prometheus por la IP privada.

Las reglas de salida del grupo de seguridad para el agente de CloudWatch deben permitir que el agente de CloudWatch se conecte al puerto de cargas de trabajo de Prometheus mediante la IP privada.

Temas

- Configuración de roles de IAM

- Instalación del agente de CloudWatch para recopilar métricas de Prometheus

Configuración de roles de IAM

El primer paso es configurar el rol de IAM necesario en el clúster. Hay dos métodos:

- Configure un rol de IAM para una cuenta de servicio, también conocida como Función de servicio. Este método funciona tanto para el tipo de lanzamiento de EC2 como para el tipo de lanzamiento de Fargate.
- Agregue una política de IAM al rol de IAM que se utiliza en el clúster. Esto solo funciona para el tipo de lanzamiento de EC2.

Configure una función de servicio (tipo de lanzamiento de EC2 y tipo de lanzamiento de Fargate)

Para configurar una función de servicio, ingrese el siguiente comando. Sustituya *MyCluster* por el nombre del clúster.

```
eksctl create iamserviceaccount \
 --name cwagent-prometheus \
--namespace amazon-cloudwatch \
 --cluster MyCluster \
--attach-policy-arn arn:aws:iam::aws:policy/CloudWatchAgentServerPolicy \
--approve \
--override-existing-serviceaccounts
```

Agregue una política al rol de IAM del clúster (sólo tipo de lanzamiento de EC2)

Para configurar la política de IAM en un clúster para que sea compatible con Prometheus

1. Abra la consola de Amazon EC2 en https://console.aws.amazon.com/ec2/.
2. En el panel de navegación, seleccione Instances (Instancia[s]).
3. Debe encontrar el prefijo en el nombre del rol de IAM de ese clúster. Para ello, active la casilla situada junto al nombre de una instancia que se encuentre en el clúster y elija Acciones, Configuración de la instancia, Asociar/reemplazar rol de IAM. Después, copie el prefijo del rol de IAM; por ejemplo, `eksctl-dev303-workshop-nodegroup`.
4. Abra la consola de IAM en https://console.aws.amazon.com/iam/.
5. Seleccione Roles en el panel de navegación.

6. Utilice el cuadro de búsqueda para buscar el prefijo que copió anteriormente en este procedimiento y elija ese rol.
7. Seleccione Asociar políticas.
8. Utilice el cuadro de búsqueda para buscar CloudWatchAgentServerPolicy. Active la casilla de verificación situada junto a CloudWatchAgentServerPolicy y elija Asociar política.

Instalación del agente de CloudWatch para recopilar métricas de Prometheus

Debe instalarse el agente de CloudWatch en el clúster para recopilar las métricas. La instalación del agente es diferente en los clústeres de Amazon EKS y en los de Kubernetes.

Elimine las versiones anteriores del agente de CloudWatch compatible con Prometheus

Si ya ha instalado en el clúster una versión del agente de CloudWatch compatible con Prometheus, debe eliminar esa versión con el siguiente comando. Solo es necesario hacerlo con las versiones anteriores del agente compatible con Prometheus. No es necesario que elimine el agente de CloudWatch que habilita Información de contenedores sin compatibilidad con Prometheus.

```
kubectl delete deployment cwagent-prometheus -n amazon-cloudwatch
```

Instalación del agente de CloudWatch en clústeres de Amazon EKS con el tipo de lanzamiento de EC2

Para instalar el agente de CloudWatch compatible con Prometheus en un clúster de Amazon EKS, siga estos pasos.

Para instalar el agente de CloudWatch compatible con Prometheus en un clúster de Amazon EKS

1. Ejecute el siguiente comando para comprobar si el espacio de nombres de `amazon-cloudwatch` ya se ha creado:

   ```
   kubectl get namespace
   ```

2. Si `amazon-cloudwatch` no aparece en los resultados, créelo con el siguiente comando:

   ```
   kubectl create namespace amazon-cloudwatch
   ```

3. Para implementar el agente con la configuración predeterminada y hacer que envíe datos a la región de AWS en la que está instalado, escriba el siguiente comando:

```
kubectl apply -f https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-
container-insights/latest/k8s-deployment-manifest-templates/deployment-mode/
service/cwagent-prometheus/prometheus-eks.yaml
```

Para que el agente envíe datos a una otra región, siga estos pasos:

a. Descargue el archivo YAML del agente con el siguiente comando:

```
curl -O https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-
container-insights/latest/k8s-deployment-manifest-templates/deployment-mode/
service/cwagent-prometheus/prometheus-eks.yaml
```

b. Abra el archivo con un editor de texto y busque el bloque `cwagentconfig.json` del archivo.

c. Agregue las líneas resaltadas, especificando la región que desee:

```
cwagentconfig.json: |
    {
      "agent": {
        "region": "us-east-2"
      },
      "logs": { ...
```

d. Guarde el archivo e implemente el agente utilizando el archivo actualizado.

```
kubectl apply -f prometheus-eks.yaml
```

Instalación del agente de CloudWatch en clústeres de Amazon EKS con el tipo de lanzamiento de Fargate

Para instalar el agente de CloudWatch compatible con Prometheus en un clúster de Amazon EKS con el tipo de lanzamiento Fargate, siga estos pasos.

Para instalar el agente de CloudWatch compatible con Prometheus en un clúster de Amazon EKS con el tipo de lanzamiento de Fargate

1. Ingrese el siguiente comando para crear un perfil de Fargate para el agente de CloudWatch de modo que pueda ejecutarse dentro del clúster. Sustituya *MyCluster* por el nombre del clúster.

```
eksctl create fargateprofile --cluster MyCluster \
--name amazon-cloudwatch \
--namespace amazon-cloudwatch
```

2. Para instalar el agente de CloudWatch, ingrese el siguiente comando. Sustituya *MyCluster* por el nombre del clúster. Este nombre se utiliza en el nombre del grupo de registros donde se almacenan los eventos de registro recopilados por el agente y también se utiliza como dimensión en las métricas recopiladas por el agente.

   Sustituya *Región* por el nombre de la Región a la que desea enviar la métrica. Por ejemplo, `us-west-1`.

   ```
   curl https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
   insights/latest/k8s-deployment-manifest-templates/deployment-mode/service/cwagent-
   prometheus/prometheus-eks-fargate.yaml |
   sed "s/{{cluster_name}}/MyCluster/;s/{{region_name}}/region/" |
   kubectl apply -f -
   ```

Instalación del agente de CloudWatch en un clúster de Kubernetes

Para instalar el agente de CloudWatch compatible con Prometheus en un clúster en el que se ejecuta Kubernetes, ingrese el siguiente comando:

```
curl https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
insights/latest/k8s-deployment-manifest-templates/deployment-mode/service/cwagent-
prometheus/prometheus-k8s.yaml |
sed "s/{{cluster_name}}/MyCluster/;s/{{region_name}}/region/" |
kubectl apply -f -
```

Sustituya *MyCluster* por el nombre del clúster. Este nombre se utiliza en el nombre del grupo de registros donde se almacenan los eventos de registro recopilados por el agente y también se utiliza como dimensión en las métricas recopiladas por el agente.

Sustituya *region* por el nombre de la región de AWS a la que desea enviar la métrica. Por ejemplo, **`us-west-1`**.

Verifique que el agente esté en ejecución

En los clústeres de Amazon EKS y en los de Kubernetes, puede ingresar el siguiente comando para confirmar que el agente se está ejecutando.

```
kubectl get pod -l "app=cwagent-prometheus" -n amazon-cloudwatch
```

Si en los resultados solo aparece un pod del agente de CloudWatch con el estado `Running`, el agente está ejecutando y recopilando métricas de Prometheus. De forma predeterminada, el agente de CloudWatch recopila métricas para App Mesh, NGINX, Memcached, Java/JMX y HAProxy por minuto. Para obtener más información sobre estas métricas, consulte Métricas de Prometheus que el agente de CloudWatch recopila. Para obtener instrucciones sobre cómo ver las métricas de Prometheus en CloudWatch, consulte Visualización de las métricas de Prometheus

También puede configurar el agente de CloudWatch para que recopile métricas de otros exportadores de Prometheus. Para obtener más información, consulte El raspado de fuentes de Prometheus adicionales y la importación de tales métricas.

El raspado de fuentes de Prometheus adicionales y la importación de tales métricas

El agente CloudWatch con supervisión de Prometheus necesita dos configuraciones para raspar las métricas de Prometheus. Una de ellas es para las configuraciones estándar de Prometheus que como se documenta en <scrape_config> en la documentación de Prometheus. La otra es para la configuración del agente de CloudWatch.

Para los clústeres de Amazon EKS, las configuraciones se definen en `prometheus-eks.yaml` (para el tipo de lanzamiento de EC2) o en `prometheus-eks-fargate.yaml` (para el tipo de lanzamiento de Fargate) como dos mapas de configuración:

- La sección `name: prometheus-config` contiene la configuración de extracción de Prometheus,
- La sección `name: prometheus-cwagentconfig` contiene la configuración del agente de CloudWatch. Puede utilizar esta sección para definir cómo CloudWatch va a recopilar las métricas de Prometheus. Por ejemplo, puede especificar las métricas que se van a importar en CloudWatch y definir las dimensiones.

Para los clústeres de Kubernetes que se ejecutan en las instancias de Amazon EC2, las configuraciones se definen en el archivo YAML `prometheus-k8s.yaml` como dos mapas de configuración:

- La sección `name: prometheus-config` contiene la configuración de extracción de Prometheus,
- La sección `name: prometheus-cwagentconfig` contiene la configuración del agente de CloudWatch.

Para raspar las fuentes de las métricas de Prometheus adicionales e importarlas a CloudWatch, debe modifique tanto la configuración de raspado de Prometheus como la configuración del agente de CloudWatch y, a continuación, volver a implementar el agente con la configuración actualizada.

Requisitos del grupo de seguridad de la VPC

Las reglas de entrada de los grupos de seguridad para las cargas de trabajo de Prometheus deben abrir los puertos de Prometheus al agente de CloudWatch para raspar las métricas de Prometheus por la IP privada.

Las reglas de salida del grupo de seguridad para el agente de CloudWatch deben permitir que el agente de CloudWatch se conecte al puerto de cargas de trabajo de Prometheus mediante la IP privada.

Configuración de raspado de Prometheus

El agente de CloudWatch es compatible con la configuración de raspado estándar de Prometheus como se describe en <scrape_config> en la documentación de Prometheus. Se puede editar esta sección para actualizar las configuraciones que ya están en este archivo y agregar destinos adicionales de raspado de Prometheus. De forma predeterminada, el archivo de configuración de muestra contiene las siguientes líneas de configuración global:

```
global:
  scrape_interval: 1m
  scrape_timeout: 10s
```

- scrape_interval: define la frecuencia con la que se deben raspar los destinos.
- scrape_timeout: define cuánto tiempo debe esperar antes de que se agote el tiempo de espera de una petición de raspado.

También puede definir valores diferentes para estos parámetros en el nivel de trabajo, para anular las configuraciones globales.

Trabajos de raspado de Prometheus

Los archivos YAML del agente de CloudWatch ya tienen algunos trabajos de raspado configurados de forma predeterminada. Por ejemplo, en `prometheus-eks.yaml`, los trabajos de raspado predeterminados se configuran en las líneas `job_name` en la sección `scrape_configs`. En este archivo, en la siguiente sección predeterminada `kubernetes-pod-jmx` se raspan las métricas de JMX exporter.

```
- job_name: 'kubernetes-pod-jmx'
   sample_limit: 10000
   metrics_path: /metrics
   kubernetes_sd_configs:
   - role: pod
   relabel_configs:
   - source_labels: [__address__]
     action: keep
     regex: '.*:9404$'
   - action: labelmap
     regex: __meta_kubernetes_pod_label_(.+)
   - action: replace
     source_labels:
     - __meta_kubernetes_namespace
     target_label: Namespace
   - source_labels: [__meta_kubernetes_pod_name]
     action: replace
     target_label: pod_name
   - action: replace
     source_labels:
     - __meta_kubernetes_pod_container_name
     target_label: container_name
   - action: replace
     source_labels:
     - __meta_kubernetes_pod_controller_name
     target_label: pod_controller_name
   - action: replace
     source_labels:
     - __meta_kubernetes_pod_controller_kind
     target_label: pod_controller_kind
   - action: replace
     source_labels:
     - __meta_kubernetes_pod_phase
     target_label: pod_phase
```

Se realiza el raspado de cada uno de estos destinos predeterminados y las métricas se envían a CloudWatch en eventos de registro mediante el formato de métrica integrada. Para obtener más información, consulte Incrustar métricas en los registros.

Los eventos de registro de los clústeres de Amazon EKS y de Kubernetes se almacenan en el grupo de registro //aws/containerinsights/***cluster_name***/prometheus en CloudWatch Logs. Los eventos de registro de los clústeres de Amazon ECS se almacenan en el grupo de registro /aws/ecs/containerinsights/***cluster_name***/prometheus.

Cada trabajo de extracción está contenido en un flujo de registros diferente en este grupo de registros. Por ejemplo, el trabajo de raspado de Prometheus `kubernetes-pod-appmesh-envoy` se define para App Mesh. Todas las métricas de Prometheus de App Mesh de los clústeres de Amazon EKS y de Kubernetes se envían al flujo de registros denominado /aws/containerinsights/***cluster_name***>prometheus/kubernetes-pod-appmesh-envoy/.

Para agregar un nuevo destino de extracción, agregue una nueva sección `job_name` a la sección `scrape_configs` del archivo YAML y reinicie el agente. Para obtener un ejemplo de este proceso, consulte Tutorial para agregar un destino de raspado nuevo de Prometheus: métricas del servidor de la API de Prometheus.

Configuración del agente de CloudWatch para Prometheus

El archivo de configuración del agente de CloudWatch cuenta con una sección `prometheus` en `metrics_collected` para la configuración de raspado de Prometheus. Incluye las siguientes opciones de configuración:

- nombre_clúster: especifica el nombre del clúster que se va a agregar como etiqueta en el evento de registro. Este campo es opcional. Si lo omite, el agente puede detectar el nombre del clúster de Amazon EKS o de Kubernetes.
- log_group_name: especifica el nombre del grupo de registros para las métricas Prometheus raspadas. Este campo es opcional. Si lo omite, CloudWatch utiliza /aws/containerinsights/***cluster_name***/prometheus para los registros de los clústeres de Amazon EKS y de Kubernetes.
- prometheus_config_path: especifica la ruta del archivo de configuración de raspado de Prometheus. Si el valor de este campo comienza con `env:`, el contenido del archivo de configuración de raspado de Prometheus se recuperará de la variable de entorno del contenedor. No cambie este campo.

- ecs_service_discovery: es la sección en la que se especifica la configuración para la detección de servicios de Prometheus de Amazon ECS. Para obtener más información, consulte Guía detallada para la detección automática en clústeres de Amazon ECS.

  La sección `ecs_service_discovery` puede incluir los siguientes campos:

  - `sd_frequency` es la frecuencia para detectar a los exportadores de Prometheus. Especifique un número y un sufijo de unidad. Por ejemplo, `1m` para una vez por minuto o `30s` para una vez cada 30 segundos. Los sufijos de unidad válidos son `ns`, `us`, `ms`, `s`, `m` y `h`.

    Este campo es opcional. El valor predeterminado es 60 segundos (1 minuto).
  - `sd_target_cluster` es el nombre de clúster de Amazon ECS de destino para la detección automática. Este campo es opcional. El valor predeterminado es el nombre del clúster de Amazon ECS donde está instalado el agente de CloudWatch.
  - `sd_cluster_region` es la Región del clúster de Amazon ECS de destino. Este campo es opcional. El valor predeterminado es la Región del clúster de Amazon ECS donde está instalado el agente de CloudWatch.
  - `sd_result_file` es la ruta del archivo YAML para los resultados de destino de Prometheus. La configuración de raspado de Prometheus hará referencia a este archivo.
  - `docker_label` es una sección opcional que se puede utilizar para especificar la configuración para la detección de servicios basada en etiquetas docker. Si omite esta sección, no se utiliza la detección basada en etiquetas docker. Esta sección puede incluir los siguientes campos:
    - `sd_port_label` es el nombre de etiqueta docker del contenedor que especifica el puerto del contenedor para las métricas de Prometheus. El valor predeterminado es `ECS_PROMETHEUS_EXPORTER_PORT`. Si el contenedor no tiene esta etiqueta docker, el agente de CloudWatch lo omitirá.
    - `sd_metrics_path_label` es el nombre de etiqueta docker del contenedor que especifica la ruta de métricas de Prometheus. El valor predeterminado es `ECS_PROMETHEUS_METRICS_PATH`. Si el contenedor no tiene esta etiqueta docker, el agente asume la ruta predeterminada `/metrics`.
    - `sd_job_name_label` es el nombre de etiqueta docker del contenedor que especifica el nombre del trabajo de raspado de Prometheus. El valor predeterminado es `job`. Si el contenedor no tiene esta etiqueta docker, el agente de CloudWatch utiliza el nombre del trabajo en la configuración de raspado de Prometheus.
  - `task_definition_list` es una sección opcional que se puede utilizar para especificar la configuración de la detección de servicios basada en definiciones de tareas. Si omite esta

sección, no se utiliza la detección basada en definiciones de tareas. Esta sección puede incluir los siguientes campos:

- `sd_task_definition_arn_pattern` es el patrón que se utiliza para especificar las definiciones de tareas de Amazon ECS que se van a detectar. Esta es una expresión regular.
- `sd_metrics_ports` enumera el containerPort para las métricas de Prometheus. Separe los containerPorts con punto y coma.
- `sd_container_name_pattern` especifica los nombres de contenedor de tareas de Amazon ECS. Esta es una expresión regular.
- `sd_metrics_path` especifica la ruta de la métrica de Prometheus. Si omite esto, el agente asume la ruta de acceso predeterminada `/metrics`
- `sd_job_name` especifica el nombre del trabajo de raspado de Prometheus. Si omite este campo, el agente de CloudWatch utilizará el nombre de trabajo en la configuración de raspado de Prometheus.

- metric_declaration: son secciones que especifican la matriz de registros con formato de métrica integrada que se van a generar. Hay secciones `metric_declaration` para cada fuente de Prometheus desde las que el agente de CloudWatch importa de forma predeterminada. Cada una de estas secciones incluye los siguientes campos:
  - `label_matcher` es una expresión regular que verifica el valor de las etiquetas que aparecen en `source_labels`. Las métricas que concuerdan se pueden incorporar al formato de métrica integrada que se envía a CloudWatch.

    Si tiene varias etiquetas especificadas en `source_labels`, se recomienda que evite el uso de los caracteres `^` o `$` en la expresión regular para `label_matcher`.
  - `source_labels` especifica el valor de las etiquetas que se comprueban con `label_matcher`.
  - `label_separator` especifica el separador que se utilizará en la línea `label_matcher` si se especifican múltiples `source_labels`. El valor predeterminado es `;`. Puede ver este valor predeterminado utilizado en la línea `label_matcher` en el siguiente ejemplo.
  - `metric_selectors` es una expresión regular que especifica las métricas que se van a recopilar y enviar a CloudWatch.
  - `dimensions` es la lista de etiquetas que se van a utilizar como dimensiones de CloudWatch en cada métrica seleccionada.

Consulte el siguiente ejemplo, `metric_declaration`.

```
"metric_declaration": [
  {
     "source_labels":[ "Service", "Namespace"],
     "label_matcher":"(.*node-exporter.*|.*kube-dns.*);kube-system",
     "dimensions":[
        ["Service", "Namespace"]
     ],
     "metric_selectors":[
        "^coredns_dns_request_type_count_total$"
     ]
  }
]
```

En este ejemplo se configura una sección de formato de métricas integradas para que se envíe como evento de registro si se cumplen las condiciones siguientes:

- El valor de `Service` contiene `node-exporter` o `kube-dns`.
- El valor de `Namespace` es `kube-system`.
- La métrica de Prometheus `coredns_dns_request_type_count_total` contiene ambas etiquetas, `Namespace` y `Service`.

El evento de registro que se envía incluye la siguiente sección resaltada:

```
{
   "CloudWatchMetrics":[
      {
         "Metrics":[
            {
               "Name":"coredns_dns_request_type_count_total"
            }
         ],
         "Dimensions":[
            [
               "Namespace",
               "Service"
            ]
         ],
         "Namespace":"ContainerInsights/Prometheus"
      }
   ],
   "Namespace":"kube-system",
```

```
    "Service":"kube-dns",
    "coredns_dns_request_type_count_total":2562,
    "eks_amazonaws_com_component":"kube-dns",
    "instance":"192.168.61.254:9153",
    "job":"kubernetes-service-endpoints",
    ...
}
```

Tutorial para agregar un destino de raspado nuevo de Prometheus: métricas del servidor de la API de Prometheus

El servidor de la API de Kubernetes expone métricas de Prometheus en los puntos de enlace de forma predeterminada. El ejemplo oficial de la configuración de extracción del servidor de la API de Kubernetes está disponible en Github.

El siguiente tutorial muestra cómo se realizan los siguientes pasos para comenzar a importar métricas del servidor de la API de Kubernetes a CloudWatch:

- Agregue la configuración de raspado de Prometheus para el servidor de la API de Kubernetes al archivo YAML del agente de CloudWatch.
- Configuración de las definiciones de métricas del formato de métrica integrada en el archivo YAML del agente de CloudWatch.
- (Opcional) Creación de un panel de CloudWatch para las métricas del servidor de la API de Kubernetes.

> **Note**
>
> El servidor de la API de Kubernetes expone métricas de medidor, contador, histograma y resumen. En esta versión de compatibilidad con métricas de Prometheus, CloudWatch importa solo las métricas con tipos de medidor, contador y de resumen.

Para comenzar a recopilar métricas del servidor de la API de Kubernetes de Prometheus en CloudWatch

1. Descargue la última versión del archivo `prometheus-eks.yaml`, `prometheus-eks-fargate.yaml` o `prometheus-k8s.yaml` con uno de los siguientes comandos.

En el caso de un clúster de Amazon EKS con el tipo de lanzamiento de EC2, ingrese el siguiente comando:

```
curl -O https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-insights/latest/k8s-deployment-manifest-templates/deployment-mode/service/cwagent-prometheus/prometheus-eks.yaml
```

En el caso de un clúster de Amazon EKS con el tipo de lanzamiento de Fargate, ingrese el siguiente comando:

```
curl -O https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-insights/latest/k8s-deployment-manifest-templates/deployment-mode/service/cwagent-prometheus/prometheus-eks-fargate.yaml
```

En los clústeres de Kubernetes que se ejecuten en una instancia de Amazon EC2, ingrese el siguiente comando:

```
curl -O https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-insights/latest/k8s-deployment-manifest-templates/deployment-mode/service/cwagent-prometheus/prometheus-k8s.yaml
```

2. Abra el archivo con un editor de texto, busque la sección `prometheus-config` y agregue la siguiente sección dentro de esa sección. Para guardar los cambios:

```
    # Scrape config for API servers
    - job_name: 'kubernetes-apiservers'
      kubernetes_sd_configs:
        - role: endpoints
          namespaces:
            names:
              - default
      scheme: https
      tls_config:
        ca_file: /var/run/secrets/kubernetes.io/serviceaccount/ca.crt
        insecure_skip_verify: true
      bearer_token_file: /var/run/secrets/kubernetes.io/serviceaccount/token
      relabel_configs:
      - source_labels: [__meta_kubernetes_service_name,
 __meta_kubernetes_endpoint_port_name]
        action: keep
```

```
        regex: kubernetes;https
      - action: replace
        source_labels:
        - __meta_kubernetes_namespace
        target_label: Namespace
      - action: replace
        source_labels:
        - __meta_kubernetes_service_name
        target_label: Service
```

3. Con el archivo YAML abierto en el editor de texto, busque la sección `cwagentconfig.json`. Agregue la subsección siguiente y guarde los cambios. Esta sección coloca las métricas de servicio de la API en la lista de permitidos del agente de CloudWatch. Se agregan tres tipos de métricas de servidor de la API a la lista de permitidos:

   - Recuentos de objetos etcd
   - Métricas del controlador de registro del servidor de la API
   - Métricas de solicitud del servidor de la API

```
{"source_labels": ["job", "resource"],
  "label_matcher": "^kubernetes-apiservers;(services|daemonsets.apps|
deployments.apps|configmaps|endpoints|secrets|serviceaccounts|replicasets.apps)",
  "dimensions": [["ClusterName","Service","resource"]],
  "metric_selectors": [
  "^etcd_object_counts$"
  ]
},
{"source_labels": ["job", "name"],
   "label_matcher": "^kubernetes-apiservers;APIServiceRegistrationController$",
   "dimensions": [["ClusterName","Service","name"]],
   "metric_selectors": [
   "^workqueue_depth$",
   "^workqueue_adds_total$",
   "^workqueue_retries_total$"
  ]
},
{"source_labels": ["job","code"],
  "label_matcher": "^kubernetes-apiservers;2[0-9]{2}$",
  "dimensions": [["ClusterName","Service","code"]],
  "metric_selectors": [
   "^apiserver_request_total$"
```

```
  ]
},
{"source_labels": ["job"],
  "label_matcher": "^kubernetes-apiservers",
  "dimensions": [["ClusterName","Service"]],
  "metric_selectors": [
  "^apiserver_request_total$"
  ]
},
```

4. Si el agente de CloudWatch compatible con Prometheus ya está implementado en el clúster, debe eliminarlo con el siguiente comando:

```
kubectl delete deployment cwagent-prometheus -n amazon-cloudwatch
```

5. Implemente el agente de CloudWatch con la configuración actualizada mediante uno de los siguientes comandos. Para un clúster de Amazon EKS con el tipo de lanzamiento de EC2, ingrese:

```
kubectl apply -f prometheus-eks.yaml
```

Para un clúster de Amazon EKS con el tipo de lanzamiento de Fargate, ingrese el siguiente comando. Reemplace *MyCluster* y *Región* con valores que concuerden con la implementación.

```
cat prometheus-eks-fargate.yaml \
| sed "s/{{cluster_name}}/MyCluster/;s/{{region_name}}/region/" \
| kubectl apply -f -
```

En los clústeres de Kubernetes, ingrese el siguiente comando: Reemplace *MyCluster* y *Región* con valores que concuerden con la implementación.

```
cat prometheus-k8s.yaml \
| sed "s/{{cluster_name}}/MyCluster/;s/{{region_name}}/region/" \
| kubectl apply -f -
```

Una vez hecho esto, debería ver un nuevo flujo de registros denominado kubernetes-apiservers en el grupo de registros /aws/containerinsights/***nombre_clúster***/prometheus. Esta secuencia de

registros debe incluir eventos de registro con una definición de formato de métricas integradas como la siguiente:

```
{
   "CloudWatchMetrics":[
      {
         "Metrics":[
            {
               "Name":"apiserver_request_total"
            }
         ],
         "Dimensions":[
            [
               "ClusterName",
               "Service"
            ]
         ],
         "Namespace":"ContainerInsights/Prometheus"
      }
   ],
   "ClusterName":"my-cluster-name",
   "Namespace":"default",
   "Service":"kubernetes",
   "Timestamp":"1592267020339",
   "Version":"0",
   "apiserver_request_count":0,
   "apiserver_request_total":0,
   "code":"0",
   "component":"apiserver",
   "contentType":"application/json",
   "instance":"192.0.2.0:443",
   "job":"kubernetes-apiservers",
   "prom_metric_type":"counter",
   "resource":"pods",
   "scope":"namespace",
   "verb":"WATCH",
   "version":"v1"
}
```

Puede visualizar las métricas en la consola de CloudWatch, en el espacio de nombres ContainerInsights/Prometheus. También tiene la opción de crear un panel de CloudWatch para las métricas del servidor de la API de Kubernetes de Prometheus.

(Opcional) Creación de un panel para las métricas del servidor de la API de Kubernetes.

Para ver las métricas del servidor de la API de Kubernetes en el panel, primero debe haber completado los pasos de las secciones anteriores para comenzar a recopilar estas métricas en CloudWatch.

Para crear un panel para las métricas del servidor de la API de Kubernetes

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. Asegúrese de tener seleccionada la región de AWS correcta.
3. En el panel de navegación, seleccione Paneles.
4. Elija Crear un panel. Escriba el nombre del nuevo panel y elija Crear un panel.
5. En Añadir a este panel, elija Cancelar.
6. Elija Actions (Acciones), View/edit source (Ver/editar código fuente).
7. Descargue el siguiente archivo JSON: Kubernetes API Dashboard source.
8. Abra el archivo JSON que descargó con un editor de texto y realice los siguientes cambios:
   - Reemplace todas las cadenas `{{YOUR_CLUSTER_NAME}}` por el nombre exacto del clúster. Tenga cuidado de no agregar espacios en blanco antes o después del texto.
   - Reemplace todas las cadenas `{{YOUR_AWS_REGION}}` por el nombre de la región donde se recopilan las métricas. Por ejemplo, `us-west-2`. Asegúrese de no agregar espacios en blanco antes o después del texto.
9. Copie todo el blob JSON y péguelo en el cuadro de texto de la consola de CloudWatch de modo que sustituya lo que ya se encuentra en el cuadro.
10. Elija Actualizar y Guardar el panel.

(Opcional) Configure las cargas de trabajo de muestra de Amazon EKS en contenedores para realizar pruebas con las métricas de Prometheus

Para probar la compatibilidad de las métricas de Prometheus en Información de contenedores de CloudWatch, puede configurar una o varias de las siguientes cargas de trabajo en contenedores. El agente de CloudWatch compatible con Prometheus recopila automáticamente las métricas de cada una de estas cargas de trabajo. Para ver las métricas que se recopilan de forma predeterminada, consulte Métricas de Prometheus que el agente de CloudWatch recopila.

Para poder instalar cualquiera de estas cargas de trabajo, debe instalar Helm 3.x con los siguientes comandos:

```
brew install helm
```

Para obtener más información, consulte Helm.

Temas

- Configuración de las cargas de trabajo de muestra de AWS App Mesh para Amazon EKS y Kubernetes
- Configure NGINX con tráfico de muestra en Amazon EKS y Kubernetes
- Configure memcached con un exportador de métricas en Amazon EKS y Kubernetes
- Configure la carga de trabajo de muestra de Java/JMX en Amazon EKS y Kubernetes
- Configure HAProxy con un exportador de métricas en Amazon EKS y Kubernetes
- Tutorial para añadir un nuevo destino de raspado de Prometheus: Redis en clústeres de Amazon EKS y de Kubernetes

Configuración de las cargas de trabajo de muestra de AWS App Mesh para Amazon EKS y Kubernetes

La compatibilidad de Prometheus con Información de contenedores de CloudWatch es compatible con AWS App Mesh. En las secciones siguientes se explica cómo se configura App Mesh.

Información de contenedores de CloudWatch también puede recopilar registros de acceso de App Mesh Envoy. Para obtener más información, consulte (Opcional) Habilite los registros de acceso de App Mesh Envoy.

Temas

- Configuración de la carga de trabajo de muestra de AWS App Mesh en un clúster de Amazon EKS con el tipo de lanzamiento de EC2 o en un clúster de Kubernetes
- Configuración de la carga de trabajo de muestra de AWS App Mesh en un clúster de Amazon EKS con el tipo de lanzamiento de Fargate

Configuración de la carga de trabajo de muestra de AWS App Mesh en un clúster de Amazon EKS con el tipo de lanzamiento de EC2 o en un clúster de Kubernetes

Siga estas instrucciones si está configurando App Mesh en un clúster que ejecute Amazon EKS con el tipo de lanzamiento de EC2 o en un clúster de Kubernetes.

Configure los permisos de IAM

Debe agregar la política AWSAppMeshFullAccess al rol de IAM del grupo de nodos de Amazon EKS o Kubernetes. En Amazon EKS, el nombre de este grupo de nodos es similar a `eksctl-integ-test-eks-prometheus-NodeInstanceRole-ABCDEFHIJKL`. En Kubernetes, podría ser parecido a `nodes.integ-test-kops-prometheus.k8s.local`.

Instale App Mesh

Para instalar el controlador App Mesh Kubernetes, siga las instrucciones que se indican en App Mesh Controller (Controlador de App Mesh).

Instale una aplicación de muestra

aws-app-mesh-examples contiene varias explicaciones de Kubernetes App Mesh. Para este tutorial, se instala una aplicación de color de muestra en la que se puede ver cómo las rutas http pueden utilizar cabeceras para hacer concordar las peticiones entrantes.

Para utilizar una aplicación de ejemplo App Mesh para probar Información de contenedores

1. Instale la aplicación con estas instrucciones: https://github.com/aws/aws-app-mesh-examples/tree/main/walkthroughs/howto-k8s-http-headers.
2. Lance un pod curler para generar tráfico:

```
kubectl -n default run -it curler --image=tutum/curl /bin/bash
```

3. Utilice el comando curl con diferentes puntos de enlace al cambiar las cabeceras HTTP. Ejecute el comando curl varias veces, como se muestra:

```
curl -H "color_header: blue" front.howto-k8s-http-headers.svc.cluster.local:8080/;
 echo;

curl -H "color_header: red" front.howto-k8s-http-headers.svc.cluster.local:8080/;
 echo;

curl -H "color_header: yellow" front.howto-k8s-http-
headers.svc.cluster.local:8080/; echo;
```

4. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
5. En la región de AWS en la que se está ejecutando el clúster, elija Metrics (Métricas) en el panel de navegación. Las métricas están en el espacio de nombres ContainerInsights/Prometheus.

6. Para ver los eventos de CloudWatch Logs, elija Log groups (grupos de registros) en el panel de navegación. Los eventos están en el grupo de registro `/aws/containerinsights/`*`your_cluster_name`*`/prometheus` en el flujo de registros `kubernetes-pod-appmesh-envoy`.

Eliminación del entorno de prueba de App Mesh

Cuando haya terminado de usar App Mesh y la aplicación de muestra, utilice los siguientes comandos para eliminar los recursos innecesarios. Elimine la aplicación de muestra con el siguiente comando:

```
cd aws-app-mesh-examples/walkthroughs/howto-k8s-http-headers/
kubectl delete -f _output/manifest.yaml
```

Elimine el controlador de App Mesh con el siguiente comando:

```
helm delete appmesh-controller -n appmesh-system
```

Configuración de la carga de trabajo de muestra de AWS App Mesh en un clúster de Amazon EKS con el tipo de lanzamiento de Fargate

Siga estas instrucciones si está configurando App Mesh en un clúster que ejecute Amazon EKS con el tipo de lanzamiento de Fargate.

Configure los permisos de IAM

Para configurar los permisos de IAM, ingrese el siguiente comando. Sustituya *`MyCluster`* por el nombre del clúster.

```
eksctl create iamserviceaccount --cluster MyCluster \
 --namespace howto-k8s-fargate \
 --name appmesh-pod \
 --attach-policy-arn arn:aws:iam::aws:policy/AWSAppMeshEnvoyAccess \
 --attach-policy-arn arn:aws:iam::aws:policy/AWSCloudMapDiscoverInstanceAccess \
 --attach-policy-arn arn:aws:iam::aws:policy/AWSXRayDaemonWriteAccess \
 --attach-policy-arn arn:aws:iam::aws:policy/CloudWatchLogsFullAccess \
 --attach-policy-arn arn:aws:iam::aws:policy/AWSAppMeshFullAccess \
 --attach-policy-arn arn:aws:iam::aws:policy/AWSCloudMapFullAccess \
 --override-existing-serviceaccounts \
 --approve
```

Instale App Mesh

Para instalar el controlador de App Mesh Kubernetes, siga las instrucciones que se indican en App Mesh Controller (Controlador de App Mesh). Asegúrese de seguir las instrucciones de Amazon EKS con el tipo de lanzamiento de Fargate.

Instale una aplicación de muestra

aws-app-mesh-ejemplos contiene varias explicaciones de Kubernetes App Mesh. Para este tutorial, se instala una aplicación de color de muestra que funcione para clústeres de Amazon EKS con el tipo de lanzamiento de Fargate.

Para utilizar una aplicación de muestra App Mesh para probar Información de contenedores

1. Instale la aplicación con estas instrucciones: https://github.com/aws/aws-app-mesh-examples/tree/main/walkthroughs/howto-k8s-fargate.

   Esas instrucciones suponen que está creando un nuevo clúster con el perfil de Fargate correcto. Si desea utilizar un clúster de Amazon EKS que ya haya configurado, puede utilizar los siguientes comandos para configurar dicho clúster para esta demostración. Sustituya *MyCluster* por el nombre del clúster.

   ```
   eksctl create iamserviceaccount --cluster MyCluster \
    --namespace howto-k8s-fargate \
    --name appmesh-pod \
    --attach-policy-arn arn:aws:iam::aws:policy/AWSAppMeshEnvoyAccess \
    --attach-policy-arn arn:aws:iam::aws:policy/AWSCloudMapDiscoverInstanceAccess \
    --attach-policy-arn arn:aws:iam::aws:policy/AWSXRayDaemonWriteAccess \
    --attach-policy-arn arn:aws:iam::aws:policy/CloudWatchLogsFullAccess \
    --attach-policy-arn arn:aws:iam::aws:policy/AWSAppMeshFullAccess \
    --attach-policy-arn arn:aws:iam::aws:policy/AWSCloudMapFullAccess \
    --override-existing-serviceaccounts \
    --approve
   ```

   ```
   eksctl create fargateprofile --cluster MyCluster \
   --namespace howto-k8s-fargate --name howto-k8s-fargate
   ```

2. Reenvíe el despliegue de la aplicación frontal:

   ```
   kubectl -n howto-k8s-fargate port-forward deployment/front 8080:8080
   ```

3. Utilice el comando curl en la aplicación frontal:

```
while true; do  curl -s http://localhost:8080/color; sleep 0.1; echo ; done
```

4. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
5. En la región de AWS en la que se está ejecutando el clúster, elija Metrics (Métricas) en el panel de navegación. Las métricas están en el espacio de nombres ContainerInsights/Prometheus.
6. Para ver los eventos de CloudWatch Logs, elija Log groups (grupos de registros) en el panel de navegación. Los eventos están en el grupo de registro `/aws/containerinsights/your_cluster_name/prometheus` en el flujo de registros `kubernetes-pod-appmesh-envoy`.

Eliminación del entorno de prueba de App Mesh

Cuando haya terminado de usar App Mesh y la aplicación de muestra, utilice los siguientes comandos para eliminar los recursos innecesarios. Elimine la aplicación de muestra con el siguiente comando:

```
cd aws-app-mesh-examples/walkthroughs/howto-k8s-fargate/
kubectl delete -f _output/manifest.yaml
```

Elimine el controlador de App Mesh con el siguiente comando:

```
helm delete appmesh-controller -n appmesh-system
```

Configure NGINX con tráfico de muestra en Amazon EKS y Kubernetes

NGINX es un servidor web que también se puede utilizar como balanceador de carga y como proxy inverso. Para obtener más información sobre cómo Kubernetes utiliza NGINX para las entradas, consulte kubernetes/ingress-nginx.

Para instalar Ingress-NGINX con un servicio de tráfico de ejemplo para probar la compatibilidad de Información de contenedores de Prometheus

1. Ingrese el siguiente comando para agregar el repositorio de nginx de acceso de Helm.

```
helm repo add ingress-nginx https://kubernetes.github.io/ingress-nginx
```

2. Ejecute los comandos siguientes:

```
kubectl create namespace nginx-ingress-sample

helm install my-nginx ingress-nginx/ingress-nginx \
--namespace nginx-ingress-sample \
--set controller.metrics.enabled=true \
--set-string controller.metrics.service.annotations."prometheus\.io/port"="10254" \
--set-string controller.metrics.service.annotations."prometheus\.io/scrape"="true"
```

3. Compruebe si los servicios se iniciaron correctamente con el siguiente comando:

```
kubectl get service -n nginx-ingress-sample
```

   Deberían aparecer varias columnas, como la columna `EXTERNAL-IP`.

4. Defina una variable `EXTERNAL-IP` con el valor de la columna `EXTERNAL-IP` que aparece en la fila del controlador de entrada de NGINX.

```
EXTERNAL_IP=your-nginx-controller-external-ip
```

5. Ponga en marcha cierto tráfico de NGINX de ejemplo con el siguiente comando.

```
SAMPLE_TRAFFIC_NAMESPACE=nginx-sample-traffic
curl https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
insights/latest/k8s-deployment-manifest-templates/deployment-mode/service/cwagent-
prometheus/sample_traffic/nginx-traffic/nginx-traffic-sample.yaml |
sed "s/{{external_ip}}/$EXTERNAL_IP/g" |
sed "s/{{namespace}}/$SAMPLE_TRAFFIC_NAMESPACE/g" |
kubectl apply -f -
```

6. Utilice el siguiente comando para confirmar que los tres pods tienen el estado `Running`.

```
kubectl get pod -n $SAMPLE_TRAFFIC_NAMESPACE
```

   Si se están ejecutando, enseguida aparecerán las métricas en el espacio de nombres ContainerInsights/Prometheus .

Para desinstalar NGINX y la aplicación de tráfico de ejemplo

1. Elimine el servicio de tráfico de ejemplo con el siguiente comando:

```
kubectl delete namespace $SAMPLE_TRAFFIC_NAMESPACE
```

2. Elimine la salida de NGINX por el nombre de versión de Helm.

```
helm uninstall my-nginx --namespace nginx-ingress-sample
kubectl delete namespace nginx-ingress-sample
```

Configure memcached con un exportador de métricas en Amazon EKS y Kubernetes

Memcached es un sistema de almacenamiento en caché de objetos de memoria de código abierto. Para obtener más información, consulte What is Memcached?.

Si está ejecutando memcached en un clúster con el tipo de lanzamiento de Fargate, debe configurar un perfil de Fargate antes de seguir los pasos de este procedimiento. Para configurar el perfil, ingrese el siguiente comando: Sustituya *MyCluster* por el nombre del clúster.

```
eksctl create fargateprofile --cluster MyCluster \
--namespace memcached-sample --name memcached-sample
```

Si desea instalar Memcached con un exportador de métricas para probar la compatibilidad de Información de contenedores de Prometheus

1. Utilice el siguiente comando para agregar el repositorio.

```
helm repo add bitnami https://charts.bitnami.com/bitnami
```

2. Utilice el siguiente comando para crear un nuevo espacio de nombres:

```
kubectl create namespace memcached-sample
```

3. Utilice el siguiente comando para instalar Memcached:

```
helm install my-memcached bitnami/memcached --namespace memcached-sample \
--set metrics.enabled=true \
--set-string serviceAnnotations.prometheus\\.io/port="9150" \
--set-string serviceAnnotations.prometheus\\.io/scrape="true"
```

4. Utilice el siguiente comando para confirmar la anotación del servicio en ejecución:

```
kubectl describe service my-memcached-metrics -n memcached-sample
```

Debería ver estas dos anotaciones:

```
Annotations:   prometheus.io/port: 9150
               prometheus.io/scrape: true
```

Para desinstalar Memcached

- Ejecute los comandos siguientes:

```
helm uninstall my-memcached --namespace memcached-sample
kubectl delete namespace memcached-sample
```

Configure la carga de trabajo de muestra de Java/JMX en Amazon EKS y Kubernetes

JMX Exporter es un exportador oficial de Prometheus que puede extraer y exponer mBeans de JMX como métricas de Prometheus. Para obtener más información, consulte prometheus/jmx_exporter.

Información de contenedores puede recopilar métricas de Prometheus predefinidas de Java Virtual Machine (JVM), Java y Tomcat (Catalina) usando JMX Exporter.

Configuración predeterminada de raspado de Prometheus

De forma predeterminada, el agente de CloudWatch compatible con Prometheus raspa las métricas de Java/JMX de Prometheus de `http://CLUSTER_IP:9404/metrics` en cada pod en un clúster de Amazon EKS o de Kubernetes. Esto se realiza por la detección del `role: pod` de Prometheus `kubernetes_sd_config`. 9404 es el puerto predeterminado que Prometheus asignó para JMX Exporter. Para obtener más información acerca de la detección `role: pod`, consulte pod. Puede configurar JMX Exporter para exponer las métricas en un puerto o ruta (metrics_path) diferente. Si cambia el puerto o la ruta, actualice la scrape_config de jmx predeterminada en el mapa de configuración del agente de CloudWatch. Ejecute el siguiente comando para obtener la configuración de Prometheus del agente de CloudWatch actual:

```
kubectl describe cm prometheus-config -n amazon-cloudwatch
```

Los campos que se van a modificar son `/metrics` y `regex: '.*:9404$'`, que aparecen resaltados en el siguiente ejemplo.

```
job_name: 'kubernetes-jmx-pod'
sample_limit: 10000
metrics_path: /metrics
kubernetes_sd_configs:
- role: pod
relabel_configs:
- source_labels: [__address__]
  action: keep
  regex: '.*:9404$'
- action: replace
  regex: (.+)
  source_labels:
```

Otra configuración de raspado de Prometheus

Si expone la aplicación que se ejecuta en un conjunto de pods con exportadores de Java/JMX Prometheus por un servicio Kubernetes, también puede cambiar el uso de la detección de `role: service` o la detección del `role: endpoint` de Prometheus `kubernetes_sd_config`. Para obtener más información sobre estos métodos de detección, consulte service (servicio), endpoints (puntos de enlace) y <kubernetes_sd_config>..

Estos dos modos de detección de servicios proporcionan más metaetiquetas que podrían ser útiles para crear las dimensiones de métricas de CloudWatch. Por ejemplo, puede volver a etiquetar `__meta_kubernetes_service_name` a `Service` e incluirla en la dimensión de las métricas. Para obtener más información sobre cómo se personalizan las métricas de CloudWatch y las dimensiones, consulte Configuración del agente de CloudWatch para Prometheus.

Imagen de Docker con JMX Exporter

A continuación, cree una imagen de Docker. En las siguientes secciones, se incluyen dos Dockerfiles de muestra.

Cuando haya creado la imagen, cárguela en Amazon EKS o Kubernetes y ejecute el siguiente comando para verificar que `JMX_EXPORTER` ha expuesto las métricas de Prometheus en el puerto 9404. Sustituya *`$JAR_SAMPLE_TRAFFIC_POD`* por el nombre del pod en ejecución y *`$JAR_SAMPLE_TRAFFIC_NAMESPACE`* por el espacio de nombres de la aplicación.

Si está ejecutando JMX Exporter en un clúster con el tipo de lanzamiento de Fargate, también debe configurar un perfil de Fargate antes de seguir los pasos de este procedimiento. Para configurar el perfil, ingrese el comando siguiente: Sustituya *MyCluster* por el nombre del clúster.

```
eksctl create fargateprofile --cluster MyCluster \
--namespace $JAR_SAMPLE_TRAFFIC_NAMESPACE\
 --name $JAR_SAMPLE_TRAFFIC_NAMESPACE
```

```
kubectl exec $JAR_SAMPLE_TRAFFIC_POD -n $JARCAT_SAMPLE_TRAFFIC_NAMESPACE -- curl
 http://localhost:9404
```

Ejemplo: Imagen de Apache Tomcat Docker con métricas de Prometheus

De forma predeterminada, el servidor Apache Tomcat expone mBeans de JMX. Puede integrar JMX Exporter con Tomcat para que los mBeans de JMX se expongan como métricas de Prometheus. En el siguiente Dockerfile de ejemplo, se indican los pasos para crear una imagen de prueba:

```
# From Tomcat 9.0 JDK8 OpenJDK
FROM tomcat:9.0-jdk8-openjdk

RUN mkdir -p /opt/jmx_exporter

COPY ./jmx_prometheus_javaagent-0.12.0.jar /opt/jmx_exporter
COPY ./config.yaml /opt/jmx_exporter
COPY ./setenv.sh /usr/local/tomcat/bin
COPY your web application.war /usr/local/tomcat/webapps/

RUN chmod  o+x /usr/local/tomcat/bin/setenv.sh

ENTRYPOINT ["catalina.sh", "run"]
```

En la siguiente lista, se explican las cuatro líneas `COPY` de este Dockerfile.

- Descargue el último archivo jar de JMX Exporter de https://github.com/prometheus/jmx_exporter.
- `config.yaml` es el archivo de configuración de JMX Exporter. Para obtener más información, consulte https://github.com/prometheus/jmx_exporter#Configuration.

  Este es un archivo de configuración de ejemplo para Java y Tomcat:

  ```
  lowercaseOutputName: true
  lowercaseOutputLabelNames: true
  ```

```
rules:
- pattern: 'java.lang<type=OperatingSystem><>(FreePhysicalMemorySize|
TotalPhysicalMemorySize|FreeSwapSpaceSize|TotalSwapSpaceSize|SystemCpuLoad|
ProcessCpuLoad|OpenFileDescriptorCount|AvailableProcessors)'
  name: java_lang_OperatingSystem_$1
  type: GAUGE

- pattern: 'java.lang<type=Threading><>(TotalStartedThreadCount|ThreadCount)'
  name: java_lang_threading_$1
  type: GAUGE

- pattern: 'Catalina<type=GlobalRequestProcessor, name=\"(\w+-\w+)-(\d+)\"><>(\w+)'
  name: catalina_globalrequestprocessor_$3_total
  labels:
    port: "$2"
    protocol: "$1"
  help: Catalina global $3
  type: COUNTER

- pattern: 'Catalina<j2eeType=Servlet, WebModule=//([-a-zA-Z0-9+&@#/%?=~_|!:.,;]*[-
a-zA-Z0-9+&@#/%=~_|]), name=([-a-zA-Z0-9+/$%~_-|!.]*), J2EEApplication=none,
 J2EEServer=none><>(requestCount|maxTime|processingTime|errorCount)'
  name: catalina_servlet_$3_total
  labels:
    module: "$1"
    servlet: "$2"
  help: Catalina servlet $3 total
  type: COUNTER

- pattern: 'Catalina<type=ThreadPool, name="(\w+-\w+)-(\d+)"><>(currentThreadCount|
currentThreadsBusy|keepAliveCount|pollerThreadCount|connectionCount)'
  name: catalina_threadpool_$3
  labels:
    port: "$2"
    protocol: "$1"
  help: Catalina threadpool $3
  type: GAUGE

- pattern: 'Catalina<type=Manager, host=([-a-zA-Z0-9+&@#/%?=~_|!:.,;]*[-a-zA-
Z0-9+&@#/%=~_|]), context=([-a-zA-Z0-9+/$%~_-|!.]*)><>(processingTime|sessionCounter|
rejectedSessions|expiredSessions)'
  name: catalina_session_$3_total
  labels:
```

```
    context: "$2"
    host: "$1"
  help: Catalina session $3 total
  type: COUNTER

- pattern: ".*"
```

- `setenv.sh` es un script de startup de Tomcat que abre JMX Exporter junto con Tomcat y expone las métricas de Prometheus en el puerto 9404 de localhost. También proporciona a JMX Exporter la ruta del archivo `config.yaml`.

```
$ cat setenv.sh
export JAVA_OPTS="-javaagent:/opt/jmx_exporter/
jmx_prometheus_javaagent-0.12.0.jar=9404:/opt/jmx_exporter/config.yaml $JAVA_OPTS"
```

- application.war es el archivo de aplicación web `war` que se va a cargar en Tomcat.

Cree una imagen de Docker con esta configuración y cárguela en un repositorio de imágenes.

Ejemplo: Imagen de Docker de una aplicación Jar de Java con métricas de Prometheus

En el siguiente Dockerfile de muestra, se indican los pasos para crear una imagen de prueba:

```
# Alpine Linux with OpenJDK JRE
FROM openjdk:8-jre-alpine

RUN mkdir -p /opt/jmx_exporter

COPY ./jmx_prometheus_javaagent-0.12.0.jar /opt/jmx_exporter
COPY ./SampleJavaApplication-1.0-SNAPSHOT.jar /opt/jmx_exporter
COPY ./start_exporter_example.sh /opt/jmx_exporter
COPY ./config.yaml /opt/jmx_exporter

RUN chmod -R o+x /opt/jmx_exporter
RUN apk add curl

ENTRYPOINT exec /opt/jmx_exporter/start_exporter_example.sh
```

En la siguiente lista, se explican las cuatro líneas `COPY` de este Dockerfile.

- Descargue el último archivo jar de JMX Exporter de https://github.com/prometheus/jmx_exporter.

- `config.yaml` es el archivo de configuración de JMX Exporter. Para obtener más información, consulte https://github.com/prometheus/jmx_exporter#Configuration.

  Este es un archivo de configuración de ejemplo para Java y Tomcat:

```
lowercaseOutputName: true
lowercaseOutputLabelNames: true

rules:
- pattern: 'java.lang<type=OperatingSystem><>(FreePhysicalMemorySize|
TotalPhysicalMemorySize|FreeSwapSpaceSize|TotalSwapSpaceSize|SystemCpuLoad|
ProcessCpuLoad|OpenFileDescriptorCount|AvailableProcessors)'
  name: java_lang_OperatingSystem_$1
  type: GAUGE

- pattern: 'java.lang<type=Threading><>(TotalStartedThreadCount|ThreadCount)'
  name: java_lang_threading_$1
  type: GAUGE

- pattern: 'Catalina<type=GlobalRequestProcessor, name=\"(\w+-\w+)-(\d+)\"><>(\w+)'
  name: catalina_globalrequestprocessor_$3_total
  labels:
    port: "$2"
    protocol: "$1"
  help: Catalina global $3
  type: COUNTER

- pattern: 'Catalina<j2eeType=Servlet, WebModule=//([-a-zA-Z0-9+&@#/%?=~_|!:.,;]*[-
a-zA-Z0-9+&@#/%=~_|]), name=([-a-zA-Z0-9+/$%~_-|!.]*), J2EEApplication=none,
 J2EEServer=none><>(requestCount|maxTime|processingTime|errorCount)'
  name: catalina_servlet_$3_total
  labels:
    module: "$1"
    servlet: "$2"
  help: Catalina servlet $3 total
  type: COUNTER

- pattern: 'Catalina<type=ThreadPool, name="(\w+-\w+)-(\d+)"><>(currentThreadCount|
currentThreadsBusy|keepAliveCount|pollerThreadCount|connectionCount)'
  name: catalina_threadpool_$3
  labels:
    port: "$2"
    protocol: "$1"
```

```
  help: Catalina threadpool $3
  type: GAUGE

- pattern: 'Catalina<type=Manager, host=([-a-zA-Z0-9+&@#/%?=~_|!:.,;]*[-a-zA-
Z0-9+&@#/%=~_|]), context=([-a-zA-Z0-9+/$%~_-|!.]*)><>(processingTime|sessionCounter|
rejectedSessions|expiredSessions)'
  name: catalina_session_$3_total
  labels:
    context: "$2"
    host: "$1"
  help: Catalina session $3 total
  type: COUNTER

- pattern: ".*"
```

- `start_exporter_example.sh` es el script para iniciar la aplicación JAR con las métricas de Prometheus exportadas. También proporciona a JMX Exporter la ruta del archivo `config.yaml`.

```
$ cat start_exporter_example.sh
java -javaagent:/opt/jmx_exporter/jmx_prometheus_javaagent-0.12.0.jar=9404:/
opt/jmx_exporter/config.yaml -cp  /opt/jmx_exporter/SampleJavaApplication-1.0-
SNAPSHOT.jar com.gubupt.sample.app.App
```

- SampleJavaApplication-1.0-SNAPSHOT.jar es el archivo jar de la aplicación Java de ejemplo. Sustitúyalo por la aplicación Java que desee supervisar.

Cree una imagen de Docker con esta configuración y cárguela en un repositorio de imágenes.

Configure HAProxy con un exportador de métricas en Amazon EKS y Kubernetes

HAProxy es una aplicación proxy de código abierto. Para obtener más información, consulte HAProxy.

Si está ejecutando HAProxy en un clúster con el tipo de lanzamiento de Fargate, debe configurar un perfil de Fargate antes de seguir los pasos de este procedimiento. Para configurar el perfil, ingrese el siguiente comando: Sustituya *`MyCluster`* por el nombre del clúster.

```
eksctl create fargateprofile --cluster MyCluster \
--namespace haproxy-ingress-sample --name haproxy-ingress-sample
```

Si desea instalar HAProxy con un exportador de métricas para probar la compatibilidad de Información de contenedores de Prometheus

1. Ejecute el siguiente comando para agregar el repositorio de incubación de Helm:

```
helm repo add haproxy-ingress https://haproxy-ingress.github.io/charts
```

2. Utilice el siguiente comando para crear un nuevo espacio de nombres:

```
kubectl create namespace haproxy-ingress-sample
```

3. Ejecute los siguientes comandos para instalar HAProxy:

```
helm install haproxy haproxy-ingress/haproxy-ingress \
--namespace haproxy-ingress-sample \
--set defaultBackend.enabled=true \
--set controller.stats.enabled=true \
--set controller.metrics.enabled=true \
--set-string controller.metrics.service.annotations."prometheus\.io/port"="9101" \
--set-string controller.metrics.service.annotations."prometheus\.io/scrape"="true"
```

4. Ejecute el siguiente comando para confirmar la anotación del servicio:

```
kubectl describe service haproxy-haproxy-ingress-metrics -n haproxy-ingress-sample
```

   Debería ver las siguientes anotaciones.

```
Annotations:   prometheus.io/port: 9101
               prometheus.io/scrape: true
```

Para desinstalar HAProxy

- Ejecute los comandos siguientes:

```
helm uninstall haproxy --namespace haproxy-ingress-sample
kubectl delete namespace haproxy-ingress-sample
```

Tutorial para añadir un nuevo destino de raspado de Prometheus: Redis en clústeres de Amazon EKS y de Kubernetes

Este tutorial proporciona una introducción práctica para raspar las métricas de Prometheus de una aplicación de muestra de Redis en Amazon EKS y Kubernetes. Redis (https://redis.io/) es un almacén de estructura de datos en memoria de código abierto (con licencia BSD), que se utiliza como base de datos, caché y agente de mensajes. Para obtener más información, consulte redis.

redis_exporter (con licencia MIT) se utiliza para exponer las métricas de Redis prometheus en el puerto especificado (predeterminado: 0.0.0.0:9121). Para obtener más información, consulte redis_exporter.

En este tutorial se utilizan las imágenes de Docker en los siguientes dos repositorios de Docker Hub:

- redis
- redis_exporter

Para instalar una carga de trabajo de Redis de muestra que exponga las métricas de Prometheus

1. Establezca el espacio de nombres para la carga de trabajo de Redis de muestra.

```
REDIS_NAMESPACE=redis-sample
```

2. Si está ejecutando Redis en un clúster con el tipo de lanzamiento de Fargate, debe configurar un perfil de Fargate. Para configurar el perfil, ingrese el siguiente comando. Sustituya *MyCluster* por el nombre del clúster.

```
eksctl create fargateprofile --cluster MyCluster \
--namespace $REDIS_NAMESPACE --name $REDIS_NAMESPACE
```

3. Ingrese el siguiente comando para instalar la carga de trabajo Redis de muestra.

```
curl https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
insights/latest/k8s-deployment-manifest-templates/deployment-mode/service/cwagent-
prometheus/sample_traffic/redis/redis-traffic-sample.yaml \
| sed "s/{{namespace}}/$REDIS_NAMESPACE/g" \
| kubectl apply -f -
```

4. La instalación incluye un servicio llamado `my-redis-metrics` que expone la métrica de Redis Prometheus en el puerto 9121; ingrese el siguiente comando para obtener los detalles del servicio:

   ```
   kubectl describe service/my-redis-metrics  -n $REDIS_NAMESPACE
   ```

   En la sección `Annotations` de los resultados, verá dos anotaciones que concuerdan con la configuración de raspado de Prometheus del agente de CloudWatch, para que pueda detectar automáticamente las cargas de trabajo:

   ```
   prometheus.io/port: 9121
   prometheus.io/scrape: true
   ```

   La configuración de raspado de Prometheus relacionada se encuentra en la sección `- job_name: kubernetes-service-endpoints`de `kubernetes-eks.yaml` o `kubernetes-k8s.yaml`.

Para empezar a recopilar métricas de Redis Prometheus en CloudWatch

1. Descargue la última versión del archivo de `kubernetes-eks.yaml` o el archivo `kubernetes-k8s.yaml` con uno de los siguientes comandos. Para un clúster de Amazon EKS con el tipo de lanzamiento de EC2, ingrese este comando.

   ```
   curl -O https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-insights/latest/k8s-deployment-manifest-templates/deployment-mode/service/cwagent-prometheus/prometheus-eks.yaml
   ```

   Para un clúster de Amazon EKS con el tipo de lanzamiento de Fargate, ingrese este comando.

   ```
   curl -O https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-insights/latest/k8s-deployment-manifest-templates/deployment-mode/service/cwagent-prometheus/prometheus-eks-fargate.yaml
   ```

   En los clústeres de Kubernetes que se ejecuten en una instancia de Amazon EC2, ingrese el siguiente comando.

```
curl -O https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-insights/latest/k8s-deployment-manifest-templates/deployment-mode/service/cwagent-prometheus/prometheus-k8s.yaml
```

2. Abra el archivo con un editor de texto y busque la sección `cwagentconfig.json`. Agregue la subsección siguiente y guarde los cambios. Asegúrese de seguir el patrón existente de sangría.

```
{
  "source_labels": ["pod_name"],
  "label_matcher": "^redis-instance$",
  "dimensions": [["Namespace","ClusterName"]],
  "metric_selectors": [
    "^redis_net_(in|out)put_bytes_total$",
    "^redis_(expired|evicted)_keys_total$",
    "^redis_keyspace_(hits|misses)_total$",
    "^redis_memory_used_bytes$",
    "^redis_connected_clients$"
  ]
},
{
  "source_labels": ["pod_name"],
  "label_matcher": "^redis-instance$",
  "dimensions": [["Namespace","ClusterName","cmd"]],
  "metric_selectors": [
    "^redis_commands_total$"
  ]
},
{
  "source_labels": ["pod_name"],
  "label_matcher": "^redis-instance$",
  "dimensions": [["Namespace","ClusterName","db"]],
  "metric_selectors": [
    "^redis_db_keys$"
  ]
},
```

La sección que agregó coloca las métricas de Redis en la lista de permitidos del agente de CloudWatch. Para obtener la lista de estas métricas, consulte la siguiente sección.

3. Si el agente de CloudWatch compatible con Prometheus ya está implementado en el clúster, debe eliminarlo con el siguiente comando:

```
kubectl delete deployment cwagent-prometheus -n amazon-cloudwatch
```

4. Implemente el agente de CloudWatch con la configuración actualizada con uno de los siguientes comandos. Reemplace *MyCluster* y la *Región* para que concuerde con la configuración.

   Para un clúster de Amazon EKS con el tipo de lanzamiento de EC2, ingrese este comando.

   ```
   kubectl apply -f prometheus-eks.yaml
   ```

   Para un clúster de Amazon EKS con el tipo de lanzamiento de Fargate, ingrese este comando.

   ```
   cat prometheus-eks-fargate.yaml \
   | sed "s/{{cluster_name}}/MyCluster/;s/{{region_name}}/region/" \
   | kubectl apply -f -
   ```

   Para un clúster de Kubernetes, escriba este comando:

   ```
   cat prometheus-k8s.yaml \
   | sed "s/{{cluster_name}}/MyCluster/;s/{{region_name}}/region/" \
   | kubectl apply -f -
   ```

Visualización de las métricas de Redis Prometheus

En este tutorial se envían las siguientes métricas al espacio de nombres ContainerInsights/Prometheus en CloudWatch. Puede utilizar la consola de CloudWatch para ver las métricas de ese espacio de nombres.

Nombre de métrica	Dimensiones	
`redis_net_input_bytes_total`	ClusterName, `Namespace`	
`redis_net_output_bytes_total`	ClusterName, `Namespace`	

Nombre de métrica	Dimensiones	
`redis_expired_keys_total`	ClusterName, `Namespace`	
`redis_evicted_keys_total`	ClusterName, `Namespace`	
`redis_keyspace_hits_total`	ClusterName, `Namespace`	
`redis_keyspace_misses_total`	ClusterName, `Namespace`	
`redis_memory_used_bytes`	ClusterName, `Namespace`	
`redis_connected_clients`	ClusterName, `Namespace`	
`redis_commands_total`	ClusterName, `Namespace` , cmd	
`redis_db_keys`	ClusterName, `Namespace` , db	

> **Note**
>
> Los valores de la dimensión cmd pueden ser: `append`, `client`, `command`, `config`, `dbsize`, `flushall`, `get`, `incr`, `info`, `latency` o `slowlog`.
> Los valores de la dimensión db pueden ser `db0` o `db15`.

También puede crear un panel de CloudWatch para las métricas de Redis Prometheus.

Para crear un panel para las métricas de Redis Prometheus

1. Cree variables de entorno al reemplazar los siguientes valores para que concuerden con la implementación.

```
DASHBOARD_NAME=your_cw_dashboard_name
REGION_NAME=your_metric_region_such_as_us-east-1
CLUSTER_NAME=your_k8s_cluster_name_here
NAMESPACE=your_redis_service_namespace_here
```

2. Ingrese el siguiente comando para crear el panel.

```
curl https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
insights/latest/k8s-deployment-manifest-templates/deployment-mode/service/cwagent-
prometheus/sample_cloudwatch_dashboards/redis/cw_dashboard_redis.json \
| sed "s/{{YOUR_AWS_REGION}}/${REGION_NAME}/g" \
| sed "s/{{YOUR_CLUSTER_NAME}}/${CLUSTER_NAME}/g" \
| sed "s/{{YOUR_NAMESPACE}}/${NAMESPACE}/g" \
```

Conversión del tipo de métrica de Prometheus realizada por el agente de CloudWatch

Las bibliotecas de cliente de Prometheus ofrecen cuatro tipos de métricas principales:

- Contador
- Calibre
- Resumen
- Histograma

El agente de CloudWatch admite los tipos de métricas de contador, medición y resumen. La compatibilidad con métricas de tipo histograma está prevista para una próxima versión.

El agente de CloudWatch descarta las métricas de Prometheus con el tipo de métrica de histograma no compatible. Para obtener más información, consulte Registro de métricas de Prometheus descartadas.

Métricas de medición

Una métrica de medición de Prometheus es una métrica que representa un único valor numérico que puede subir y bajar arbitrariamente. El agente de CloudWatch raspa las métricas de medición y envía estos valores directamente.

Métricas de contador

Una métrica de contador de Prometheus es una métrica acumulada que representa un contador monotónico en aumento cuyo valor sólo puede aumentar o reiniciarse a cero. El agente de CloudWatch calcula un delta desde el raspado anterior y envía el valor delta como valor de métrica en el evento de registro. Por lo tanto, el agente de CloudWatch comenzará a producir un evento de registro desde el segundo raspado y continuará con los raspados posteriores, si los hay.

Métricas de resumen

Una métrica de resumen de Prometheus es un tipo de métrica compleja que está representada por varios puntos de datos. Proporciona un recuento total de observaciones y una suma de todos los valores observados. Calcula cuantiles configurables a través de una ventana de tiempo variable.

La suma y el recuento de una métrica de resumen son acumulativos, pero los cuantiles no lo son. El siguiente ejemplo muestra la desviación de los cuantiles.

```
# TYPE go_gc_duration_seconds summary
go_gc_duration_seconds{quantile="0"} 7.123e-06
go_gc_duration_seconds{quantile="0.25"} 9.204e-06
go_gc_duration_seconds{quantile="0.5"} 1.1065e-05
go_gc_duration_seconds{quantile="0.75"} 2.8731e-05
go_gc_duration_seconds{quantile="1"} 0.003841496
go_gc_duration_seconds_sum 0.37630427
go_gc_duration_seconds_count 9774
```

El agente de CloudWatch gestiona la suma y el recuento de una métrica de resumen de la misma manera que maneja las métricas de contador, como se describe en la sección anterior. El agente de CloudWatch conserva los valores del cuantil tal como se notifican originalmente.

Métricas de Prometheus que el agente de CloudWatch recopila

El agente de CloudWatch compatible con Prometheus recopila automáticamente las métricas de varios servicios y cargas de trabajo. En las siguientes secciones se detallan las métricas que se recopilan de forma predeterminada. También puede configurar el agente para que recopile más métricas de otros servicios y métricas de Prometheus desde diferentes aplicaciones y servicios. Para

obtener más información acerca de la recopilación de métricas adicionales, consulte Configuración del agente de CloudWatch para Prometheus.

Las métricas de Prometheus recopiladas de los clústeres de Amazon EKS y Kubernetes se encuentran en el espacio de nombres ContainerInsights/Prometheus. Las métricas de Prometheus que se recopilan de los clústeres de Amazon ECS se encuentran en el espacio de nombres ECS/ContainerInsights/Prometheus.

Temas

Métricas de Prometheus para App Mesh

Las siguientes métricas se recopilan automáticamente de App Mesh.

Información de contenedores de CloudWatch también puede recopilar registros de acceso de App Mesh Envoy. Para obtener más información, consulte (Opcional) Habilite los registros de acceso de App Mesh Envoy.

Métricas de Prometheus para App Mesh en clústeres de Amazon EKS y de Kubernetes

Nombre de métrica	Dimensiones	
`envoy_http_downstream_rq_total`	ClusterName, `Namespace`	
`envoy_http_downstream_rq_xx`	ClusterName, `Namespace` ClusterName, `Namespace` , envoy_http_conn_manager_prefix, envoy_response_code_class	

Nombre de métrica	Dimensiones	
envoy_cluster_upstream_cx_rx_bytes_total	ClusterName, Namespace	
envoy_cluster_upstream_cx_tx_bytes_total	ClusterName, Namespace	
envoy_cluster_membership_healthy	ClusterName, Namespace	
envoy_cluster_membership_total	ClusterName, Namespace	
envoy_server_memory_heap_size	ClusterName, Namespace	
envoy_server_memory_allocated	ClusterName, Namespace	
envoy_cluster_upstream_cx_connect_timeout	ClusterName, Namespace	
envoy_cluster_upstream_rq_pending_failure_eject	ClusterName, Namespace	

Nombre de métrica	Dimensiones	
envoy_cluster_upstream_rq_pending_overflow	ClusterName, Namespace	
envoy_cluster_upstream_rq_timeout	ClusterName, Namespace	
envoy_cluster_upstream_rq_try_per_timeout	ClusterName, Namespace	
envoy_cluster_upstream_rq_rx_reset	ClusterName, Namespace	
envoy_cluster_upstream_cx_destroy_local_with_active_rq	ClusterName, Namespace	
envoy_cluster_upstream_cx_destroy_remote_active_rq	ClusterName, Namespace	

Nombre de métrica	Dimensiones	
envoy_cluster_upstream_rq_maintenance_mode	ClusterName, Namespace	
envoy_cluster_upstream_flow_control_paused_reading_total	ClusterName, Namespace	
envoy_cluster_upstream_flow_control_resumed_reading_total	ClusterName, Namespace	
envoy_cluster_upstream_flow_control_backed_up_total	ClusterName, Namespace	
envoy_cluster_upstream_flow_control_drained_total	ClusterName, Namespace	

Nombre de métrica	Dimensiones	
envoy_cluster_upstream_rq_retry	ClusterName, Namespace	
envoy_cluster_upstream_rq_retry_success	ClusterName, Namespace	
envoy_cluster_upstream_rq_retry_overflow	ClusterName, Namespace	
envoy_server_live	ClusterName, Namespace	
envoy_server_uptime	ClusterName, Namespace	

Métricas de Prometheus para App Mesh en clústeres de Amazon ECS

Nombre de métrica	Dimensiones	
envoy_http_downstream_rq_total	ClusterName, TaskDefinitionFamily	
envoy_http_downstream_rq_xx	ClusterName, TaskDefinitionFamily	
envoy_cluster_upst	ClusterName, TaskDefinitionFamily	

Nombre de métrica	Dimensiones	
ream_cx_r x_bytes_total		
envoy_clu ster_upst ream_cx_t x_bytes_total	ClusterName, TaskDefinitionFamily	
envoy_clu ster_memb ership_healthy	ClusterName, TaskDefinitionFamily	
envoy_clu ster_memb ership_total	ClusterName, TaskDefinitionFamily	
envoy_ser ver_memor y_heap_size	ClusterName, TaskDefinitionFamily	
envoy_ser ver_memor y_allocated	ClusterName, TaskDefinitionFamily	
envoy_clu ster_upst ream_cx_c onnect_timeout	ClusterName, TaskDefinitionFamily	
envoy_clu ster_upst ream_rq_p ending_fa ilure_eject	ClusterName, TaskDefinitionFamily	

Nombre de métrica	Dimensiones	
envoy_cluster_upstream_rq_pending_overflow	ClusterName, TaskDefinitionFamily	
envoy_cluster_upstream_rq_timeout	ClusterName, TaskDefinitionFamily	
envoy_cluster_upstream_rq_try_per_timeout	ClusterName, TaskDefinitionFamily	
envoy_cluster_upstream_rq_rx_reset	ClusterName, TaskDefinitionFamily	
envoy_cluster_upstream_cx_destroy_local_with_active_rq	ClusterName, TaskDefinitionFamily	
envoy_cluster_upstream_cx_destroy_remote_active_rq	ClusterName, TaskDefinitionFamily	

Nombre de métrica	Dimensiones	
envoy_cluster_upstream_rq_maintenance_mode	ClusterName, TaskDefinitionFamily	
envoy_cluster_upstream_flow_control_paused_reading_total	ClusterName, TaskDefinitionFamily	
envoy_cluster_upstream_flow_control_resumed_reading_total	ClusterName, TaskDefinitionFamily	
envoy_cluster_upstream_flow_control_backed_up_total	ClusterName, TaskDefinitionFamily	
envoy_cluster_upstream_flow_control_drained_total	ClusterName, TaskDefinitionFamily	

Nombre de métrica	Dimensiones
envoy_cluster_upstream_rq_retry	ClusterName, TaskDefinitionFamily
envoy_cluster_upstream_rq_retry_success	ClusterName, TaskDefinitionFamily
envoy_cluster_upstream_rq_retry_overflow	ClusterName, TaskDefinitionFamily
envoy_server_live	ClusterName, TaskDefinitionFamily
envoy_server_uptime	ClusterName, TaskDefinitionFamily
envoy_http_downstream_rq_xx	ClusterName, TaskDefinitionFamily, envoy_http_conn_manager_prefix, envoy_response_code_class ClusterName, TaskDefinitionFamily, envoy_response_code_class

Note

TaskDefinitionFamily es el espacio de nombres Kubernetes de la malla.
El valor de envoy_http_conn_manager_prefix puede ser ingress, egress o admin.
El valor de envoy_response_code_class puede ser 1 (corresponde a 1xx), 2 corresponde a 2xx), 3 corresponde a 3xx), 4 corresponde a 4xx) o 5 que corresponde a 5xx).

Métricas de Prometheus para NGINX

Las siguientes métricas se recopilan automáticamente de NGINX en clústeres de Amazon EKS y de Kubernetes.

Nombre de métrica	Dimensiones	
`nginx_ingress_controller_nginx_process_cpu_seconds_total`	ClusterName, `Namespace` , Servicio	
`nginx_ingress_controller_success`	ClusterName, `Namespace` , Servicio	
`nginx_ingress_controller_requests`	ClusterName, `Namespace` , Servicio	
`nginx_ingress_controller_nginx_process_connections`	ClusterName, `Namespace` , Servicio	
`nginx_ingress_controller_nginx_process_connections_total`	ClusterName, `Namespace` , Servicio	
`nginx_ingress_cont`	ClusterName, `Namespace` , Servicio	

Nombre de métrica	Dimensiones	
roller_nginx_process_resident_memory_bytes		
nginx_ingress_controller_config_last_reload_successful	ClusterName, Namespace , Servicio	
nginx_ingress_controller_requests	ClusterName, Namespace , Servicio, estado	

Métricas de Prometheus para Memcached

Las siguientes métricas se recopilan automáticamente de Memcached en clústeres de Amazon EKS y de Kubernetes.

Nombre de métrica	Dimensiones	
memcached_current_items	ClusterName, Namespace , Servicio	
memcached_current_connections	ClusterName, Namespace , Servicio	
memcached_limit_bytes	ClusterName, Namespace , Servicio	

Nombre de métrica	Dimensiones
memcached_current_bytes	ClusterName, Namespace , Servicio
memcached_written_bytes_total	ClusterName, Namespace , Servicio
memcached_read_bytes_total	ClusterName, Namespace , Servicio
memcached_items_evicted_total	ClusterName, Namespace , Servicio
memcached_items_reclaimed_total	ClusterName, Namespace , Servicio
memcached_commands_total	ClusterName, Namespace , Servicio ClusterName, Namespace , Servicio, comando ClusterName, Namespace , Servicio, estado, comando

Métricas de Prometheus para Java/JMX

Métricas recopiladas en clústeres de Amazon EKS y de Kubernetes

En los clústeres de Amazon EKS y de Kubernetes, Información de contenedores puede recopilar las siguientes métricas de Prometheus predefinidas de Java Virtual Machine (JVM), Java y Tomcat (Catalina) mediante JMX Exporter. Para obtener más información, consulte prometheus/jmx_exporter en Github.

Java/JMX en clústeres de Amazon EKS y de Kubernetes

Nombre de métrica	Dimensiones
jvm_classes_loaded	ClusterName , Namespace
jvm_threads_current	ClusterName , Namespace
jvm_threads_daemon	ClusterName , Namespace
java_lang_operatingsystem_totalswapspacesize	ClusterName , Namespace
java_lang_operatingsystem_systemcpuload	ClusterName , Namespace
java_lang_operatingsystem_processcpuload	ClusterName , Namespace
java_lang_operatingsystem_freeswapspacesize	ClusterName , Namespace
java_lang_operatingsystem_totalphysicalmemorysize	ClusterName , Namespace

Nombre de métrica	Dimensiones	
java_lang_operatingsystem_freephysicalmemorysize	ClusterName , Namespace	
java_lang_operatingsystem_openfiledescriptorcount	ClusterName , Namespace	
java_lang_operatingsystem_availableprocessors	ClusterName , Namespace	
jvm_memory_bytes_used	ClusterName , Namespace , área	
jvm_memory_pool_bytes_used	ClusterName , Namespace , grupo	

> **Note**
>
> Los valores de la dimensión `area` pueden ser `heap` o `nonheap`.
> Los valores de la dimensión `pool` pueden ser `Tenured Gen`, `Compress Class Space`, `Survivor Space`, `Eden Space`, `Code Cache` o `Metaspace`.

TomCat/JMX en clústeres de Amazon EKS y de Kubernetes

Además de las métricas de Java/JMX de la tabla anterior, también se recopilan las siguientes métricas de la carga de trabajo de Tomcat.

Nombre de métrica	Dimensiones	
`catalina_manager_activesessions`	`ClusterName` , `Namespace`	
`catalina_manager_rejectedsessions`	`ClusterName` , `Namespace`	
`catalina_globalrequestprocessor_bytesreceived`	`ClusterName` , `Namespace`	
`catalina_globalrequestprocessor_bytessent`	`ClusterName` , `Namespace`	
`catalina_globalrequestprocessor_requestcount`	`ClusterName` , `Namespace`	
`catalina_globalrequestprocessor_errorcount`	`ClusterName` , `Namespace`	

Nombre de métrica	Dimensiones	
catalina_globalrequestprocessor_processingtime	ClusterName , Namespace	

Java/JMX en clústeres de Amazon ECS

Nombre de métrica	Dimensiones	
jvm_classes_loaded	ClusterName , TaskDefinitionFamily	
jvm_threads_current	ClusterName , TaskDefinitionFamily	
jvm_threads_daemon	ClusterName , TaskDefinitionFamily	
java_lang_operatingsystem_totalswapspacesize	ClusterName , TaskDefinitionFamily	
java_lang_operatingsystem_systemcpuload	ClusterName , TaskDefinitionFamily	
java_lang_operatingsystem_processcpuload	ClusterName , TaskDefinitionFamily	

Nombre de métrica	Dimensiones	
java_lang_operatingsystem_freeswapspacesize	ClusterName , TaskDefinitionFamily	
java_lang_operatingsystem_totalphysicalmemorysize	ClusterName , TaskDefinitionFamily	
java_lang_operatingsystem_freephysicalmemorysize	ClusterName , TaskDefinitionFamily	
java_lang_operatingsystem_openfiledescriptorcount	ClusterName , TaskDefinitionFamily	
java_lang_operatingsystem_availableprocessors	ClusterName , TaskDefinitionFamily	
jvm_memory_bytes_used	ClusterName , TaskDefinitionFamily, área	
jvm_memory_pool_bytes_used	ClusterName , TaskDefinitionFamily, grupo	

> **Note**
>
> Los valores de la dimensión `area` pueden ser `heap` o `nonheap`.
> Los valores de la dimensión `pool` pueden ser `Tenured Gen`, `Compress Class Space`, `Survivor Space`, `Eden Space`, `Code Cache` o `Metaspace`.

Tomcat/JMX en clústeres de Amazon ECS

Además de las métricas de Java/JMX de la tabla anterior, también se recopilan las siguientes métricas de la carga de trabajo de Tomcat en clústeres de Amazon ECS.

Nombre de métrica	Dimensiones	
`catalina_manager_activesessions`	`ClusterName` , `TaskDefinitionFamily`	
`catalina_manager_rejectedsessions`	`ClusterName` , `TaskDefinitionFamily`	
`catalina_globalrequestprocessor_bytesreceived`	`ClusterName` , `TaskDefinitionFamily`	
`catalina_globalrequestprocessor_bytessent`	`ClusterName` , `TaskDefinitionFamily`	
`catalina_globalrequestproce`	`ClusterName` , `TaskDefinitionFamily`	

Nombre de métrica	Dimensiones	
ssor_requestcount		
catalina_globalrequestprocessor_errorcount	ClusterName , TaskDefinitionFamily	
catalina_globalrequestprocessor_processingtime	ClusterName , TaskDefinitionFamily	

Métricas de Prometheus para HAProxy

Las siguientes métricas se recopilan automáticamente de HAProxy en clústeres de Amazon EKS y de Kubernetes.

Las métricas recopiladas dependen de la versión de HAProxy Ingress que esté utilizando. Para obtener más información sobre HAProxy Ingress y sus versiones, consulte haproxy-ingress.

Nombre de métrica	Dimensiones	Disponibilidad
haproxy_backend_bytes_in_total	ClusterName , Namespace , Servicio	Todas las versiones de HAProxy Ingress
haproxy_backend_bytes_out_total	ClusterName , Namespace , Servicio	Todas las versiones de HAProxy Ingress
haproxy_backend_co	ClusterName , Namespace , Servicio	Todas las versiones de HAProxy Ingress

Nombre de métrica	Dimensiones	Disponibilidad
nnection_ errors_total		
haproxy_b ackend_co nnections _total	ClusterName , Namespace , Servicio	Todas las versiones de HAProxy Ingress
haproxy_b ackend_cu rrent_sessions	ClusterName , Namespace , Servicio	Todas las versiones de HAProxy Ingress
haproxy_b ackend_ht tp_respon ses_total	ClusterName , Namespace , Servicio, código, backend	Todas las versiones de HAProxy Ingress
haproxy_b ackend_status	ClusterName , Namespace , Servicio	Sólo en versiones de HAProxy Ingress 0.10 o posteriores
haproxy_b ackend_up	ClusterName , Namespace , Servicio	Sólo en versiones de HAProxy Ingress anteriores a 0.10
haproxy_f rontend_b ytes_in_total	ClusterName , Namespace , Servicio	Todas las versiones de HAProxy Ingress
haproxy_f rontend_b ytes_out_total	ClusterName , Namespace , Servicio	Todas las versiones de HAProxy Ingress
haproxy_f rontend_c onnection s_total	ClusterName , Namespace , Servicio	Todas las versiones de HAProxy Ingress

Nombre de métrica	Dimensiones	Disponibilidad
haproxy_frontend_current_sessions	ClusterName , Namespace , Servicio	Todas las versiones de HAProxy Ingress
haproxy_frontend_http_requests_total	ClusterName , Namespace , Servicio	Todas las versiones de HAProxy Ingress
haproxy_frontend_http_responses_total	ClusterName ,Namespace , Servicio, código, frontend	Todas las versiones de HAProxy Ingress
haproxy_frontend_request_errors_total	ClusterName , Namespace , Servicio	Todas las versiones de HAProxy Ingress
haproxy_frontend_requests_denied_total	ClusterName , Namespace , Servicio	Todas las versiones de HAProxy Ingress

Note

Los valores de la dimensión `code` pueden ser `1xx`, `2xx`, `3xx`, `4xx`, `5xx` o `other`.
Los valores de la dimensión `backend` pueden ser:

- `http-default-backend`, `http-shared-backend` o `httpsback-shared-backend` para HAProxy Ingress versión 0.0.27 o anteriores.
- `_default_backend` para las versiones de HAProxy Ingress posteriores a 0.0.27.

Los valores de la dimensión `frontend` pueden ser:

- `httpfront-default-backend`, `httpfront-shared-frontend` o `httpfronts` para HAProxy Ingress versión 0.0.27 o anteriores.
- `_front_http` o `_front_https` para las versiones de HAProxy Ingress posteriores a 0.0.27.

Visualización de las métricas de Prometheus

Puede supervisión todas las métricas de Prometheus e crear alarmas sobre ellas, incluidas las que se seleccionaron y se agregaron previamente desde App Mesh, NGINX, Java/JMX, Memcached y HAProxy, así como de cualquier otro exportador de Prometheus que se haya agregado y configurado manualmente . Para obtener más información acerca de cómo recopilar métricas de otros exportadores de Prometheus, consulte Tutorial para agregar un destino de raspado nuevo de Prometheus: métricas del servidor de la API de Prometheus.

En la consola de CloudWatch, Información de contenedores proporciona los siguientes informes preconstruidos:

- Para los clústeres de Amazon EKS y de Kubernetes, hay informes preconstruidos para App Mesh, NGINX, HAPROXY, Memcached y Java/JMX.
- Para los clústeres de Amazon ECS, hay informes preconstruidos para App Mesh y Java/JMX.

Información de contenedores también proporciona paneles personalizados para cada una de las cargas de trabajo de las que Información de contenedores recopila métricas seleccionadas. Puede descargar estos paneles de GitHub

Para ver todas las métricas de Prometheus

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, seleccione Métricas.
3. En la lista de espacios de nombres, elija ContainerInsights/Prometheus o ECS/ContainerInsights/Prometheus.
4. Elija uno de los conjuntos de dimensiones de la siguiente lista. A continuación, active la casilla situada junto a las métricas que desee ver.

Para ver los informes prediseñados de las métricas de Prometheus

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Performance monitoring (Supervisión del rendimiento).
3. En el cuadro desplegable situado cerca de la parte superior de la página, elija cualquiera de las opciones Prometheus.

   En el otro cuadro desplegable, elija el clúster que desee ver

También se han incluido paneles personalizados para NGINX, App Mesh, Memcached, HAProxy y Java/JMX.

Para utilizar un panel personalizado proporcionado por Amazon

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Dashboards (Paneles).
3. Elija Crear un panel. Escriba el nombre del nuevo panel y elija Crear un panel.
4. En Añadir a este panel, elija Cancelar.
5. Elija Actions (Acciones), View/edit source (Ver/editar código fuente).
6. Descargue uno de los siguientes archivos JSON:
   - Origen de paneles personalizados para NGINX en Github.
   - App Mesh custom dashboard source on Github (Fuente de paneles personalizados de App Mesh en Github).
   - Origen de paneles personalizados de Memcached en Github
   - Origen de paneles personalizados de HAProxy-Ingress en Github
   - Origen de paneles personalizados para Java/JMX en Github.
7. Abra el archivo JSON que descargó con un editor de texto y realice los siguientes cambios:
   - Reemplace todas las cadenas `{{YOUR_CLUSTER_NAME}}` por el nombre exacto del clúster. Tenga cuidado de no agregar espacios en blanco antes o después del texto.
   - Reemplace todas las cadenas `{{YOUR_REGION}}` con la Región de AWS en la que se está ejecutando el clúster. Por ejemplo, en **`us-west-1`** asegúrese de no agregar espacios en blanco antes o después del texto.

- Reemplace todas las cadenas `{{YOUR_NAMESPACE}}` por el espacio de nombres exacto de la carga de trabajo.
- Reemplace todas las cadenas `{{YOUR_SERVICE_NAME}}` por el nombre exacto del servicio de la carga de trabajo. Por ejemplo, **`haproxy-haproxy-ingress-controller-metrics`**

8. Copie todo el blob JSON y péguelo en el cuadro de texto de la consola de CloudWatch de manera que sustituya el texto del cuadro.
9. Elija Actualizar y Guardar el panel.

Solución de problemas de las métricas de Prometheus

Esta sección le ayudará a solucionar problemas relacionados con la configuración de las métricas de Prometheus.

Temas

- Solución de problemas de las métricas de Prometheus en Amazon ECS
- Solución de problemas de métricas de Prometheus en clústeres de Amazon EKS y de Kubernetes

Solución de problemas de las métricas de Prometheus en Amazon ECS

Esta sección le ayudará a solucionar problemas relacionados con la configuración de las métricas de Prometheus en clústeres de Amazon ECS.

No se pueden ver las métricas de Prometheus que se enviaron a los CloudWatch Logs

Las métricas de Prometheus se deberían capturar como eventos de registro en el grupo de registros /aws/containerInsights/nombre-del-clúster/Prometheus. Si no se crea el grupo de registros o las métricas de Prometheus no se envían al grupo de registros, primero se deberá verificar si el agente de CloudWatch ha detectado correctamente los destinos de Prometheus. A continuación, verifique el grupo de seguridad y la configuración de permisos del agente de CloudWatch. Los siguientes pasos lo guiarán para realizar la depuración.

Paso 1: habilite el modo de depuración del agente de CloudWatch

Primero, cambie el agente de CloudWatch al modo de depuración al agregar los siguientes renglones en negrita al archivo de plantilla de AWS CloudFormation, `cwagent-ecs-prometheus-metric-for-bridge-host.yaml` o `cwagent-ecs-prometheus-metric-for-awsvpc.yaml`. A continuación, guarde el archivo.

```
cwagentconfig.json: |
    {
      "agent": {
        "debug": true
      },
      "logs": {
        "metrics_collected": {
```

Cree un nuevo conjunto de cambios de AWS CloudFormation con respecto a la pila existente. Establezca otros parámetros en el conjunto de cambios a los mismos valores que tiene en la pila de AWS CloudFormation. El siguiente ejemplo es para un agente de CloudWatch instalado en un clúster de Amazon ECS que utiliza el tipo de lanzamiento de EC2 y el modo de red puente.

```
ECS_NETWORK_MODE=bridge
 CREATE_IAM_ROLES=True
ECS_TASK_ROLE_NAME=your_selected_ecs_task_role_name
ECS_EXECUTION_ROLE_NAME=your_selected_ecs_execution_role_name
NEW_CHANGESET_NAME=your_selected_ecs_execution_role_name

aws cloudformation create-change-set --stack-name CWAgent-Prometheus-ECS-
${ECS_CLUSTER_NAME}-EC2-${ECS_NETWORK_MODE} \
    --template-body file://cwagent-ecs-prometheus-metric-for-bridge-host.yaml \
    --parameters ParameterKey=ECSClusterName,ParameterValue=$ECS_CLUSTER_NAME \
                 ParameterKey=CreateIAMRoles,ParameterValue=$CREATE_IAM_ROLES \
                 ParameterKey=ECSNetworkMode,ParameterValue=$ECS_NETWORK_MODE \
                 ParameterKey=TaskRoleName,ParameterValue=$ECS_TASK_ROLE_NAME \
                 ParameterKey=ExecutionRoleName,ParameterValue=$ECS_EXECUTION_ROLE_NAME
 \
    --capabilities CAPABILITY_NAMED_IAM \
    --region $AWS_REGION \
    --change-set-name $NEW_CHANGESET_NAME
```

Diríjase a la consola de AWS CloudFormation para revisar el nuevo conjunto de cambios, `$NEW_CHANGESET_NAME`. Debe haber un cambio aplicado al recurso CWAgentConfigSSMParameter. Ejecute el conjunto de cambios y reinicie la tarea del agente de CloudWatch mediante los siguientes comandos.

```
aws ecs update-service --cluster $ECS_CLUSTER_NAME \
--desired-count 0 \
--service your_service_name_here \
--region $AWS_REGION
```

Espere aproximadamente 10 segundos y, a continuación, ingrese el siguiente comando.

```
aws ecs update-service --cluster $ECS_CLUSTER_NAME \
--desired-count 1 \
--service your_service_name_here \
--region $AWS_REGION
```

Paso 2: verifique los registros de detección del servicio de ECS

La definición de tarea ECS del agente de CloudWatch habilita los registros de forma predeterminada en la siguiente sección. Los registros se envían a CloudWatch Logs en el grupo de registros /ecs/ecs-cwagent-prometheus.

```
LogConfiguration:
  LogDriver: awslogs
    Options:
      awslogs-create-group: 'True'
      awslogs-group: "/ecs/ecs-cwagent-prometheus"
      awslogs-region: !Ref AWS::Region
      awslogs-stream-prefix: !Sub 'ecs-${ECSLaunchType}-awsvpc'
```

Filtre los registros mediante la cadena `ECS_SD_Stats` para obtener las métricas relacionadas con la detección de servicios de ECS, tal y como se muestra en el siguiente ejemplo.

```
2020-09-1T01:53:14Z D! ECS_SD_Stats: AWSCLI_DescribeContainerInstances: 1
2020-09-1T01:53:14Z D! ECS_SD_Stats: AWSCLI_DescribeInstancesRequest: 1
2020-09-1T01:53:14Z D! ECS_SD_Stats: AWSCLI_DescribeTaskDefinition: 2
2020-09-1T01:53:14Z D! ECS_SD_Stats: AWSCLI_DescribeTasks: 1
2020-09-1T01:53:14Z D! ECS_SD_Stats: AWSCLI_ListTasks: 1
2020-09-1T01:53:14Z D! ECS_SD_Stats: Exporter_DiscoveredTargetCount: 1
2020-09-1T01:53:14Z D! ECS_SD_Stats: LRUCache_Get_EC2MetaData: 1
2020-09-1T01:53:14Z D! ECS_SD_Stats: LRUCache_Get_TaskDefinition: 2
2020-09-1T01:53:14Z D! ECS_SD_Stats: LRUCache_Size_ContainerInstance: 1
2020-09-1T01:53:14Z D! ECS_SD_Stats: LRUCache_Size_TaskDefinition: 2
2020-09-1T01:53:14Z D! ECS_SD_Stats: Latency: 43.399783ms
```

El significado de cada métrica para un ciclo de detección de servicios de ECS particular es el siguiente:

- AWSCLI_DescribeContainerInstances: el número de llamadas a la API `ECS::DescribeContainerInstances` realizadas.

- AWSCLI_DescribeInstancesRequest: el número de llamadas a la API `ECS::DescribeInstancesRequest` realizadas.
- AWSCLI_DescribetaskDefinition: el número de llamadas a la API `ECS::DescribeTaskDefinition` realizadas.
- AWSCLI_DescribeTasks: el número de llamadas a la API `ECS::DescribeTasks` realizadas.
- AWSCLI_ListTasks: el número de llamadas a la API `ECS::ListTasks` realizadas.
- ExporterDiscoveredTargetCount: el número de destinos de Prometheus que se detectaron y exportaron correctamente al archivo de resultados de destino dentro del contenedor.
- LRUCache_Get_EC2MetaData: el número de veces que se recuperaron los metadatos de instancias de contenedor de la caché.
- LRUCache_Get_TaskDefinition: el número de veces que los metadatos de definición de tareas de ECS se recuperaron de la caché.
- LRUCache_Size_ContainerInstance: el número de metadatos de la instancia de contenedor únicos almacenados en caché en la memoria.
- LRUCache_Size_TaskDefinition: el número de definiciones de tareas ECS únicas almacenadas en caché en la memoria.
- Latencia: cuánto tiempo tarda el ciclo de detección de servicios.

Verifique el valor de `ExporterDiscoveredTargetCount` para ver si los destinos de Prometheus detectados concuerdan con las expectativas. De lo contrario, las razones posibles son las siguientes:

- Es posible que la configuración de la detección de servicios de ECS no concuerde con la configuración de la aplicación. Para la detección de servicios basada en etiquetas docker, es posible que los contenedores de destino no tengan la etiqueta docker necesaria configurada en el agente de CloudWatch para detectarlos automáticamente. Para la detección de servicios basado en expresiones regulares ARN de la definición de tarea de ECS, es posible que la configuración de las expresiones regulares en el agente de CloudWatch no coincida con la definición de tarea de la aplicación.
- Es posible que el rol de tarea de ECS del agente de CloudWatch no tenga permiso para recuperar los metadatos de las tareas de ECS. Verifique que el agente de CloudWatch ha obtenido los siguientes permisos de sólo lectura:
  - `ec2:DescribeInstances`
  - `ecs:ListTasks`
  - `ecs:DescribeContainerInstances`

- `ecs:DescribeTasks`
- `ecs:DescribeTaskDefinition`

Paso 3: verifique la conexión de red y la política de rol de tareas de ECS

Si todavía no hay eventos de registro enviados al grupo de registro de CloudWatch Logs de destino aunque el valor de `Exporter_DiscoveredTargetCount` indica que hay destinos de Prometheus detectados, esto podría deberse a una de las siguientes causas:

- Es posible que el agente de CloudWatch no pueda conectarse a los puertos de destino de Prometheus. Verifique la configuración del grupo de seguridad detrás del agente de CloudWatch. La IP privada debe permitir que el agente de CloudWatch se conecte a los puertos del exportador de Prometheus.
- Es posible que el rol de tarea de ECS del agente de CloudWatch no cuente con la política administrada CloudWatchAgentServerPolicy. Se necesita contar con esta política para que el rol de tarea de ECS del agente de CloudWatch pueda enviar las métricas de Prometheus como eventos de registro. Si utilizó la plantilla de muestra de AWS CloudFormation para crear los roles de IAM de forma automática, tanto el rol de tarea de ECS como el rol de ejecución de ECS se otorgan con el privilegio mínimo para la supervisión de Prometheus.

Solución de problemas de métricas de Prometheus en clústeres de Amazon EKS y de Kubernetes

Esta sección proporciona ayuda para solucionar problemas relacionados con la configuración de las métricas de Prometheus en clústeres de Amazon EKS y de Kubernetes.

Pasos generales de solución de problemas en Amazon EKS

Ingrese el siguiente comando para confirmar que el agente de CloudWatch se está ejecutando.

```
kubectl get pod -n amazon-cloudwatch
```

En la salida, debería aparecer una fila con el valor `cwagent-prometheus-`*`id`* en la columna `NAME` y el valor `Running` en `STATUS column.`

Para obtener más información sobre el pod en ejecución, escriba el siguiente comando. Sustituya *`pod-name`* por el nombre completo del pod que debe comenzar por `cw-agent-prometheus`.

```
kubectl describe pod pod-name -n amazon-cloudwatch
```

Si Información de contenedores de CloudWatch está instalado, puede utilizar Información de registros de CloudWatch para consultar los registros del agente de CloudWatch que recopila las métricas de Prometheus.

Para consultar los registros de la aplicación

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija CloudWatch Logs Insights.
3. Seleccione el grupo de registros de la aplicación, /aws/containerinsights/***nombre-del-clúster***/application
4. Sustituya la expresión de la consulta de búsqueda por la siguiente consulta y elija Ejecutar la consulta.

```
fields ispresent(kubernetes.pod_name) as haskubernetes_pod_name, stream,
 kubernetes.pod_name, log |
filter haskubernetes_pod_name and kubernetes.pod_name like /cwagent-prometheus
```

También se puede confirmar que las métricas y los metadatos de Prometheus se están capturando como eventos de CloudWatch Logs.

Para confirmar que los datos de Prometheus se están ingiriendo

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija CloudWatch Logs Insights.
3. Seleccione /aws/containerinsights/***nombre-del-clúster***/prometheus
4. Sustituya la expresión de la consulta de búsqueda por la siguiente consulta y elija Ejecutar la consulta.

```
fields @timestamp, @message | sort @timestamp desc | limit 20
```

Registro de métricas de Prometheus descartadas

Esta versión no recopila métricas de Prometheus de tipo histograma. Puede utilizar el agente de CloudWatch para verificar si se está descartando alguna métrica de Prometheus porque son de tipo histograma. También puede registrar una lista de las primeras 500 métricas de Prometheus descartadas que no se envían a CloudWatch porque son de tipo histograma.

Para comprobar si se está descartando alguna métrica, ejecute el siguiente comando:

```
kubectl logs -l "app=cwagent-prometheus" -n amazon-cloudwatch --tail=-1
```

Si se está descartando alguna métrica, aparecerán las siguientes líneas en el archivo `/opt/aws/amazon-cloudwatch-agent/logs/amazon-cloudwatch-agent.log`.

```
I! Drop Prometheus metrics with unsupported types. Only Gauge, Counter and Summary are
 supported.
I! Please enable CWAgent debug mode to view the first 500 dropped metrics
```

Si ve esas líneas y desea saber qué métricas se están descartando, siga estos pasos.

Para registrar una lista de métricas de Prometheus descartadas

1. Cambie el agente de CloudWatch al modo de depuración al agregar las siguientes líneas en negrita al archivo `prometheus-eks.yaml` o `prometheus-k8s.yaml`, y guarde el archivo.

   ```
   {
         "agent": {
           "debug": true
         },
   ```

   Esta sección del archivo debería aparecer así:

   ```
   cwagentconfig.json: |
       {
         "agent": {
           "debug": true
         },
         "logs": {
           "metrics_collected": {
   ```

2. Vuelva a instalar el agente de CloudWatch para habilitar el modo de depuración con los siguientes comandos:

   ```
   kubectl delete deployment cwagent-prometheus -n amazon-cloudwatch
   kubectl apply -f prometheus.yaml
   ```

   Las métricas descartadas se registran en el pod del agente de CloudWatch.

3. Para recuperar los registros del pod del agente de CloudWatch, ingrese el siguiente comando:

```
kubectl logs -l "app=cwagent-prometheus" -n amazon-cloudwatch --tail=-1
```

   O bien, si tiene instalado el registro Fluentd de Información de contenedores, los registros también se guardan en el grupo de registro de registros de CloudWatch /aws/containerinsights/***cluster_name***/application.

   Para consultar estos registros, siga los pasos que se indican en Pasos generales de solución de problemas en Amazon EKS.

¿Dónde se capturan las métricas de Prometheus como eventos de registro de CloudWatch Logs?

El agente de CloudWatch crea un flujo de registros en la configuración de cada trabajo de raspado de Prometheus. Por ejemplo, en los archivos `prometheus-eks.yaml` y `prometheus-k8s.yaml`, la línea `job_name: 'kubernetes-pod-appmesh-envoy'` raspa las métricas de App Mesh. El destino de Prometheus está definido como `kubernetes-pod-appmesh-envoy`. Por tanto, todas las métricas de Prometheus de App Mesh se ingieren como eventos de CloudWatch Logs en el flujo de registro kubernetes-pod-appmesh-envoy bajo el grupo de registros /aws/containerinsights/cluster-name/Prometheus.

No se pueden ver las métricas de Amazon EKS o de Kubernetes Prometheus en las métricas de CloudWatch

En primer lugar, asegúrese de que las métricas de Prometheus se están ingiriendo como eventos de registro en el grupo de registros /aws/containerInsights/nombre-del-clúster/Prometheus. Utilice la información de ¿Dónde se capturan las métricas de Prometheus como eventos de registro de CloudWatch Logs? para que le ayude a comprobar el flujo de registro de destino. Si el flujo de registro no se ha creado o no hay nuevos eventos de registro en el flujo, compruebe lo siguiente:

- Compruebe que los puntos de enlace del exportador de métricas de Prometheus están configurados correctamente.
- Verifique que las configuraciones de raspado de Prometheus incluidas en la sección `config map: cwagent-prometheus` del archivo YAML del agente de CloudWatch sean correctas. La configuración debe ser la misma que la de un archivo de configuración de Prometheus. Para obtener más información, consulte <scrape_config> en la documentación de Prometheus.

Si las métricas de Prometheus se ingieren correctamente como eventos de registro, compruebe que la configuración de formato de métrica integrado se ha agregado a los eventos de registro para generar las métricas de CloudWatch.

```
"CloudWatchMetrics":[
   {
      "Metrics":[
         {
            "Name":"envoy_http_downstream_cx_destroy_remote_active_rq"
         }
      ],
      "Dimensions":[
         [
            "ClusterName",
            "Namespace"
         ]
      ],
      "Namespace":"ContainerInsights/Prometheus"
   }
],
```

Para obtener más información sobre el formato de métrica integrado, consulte Especificación: Formato de métricas integradas .

Si no hay ningún formato de métrica integrada en los eventos de registro, verifique que la sección `metric_declaration` esté configurada correctamente en la sección `config map: prometheus-cwagentconfig` del archivo YAML de instalación del agente de CloudWatch. Para obtener más información, consulte Tutorial para agregar un destino de raspado nuevo de Prometheus: métricas del servidor de la API de Prometheus.

Integración con Información de aplicaciones

Información de aplicaciones de Amazon CloudWatch ayuda a supervisar las aplicaciones e identifica y configura las métricas clave, los registros y las alarmas entre los recursos de aplicaciones y la pila de tecnología. Para obtener más información, consulte Información de aplicaciones de Amazon CloudWatch.

Puede habilitar Información de aplicaciones para recopilar datos adicionales de las aplicaciones y microservicios en contenedores. Si aún no lo ha hecho, puede habilitarlo al seleccionar Configuración automática de Información de aplicaciones debajo de la vista de rendimiento en el panel de Información de contenedores.

Si ya ha configurado Información de aplicaciones de CloudWatch para supervisar las aplicaciones en contenedores, el panel de Información de aplicaciones aparece debajo del panel de Información de contenedores.

Para obtener más información acerca de Información de aplicaciones y aplicaciones en contenedores, consulte Habilitación del monitoreo de recursos de Información de aplicaciones para Amazon ECS y Amazon EKS.

Consulte los eventos del ciclo de vida de Amazon ECS en Información de contenedores

Puede ver los eventos del ciclo de vida de Amazon ECS en la consola de Información de contenedores. Esto le ayuda a correlacionar las métricas, los registros y los eventos de sus contenedores en una sola vista para ofrecerle una visibilidad operativa más completa.

Los eventos incluyen eventos de cambio de estado de instancia de contenedor, eventos de cambio de estado de tarea y eventos de acciones de servicio. Amazon ECS los envía automáticamente a Amazon EventBridge y también se recopilan en CloudWatch en formato de registro de eventos. Para obtener más información acerca de estos eventos, consulte Eventos de Amazon ECS.

Se aplican los precios estándar de Container Insights para eventos de Amazon ECS Lifecycle. Para obtener más información, consulte Precios de Amazon CloudWatch.

Si quiere configurar la tabla de eventos del ciclo de vida y crear reglas para un clúster, debe disponer de los permisos `events:PutRule`, `events:PutTargets` y `logs:CreateLogGroup`. También debe asegurarse de que haya una política de recursos que permita a EventBridge crear el flujo de registros y enviar los registros a los registros de CloudWatch. Si esta política de recursos no existe, puede introducir el siguiente comando para crearla:

```
aws --region region logs put-resource-policy --policy-name 'EventBridgeCloudWatchLogs'
 --policy-document '{
    "Statement": [
        {
            "Action": [
                "logs:CreateLogStream",
                "logs:PutLogEvents"
            ],
            "Effect": "Allow",
            "Principal": {
                "Service": ["events.amazonaws.com", "delivery.logs.amazonaws.com"]
            },
```

```
            "Resource": "arn:aws:logs:region:account-id:log-group:/aws/events/ecs/
containerinsights/*:*",
            "Sid": "TrustEventBridgeToStoreECSLifecycleLogEvents"
        }
    ],
    "Version": "2012-10-17"
}'
```

Puede usar el siguiente comando para comprobar si ya dispone de esta política y para confirmar que la asociación funcionó correctamente.

```
aws logs describe-resource-policies --region region --output json
```

Para ver la tabla de los eventos del ciclo de vida, debe disponer de los permisos `events:DescribeRule`, `events:ListTargetsByRule` y `logs:DescribeLogGroups`.

Cómo ver los eventos del ciclo de vida de Amazon ECS en la consola de Información de contenedores de CloudWatch

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. Elija Información, Información de contenedores.
3. Seleccione Ver paneles de rendimiento.
4. En el siguiente menú desplegable, elija ECS Clusters (Clústeres de ECS), ECS Services (Servicios de ECS) o ECS Tasks (Tareas de ECS).
5. Si elige ECS Services (Servicios de ECS) o ECS Tasks (Tareas de ECS) en el paso anterior, seleccione la pestaña Lifecycle events (Eventos del ciclo de vida).
6. En la parte inferior de la página, si ve Configure lifecycle events (Configurar eventos del ciclo de vida), selecciónelo para crear reglas de EventBridge para el clúster.

   Los eventos se muestran debajo de los paneles de información del contenedor y encima de la sección Información de aplicaciones. Para ejecutar análisis adicionales y crear más visualizaciones sobre estos eventos, elija View in Logs Insights (Ver en Logs Insights) en la tabla Lifecycle Events (Eventos del ciclo de vida).

Solución de problemas de Información de contenedores

Las siguientes secciones pueden servir de ayuda si está teniendo problemas con Información de contenedores.

Error de implementación en Amazon EKS o Kubernetes

Si el agente no se implementa correctamente en un clúster de Kubernetes, pruebe lo siguiente:

- Ejecute el siguiente comando para obtener la lista de pods.

```
kubectl get pods -n amazon-cloudwatch
```

- Ejecute el siguiente comando y compruebe los eventos de la parte inferior de la salida.

```
kubectl describe pod pod-name -n amazon-cloudwatch
```

- Ejecute el siguiente comando para comprobar los registros.

```
kubectl logs pod-name -n amazon-cloudwatch
```

Pánico no autorizado: no se puede recuperar datos de cadvisor desde kubelet

Si su implementación falla con el error `Unauthorized panic: Cannot retrieve cadvisor data from kubelet`, su kubelet podría no tener el modo de autorización Webhook habilitado. Este modo es obligatorio para Información de contenedores. Para obtener más información, consulte Verificación de los requisitos previos de .

Implementación de Información de contenedores en un clúster que se eliminó y recreó en Amazon ECS

Si elimina un clúster de Amazon ECS existente que no tiene habilitado Información de contenedores y lo vuelve a crear con el mismo nombre, no podrá habilitar Información de contenedores en ese nuevo clúster en el momento de volver a crearlo. Puede habilitarlo si lo vuelve a crear y, a continuación, escribe el siguiente comando:

```
aws ecs update-cluster-settings --cluster myCICluster --settings
 name=containerInsights,value=enabled
```

Error de punto de enlace inválido

Si aparece un mensaje de error similar al siguiente, compruebe que ha reemplazado todos los marcadores de posición, como *nombre-clúster* y *nombre-región*, por la información correcta para la implementación en los comandos que utiliza.

```
"log": "2020-04-02T08:36:16Z E! cloudwatchlogs: code: InvalidEndpointURL, message:
 invalid endpoint uri, original error: &url.Error{Op:\"parse\", URL:\"https://
logs.{{region_name}}.amazonaws.com/\", Err:\"{\"}, &awserr.baseError{code:
\"InvalidEndpointURL\", message:\"invalid endpoint uri\", errs:[]error{(*url.Error)
(0xc0008723c0)}}\n",
```

Las métricas no aparecen en la consola

Si no ve ningún métrica de Información de contenedores en la AWS Management Console, asegúrese de haber completado la configuración de Información de contenedores. Las métricas no aparecen antes de haber configurado por completo Información de contenedores. Para obtener más información, consulte Configuración de Información de contenedores.

Faltan métricas de pod en Amazon EKS o en Kubernetes después de actualizar el clúster

Esta sección puede ser útil si faltan todas o algunas métricas de pod después de implementar el agente de CloudWatch como un conjunto de daemons en un clúster nuevo o actualizado, o si ve un registro de errores con el mensaje `W! No pod metric collected`.

Estos errores pueden deberse a cambios en el tiempo de ejecución del contenedor, como containerd o el controlador docker systemd cgroup. Por lo general, puede resolverlo mediante la actualización del manifiesto de implementación para que el socket containerd del host esté montado en el contenedor. Vea el siguiente ejemplo:

```
# For full example see https://github.com/aws-samples/amazon-cloudwatch-container-
insights/blob/latest/k8s-deployment-manifest-templates/deployment-mode/daemonset/
container-insights-monitoring/cwagent/cwagent-daemonset.yaml
apiVersion: apps/v1
kind: DaemonSet
metadata:
  name: cloudwatch-agent
  namespace: amazon-cloudwatch
spec:
  template:
    spec:
      containers:
        - name: cloudwatch-agent
# ...
          # Don't change the mountPath
```

```
          volumeMounts:
# ...
            - name: dockersock
              mountPath: /var/run/docker.sock
              readOnly: true
            - name: varlibdocker
              mountPath: /var/lib/docker
              readOnly: true
            - name: containerdsock # NEW mount
              mountPath: /run/containerd/containerd.sock
              readOnly: true
# ...
      volumes:
# ...
        - name: dockersock
          hostPath:
            path: /var/run/docker.sock
        - name: varlibdocker
          hostPath:
            path: /var/lib/docker
        - name: containerdsock # NEW volume
          hostPath:
            path: /run/containerd/containerd.sock
```

No hay métricas de pod cuando se utiliza Bottlerocket para Amazon EKS

Bottlerocket es un sistema operativo de código abierto basado en Linux que está diseñado específicamente por AWS para ejecutar contenedores.

Bottlerocket utiliza una ruta diferente `containerd` en el host, por lo que debe cambiar los volúmenes a la ubicación. De lo contrario, aparece un error en los registros que incluye `W! No pod metric collected`. Consulte el siguiente ejemplo.

```
volumes:
  # ...
    - name: containerdsock
      hostPath:
        # path: /run/containerd/containerd.sock
        # bottlerocket does not mount containerd sock at normal place
        # https://github.com/bottlerocket-os/bottlerocket/
commit/91810c85b83ff4c3660b496e243ef8b55df0973b
        path: /run/dockershim.sock
```

No hay métricas del sistema de archivos de contenedor cuando se utiliza el tiempo de ejecución containerd para Amazon EKS o Kubernetes

Se trata de un problema conocido y los colaboradores de la comunidad lo están tratando. Para obtener más información, consulte Disk usage metric for containerd (Métrica de uso de disco para containerd) y container file system metrics is not supported by cadvisor for containerd (métricas del sistema de archivos de contenedor no son compatibles con cadvisor para containerd) en GitHub.

Aumento inesperado del volumen de registro del agente de CloudWatch al recopilar métricas de Prometheus

Esta fue una regresión que se presentó en la versión 1.247347.6b250880 del agente CloudWatch. Esta regresión ya se ha corregido en versiones más recientes del agente. Su impacto se limitó a situaciones en las que los clientes recopilaban los registros del propio agente de CloudWatch y también utilizaban Prometheus. Para obtener más información, consulte [prometheus] agent is printing all the scraped metrics in log ([prometheus] el agente está imprimiendo todas las métricas raspadas en el registro) en GitHub.

La última imagen de docker mencionada en las notas de la versión que no se encontró en Dockerhub

Hemos actualizado la nota de la versión y la etiqueta en Github antes de comenzar la versión real internamente. Por lo general, toma 1 o 2 semanas ver la última imagen de docker en los registros después de que se actualizó el número de versión en Github. No hay ninguna versión nocturna para la imagen del contenedor del agente de CloudWatch. Puede crear la imagen directamente desde la fuente en la siguiente ubicación: https://github.com/aws/amazon-cloudwatch-agent/tree/main/amazon-cloudwatch-container-insights/cloudwatch-agent-dockerfile

Error de CrashLoopBackoff en el agente de CloudWatch

Si ve un error `CrashLoopBackOff` del agente de CloudWatch, asegúrese de que los permisos de IAM estén establecidos correctamente. Para obtener más información, consulte Verificación de los requisitos previos de .

Agente de CloudWatch o pod FluentD bloqueado en pendiente

Si dispone de un agente de CloudWatch o un pod Fluentd bloqueado en `Pending` o con un error `FailedScheduling`, determine si los nodos tienen suficientes recursos informáticos en función

del número de núcleos y la cantidad de RAM que necesitan los agentes. Especifique el siguiente comando para describir el pod:

```
kubectl describe pod cloudwatch-agent-85ppg -n amazon-cloudwatch
```

Creación de su propia imagen de Docker del agente de CloudWatch

Puede crear su propia imagen de Docker del agente de CloudWatch mediante una referencia al Dockerfile ubicado en https://github.com/aws-samples/amazon-cloudwatch-container-insights/blob/latest/cloudwatch-agent-dockerfile/Dockerfile.

Dockerfile admite la creación de imágenes multiarquitectura directamente con `docker buildx`.

Implementación de otras características del agente de CloudWatch en los contenedores

Puede implementar otras características de supervisión en los contenedores con el agente de CloudWatch. Estas son algunas de ellas:

- Formato de métricas integradas: para obtener más información, consulte Incrustar métricas en los registros.
- StatsD: para obtener más información, consulte Recuperación de las métricas personalizadas con StatsD .

Las instrucciones y los archivos necesarios se encuentran en las siguientes ubicaciones de GitHub:

- En el caso de los contenedores de Amazon ECS, consulte Example Amazon ECS task definitions based on deployment modes (Ejemplo de definiciones de tareas de Amazon ECS basadas en modos de implementación).
- En el caso de los contenedores de Amazon EKS y de Kubernetes, consulte Example Kubernetes YAML files based on deployment modes (Ejemplo de archivos de YAML de Kubernetes basados en modos de implementación).

Lambda Insights

CloudWatch Lambda Insights es una solución de supervisión y solución de problemas para aplicaciones sin servidor que se ejecutan en AWS Lambda. La solución recopila, agrega y resume

las métricas a nivel de sistema, incluido el tiempo de la CPU, la memoria, el disco y el uso de red. También recopila, agrega y resume información de diagnóstico como inicios en frío y paradas de trabajo de Lambda para ayudarle a aislar problemas con las funciones de Lambda y resolverlos rápidamente.

Lambda Insights utiliza una extensión nueva de CloudWatch Lambda que se proporciona como una capa de Lambda. Cuando instale la extensión en una función de Lambda, esta recopila métricas a nivel de sistema y emite un único evento de registro de rendimiento para cada vez que utilice esa función de Lambda. CloudWatch utiliza el formato de métrica incrustado para extraer métricas de los eventos de registro.

Para obtener más información acerca de las extensiones Lambda, consulte Uso de extensiones AWS Lambda. Para obtener más información sobre el formato de métrica integrado, consulte Incrustar métricas en los registros.

Puede utilizar Lambda Insights con cualquier función de Lambda que utilice un tiempo de ejecución de Lambda que admita extensiones de Lambda. Para obtener una lista de los tiempos de ejecución, consulte Lambda Extensions API (Extensiones de la API de Lambda).

Precios

De cada función de Lambda habilitada para Lambda Insights, solo paga por lo que usa para métricas y registros. Para obtener más información sobre los precios, consulte Precios de Amazon CloudWatch.

Se le cobra por el tiempo de ejecución que la extensión de Lambda consumió en incrementos de 1 ms. Para obtener más información acerca de los precios de Lambda, consulte Precios de AWS Lambda.

Introducción a Lambda Insights

Para habilitar Lambda Insights en una función de Lambda, puede utilizar un interruptor de un clic en la consola de Lambda. También puede utilizar AWS CLI, AWS CloudFormation, la CLI de AWS Serverless Application Model o AWS Cloud Development Kit (AWS CDK).

En las siguientes secciones se proporcionan instrucciones detalladas para estos pasos.

Temas

- Versiones disponibles de la extensión de Lambda Insights
- Uso de la consola para habilitar Lambda Insights en una función de Lambda existente

- Uso de AWS CLI para habilitar Lambda Insights en una función de Lambda existente
- Uso de la CLI de AWS SAM para habilitar Lambda Insights en una función de Lambda existente
- Uso de AWS CloudFormation para habilitar Lambda Insights en una función de Lambda existente
- Uso de AWS CDK para habilitar Lambda Insights en una función de Lambda existente
- Uso de Serverless Framework para habilitar Lambda Insights en una función de Lambda existente
- Habilitación de Lambda Insights en una implementación de imágenes de contenedor de Lambda

Versiones disponibles de la extensión de Lambda Insights

En esta sección se enumeran las versiones de la extensión de Lambda Insights y los ARN que se utilizan para estas extensiones en cada Región de AWS.

Temas

- plataformas x86-64
- Plataformas ARM64

plataformas x86-64

En esta sección se enumeran las versiones de la extensión de Lambda Insights para plataformas x84-64 y los ARN que se utilizan para estas extensiones en cada región de AWS.

> **Important**
>
> Las extensiones 1.0.317.0 y posteriores de Lambda Insights no son compatibles con Amazon Linux 1.

1.0.317.0

En la versión 1.0.317.0 se retira la compatibilidad con la plataforma Amazon Linux 1 y se corrigen errores. También se admiten las regiones de AWS GovCloud (US).

ARN para la versión 1.0.317.0

En la siguiente tabla se muestran los ARN que se deben utilizar para esta versión de la extensión en cada Región de AWS en la que está disponible.

Región	ARN
Este de EE. UU. (Norte de Virginia)	`arn:aws:lambda:us-east-1:580247275435:layer:LambdaInsightsExtension:52`
Este de EE. UU. (Ohio)	`arn:aws:lambda:us-east-2:580247275435:layer:LambdaInsightsExtension:52`
Oeste de EE. UU. (Norte de California)	`arn:aws:lambda:us-west-1:580247275435:layer:LambdaInsightsExtension:52`
Oeste de EE. UU. (Oregón)	`arn:aws:lambda:us-west-2:580247275435:layer:LambdaInsightsExtension:52`
África (Ciudad del Cabo)	`arn:aws:lambda:af-south-1:012438385374:layer:LambdaInsightsExtension:43`
Asia-Pacífico (Hong Kong)	`arn:aws:lambda:ap-east-1:519774774795:layer:LambdaInsightsExtension:43`
Asia-Pacífico (Hyderabad)	`arn:aws:lambda:ap-south-2:891564319516:layer:LambdaInsightsExtension:25`
Asia-Pacífico (Yakarta)	`arn:aws:lambda:ap-southeast-3:439286490199:layer:LambdaInsightsExtension:29`
Asia-Pacífico (Melbourne)	`arn:aws:lambda:ap-southeast-4:158895979263:layer:LambdaInsightsExtension:20`
Asia-Pacífico (Bombay)	`arn:aws:lambda:ap-south-1:580247275435:layer:LambdaInsightsExtension:50`
Asia-Pacífico (Osaka)	`arn:aws:lambda:ap-northeast-3:194566237122:layer:LambdaInsightsExtension:33`
Asia-Pacífico (Seúl)	`arn:aws:lambda:ap-northeast-2:580247275435:layer:LambdaInsightsExtension:51`

Región	ARN
Asia-Pacífico (Singapur)	arn:aws:lambda:ap-southeast-1:580247275435:layer:LambdaInsightsExtension:52
Asia-Pacífico (Sídney)	arn:aws:lambda:ap-southeast-2:580247275435:layer:LambdaInsightsExtension:52
Asia-Pacífico (Tokio)	arn:aws:lambda:ap-northeast-1:580247275435:layer:LambdaInsightsExtension:79
Canadá (centro)	arn:aws:lambda:ca-central-1:580247275435:layer:LambdaInsightsExtension:51
Oeste de Canadá (Calgary	arn:aws:lambda:ca-west-1:946466191631:layer:LambdaInsightsExtension:12
China (Pekín)	arn:aws-cn:lambda:cn-north-1:488211338238:layer:LambdaInsightsExtension:42
China (Ningxia);	arn:aws-cn:lambda:cn-northwest-1:488211338238:layer:LambdaInsightsExtension:42
Europa (Fráncfort)	arn:aws:lambda:eu-central-1:580247275435:layer:LambdaInsightsExtension:52
Europa (Irlanda)	arn:aws:lambda:eu-west-1:580247275435:layer:LambdaInsightsExtension:52
Europa (Londres)	arn:aws:lambda:eu-west-2:580247275435:layer:LambdaInsightsExtension:52
Europa (Milán)	arn:aws:lambda:eu-south-1:339249233099:layer:LambdaInsightsExtension:43
Europa (París)	arn:aws:lambda:eu-west-3:580247275435:layer:LambdaInsightsExtension:51
Europa (España)	arn:aws:lambda:eu-south-2:352183217350:layer:LambdaInsightsExtension:27

Región	ARN
Europa (Estocolmo)	`arn:aws:lambda:eu-north-1:580247275435:layer:LambdaInsightsExtension:49`
Europa (Zúrich)	`arn:aws:lambda:eu-central-2:033019950311:layer:LambdaInsightsExtension:26`
Israel (Tel Aviv)	`arn:aws:lambda:il-central-1:459530977127:layer:LambdaInsightsExtension:20`
Medio Oriente (Baréin)	`arn:aws:lambda:me-south-1:285320876703:layer:LambdaInsightsExtension:43`
Medio Oriente (EAU)	`arn:aws:lambda:me-central-1:732604637566:layer:LambdaInsightsExtension:26`
América del Sur (São Paulo)	`arn:aws:lambda:sa-east-1:580247275435:layer:LambdaInsightsExtension:51`
AWS GovCloud (Este de EE. UU.)	`arn:aws-us-gov:lambda:us-gov-east-1:122132214140:layer:LambdaInsightsExtension:19`
AWS GovCloud (Oeste de EE. UU.)	`arn:aws-us-gov:lambda:us-gov-west-1:751350123760:layer:LambdaInsightsExtension:19`

1.0.295.0

La versión 1.0.295.0 incluye actualizaciones de dependencias para todos los tiempos de ejecución compatibles.

ARN para la versión 1.0.295.0

En la siguiente tabla se muestran los ARN que se deben utilizar para esta versión de la extensión en cada Región de AWS en la que está disponible.

Región	ARN
Este de EE. UU. (Norte de Virginia)	arn:aws:lambda:us-east-1:580247275435:layer:LambdaInsightsExtension:51
Este de EE. UU. (Ohio)	arn:aws:lambda:us-east-2:580247275435:layer:LambdaInsightsExtension:51
Oeste de EE. UU. (Norte de California)	arn:aws:lambda:us-west-1:580247275435:layer:LambdaInsightsExtension:51
Oeste de EE. UU. (Oregón)	arn:aws:lambda:us-west-2:580247275435:layer:LambdaInsightsExtension:51
África (Ciudad del Cabo)	arn:aws:lambda:af-south-1:012438385374:layer:LambdaInsightsExtension:42
Asia-Pacífico (Hong Kong)	arn:aws:lambda:ap-east-1:519774774795:layer:LambdaInsightsExtension:42
Asia-Pacífico (Hyderabad)	arn:aws:lambda:ap-south-2:891564319516:layer:LambdaInsightsExtension:24
Asia-Pacífico (Yakarta)	arn:aws:lambda:ap-southeast-3:439286490199:layer:LambdaInsightsExtension:28
Asia-Pacífico (Melbourne)	arn:aws:lambda:ap-southeast-4:158895979263:layer:LambdaInsightsExtension:19
Asia-Pacífico (Bombay)	arn:aws:lambda:ap-south-1:580247275435:layer:LambdaInsightsExtension:49
Asia-Pacífico (Osaka)	arn:aws:lambda:ap-northeast-3:194566237122:layer:LambdaInsightsExtension:32
Asia-Pacífico (Seúl)	arn:aws:lambda:ap-northeast-2:580247275435:layer:LambdaInsightsExtension:50

Región	ARN
Asia-Pacífico (Singapur)	arn:aws:lambda:ap-southeast-1:580247275435:layer:LambdaInsightsExtension:51
Asia-Pacífico (Sídney)	arn:aws:lambda:ap-southeast-2:580247275435:layer:LambdaInsightsExtension:51
Asia-Pacífico (Tokio)	arn:aws:lambda:ap-northeast-1:580247275435:layer:LambdaInsightsExtension:78
Canadá (centro)	arn:aws:lambda:ca-central-1:580247275435:layer:LambdaInsightsExtension:50
Oeste de Canadá (Calgary	arn:aws:lambda:ca-west-1:946466191631:layer:LambdaInsightsExtension:11
China (Pekín)	arn:aws-cn:lambda:cn-north-1:488211338238:layer:LambdaInsightsExtension:41
China (Ningxia);	arn:aws-cn:lambda:cn-northwest-1:488211338238:layer:LambdaInsightsExtension:41
Europa (Fráncfort)	arn:aws:lambda:eu-central-1:580247275435:layer:LambdaInsightsExtension:51
Europa (Irlanda)	arn:aws:lambda:eu-west-1:580247275435:layer:LambdaInsightsExtension:51
Europa (Londres)	arn:aws:lambda:eu-west-2:580247275435:layer:LambdaInsightsExtension:51
Europa (Milán)	arn:aws:lambda:eu-south-1:339249233099:layer:LambdaInsightsExtension:42
Europa (París)	arn:aws:lambda:eu-west-3:580247275435:layer:LambdaInsightsExtension:50
Europa (España)	arn:aws:lambda:eu-south-2:352183217350:layer:LambdaInsightsExtension:26

Región	ARN
Europa (Estocolmo)	`arn:aws:lambda:eu-north-1:580247275435:layer:LambdaInsightsExtension:48`
Europa (Zúrich)	`arn:aws:lambda:eu-central-2:033019950311:layer:LambdaInsightsExtension:25`
Israel (Tel Aviv)	`arn:aws:lambda:il-central-1:459530977127:layer:LambdaInsightsExtension:19`
Medio Oriente (Baréin)	`arn:aws:lambda:me-south-1:285320876703:layer:LambdaInsightsExtension:42`
Medio Oriente (EAU)	`arn:aws:lambda:me-central-1:732604637566:layer:LambdaInsightsExtension:25`
América del Sur (São Paulo)	`arn:aws:lambda:sa-east-1:580247275435:layer:LambdaInsightsExtension:50`

1.0.275.0

La versión 1.0.275.0 incluye actualizaciones de dependencias importantes para todos los tiempos de ejecución compatibles.

ARN para la versión 1.0.275.0

En la siguiente tabla se muestran los ARN que se deben utilizar para esta versión de la extensión en cada Región de AWS en la que está disponible.

Región	ARN
Este de EE. UU. (Norte de Virginia)	`arn:aws:lambda:us-east-1:580247275435:layer:LambdaInsightsExtension:49`
Este de EE. UU. (Ohio)	`arn:aws:lambda:us-east-2:580247275435:layer:LambdaInsightsExtension:49`

Región	ARN
Oeste de EE. UU. (Norte de California)	arn:aws:lambda:us-west-1:580247275435:layer:LambdaInsightsExtension:49
Oeste de EE. UU. (Oregón)	arn:aws:lambda:us-west-2:580247275435:layer:LambdaInsightsExtension:49
África (Ciudad del Cabo)	arn:aws:lambda:af-south-1:012438385374:layer:LambdaInsightsExtension:40
Asia-Pacífico (Hong Kong)	arn:aws:lambda:ap-east-1:519774774795:layer:LambdaInsightsExtension:40
Asia-Pacífico (Hyderabad)	arn:aws:lambda:ap-south-2:891564319516:layer:LambdaInsightsExtension:22
Asia-Pacífico (Yakarta)	arn:aws:lambda:ap-southeast-3:439286490199:layer:LambdaInsightsExtension:26
Asia-Pacífico (Melbourne)	arn:aws:lambda:ap-southeast-4:158895979263:layer:LambdaInsightsExtension:17
Asia-Pacífico (Bombay)	arn:aws:lambda:ap-south-1:580247275435:layer:LambdaInsightsExtension:47
Asia-Pacífico (Osaka)	arn:aws:lambda:ap-northeast-3:194566237122:layer:LambdaInsightsExtension:30
Asia-Pacífico (Seúl)	arn:aws:lambda:ap-northeast-2:580247275435:layer:LambdaInsightsExtension:48
Asia-Pacífico (Singapur)	arn:aws:lambda:ap-southeast-1:580247275435:layer:LambdaInsightsExtension:49
Asia-Pacífico (Sídney)	arn:aws:lambda:ap-southeast-2:580247275435:layer:LambdaInsightsExtension:49
Asia-Pacífico (Tokio)	arn:aws:lambda:ap-northeast-1:580247275435:layer:LambdaInsightsExtension:76

Región	ARN
Canadá (centro)	arn:aws:lambda:ca-central-1:580247275435:layer:LambdaInsightsExtension:48
Oeste de Canadá (Calgary	arn:aws:lambda:ca-west-1:946466191631:layer:LambdaInsightsExtension:9
China (Pekín)	arn:aws-cn:lambda:cn-north-1:488211338238:layer:LambdaInsightsExtension:39
China (Ningxia);	arn:aws-cn:lambda:cn-northwest-1:488211338238:layer:LambdaInsightsExtension:39
Europa (Fráncfort)	arn:aws:lambda:eu-central-1:580247275435:layer:LambdaInsightsExtension:49
Europa (Irlanda)	arn:aws:lambda:eu-west-1:580247275435:layer:LambdaInsightsExtension:49
Europa (Londres)	arn:aws:lambda:eu-west-2:580247275435:layer:LambdaInsightsExtension:49
Europa (Milán)	arn:aws:lambda:eu-south-1:339249233099:layer:LambdaInsightsExtension:40
Europa (París)	arn:aws:lambda:eu-west-3:580247275435:layer:LambdaInsightsExtension:48
Europa (España)	arn:aws:lambda:eu-south-2:352183217350:layer:LambdaInsightsExtension:24
Europa (Estocolmo)	arn:aws:lambda:eu-north-1:580247275435:layer:LambdaInsightsExtension:46
Europa (Zúrich)	arn:aws:lambda:eu-central-2:033019950311:layer:LambdaInsightsExtension:23
Israel (Tel Aviv)	arn:aws:lambda:il-central-1:459530977127:layer:LambdaInsightsExtension:17

Región	ARN
Medio Oriente (Baréin)	`arn:aws:lambda:me-south-1:285320876703:layer:LambdaInsightsExtension:40`
Medio Oriente (EAU)	`arn:aws:lambda:me-central-1:732604637566:layer:LambdaInsightsExtension:23`
América del Sur (São Paulo)	`arn:aws:lambda:sa-east-1:580247275435:layer:LambdaInsightsExtension:48`

1.0.273.0

La versión 1.0.273.0 incluye importantes correcciones de errores para todos los tiempos de ejecución compatibles y agrega compatibilidad con la región Oeste de Canadá (Calgary).

ARN para la versión 1.0.273.0

En la siguiente tabla se muestran los ARN que se deben utilizar para esta versión de la extensión en cada Región de AWS en la que está disponible.

Región	ARN
Este de EE. UU. (Norte de Virginia)	`arn:aws:lambda:us-east-1:580247275435:layer:LambdaInsightsExtension:45`
Este de EE. UU. (Ohio)	`arn:aws:lambda:us-east-2:580247275435:layer:LambdaInsightsExtension:45`
Oeste de EE. UU. (Norte de California)	`arn:aws:lambda:us-west-1:580247275435:layer:LambdaInsightsExtension:45`
Oeste de EE. UU. (Oregón)	`arn:aws:lambda:us-west-2:580247275435:layer:LambdaInsightsExtension:45`
África (Ciudad del Cabo)	`arn:aws:lambda:af-south-1:012438385374:layer:LambdaInsightsExtension:35`

Región	ARN
Asia-Pacífico (Hong Kong)	arn:aws:lambda:ap-east-1:519774774795:layer:LambdaInsightsExtension:35
Asia-Pacífico (Hyderabad)	arn:aws:lambda:ap-south-2:891564319516:layer:LambdaInsightsExtension:17
Asia-Pacífico (Yakarta)	arn:aws:lambda:ap-southeast-3:439286490199:layer:LambdaInsightsExtension:21
Asia-Pacífico (Melbourne)	arn:aws:lambda:ap-southeast-4:158895979263:layer:LambdaInsightsExtension:12
Asia-Pacífico (Bombay)	arn:aws:lambda:ap-south-1:580247275435:layer:LambdaInsightsExtension:43
Asia-Pacífico (Osaka)	arn:aws:lambda:ap-northeast-3:194566237122:layer:LambdaInsightsExtension:26
Asia-Pacífico (Seúl)	arn:aws:lambda:ap-northeast-2:580247275435:layer:LambdaInsightsExtension:44
Asia-Pacífico (Singapur)	arn:aws:lambda:ap-southeast-1:580247275435:layer:LambdaInsightsExtension:45
Asia-Pacífico (Sídney)	arn:aws:lambda:ap-southeast-2:580247275435:layer:LambdaInsightsExtension:45
Asia-Pacífico (Tokio)	arn:aws:lambda:ap-northeast-1:580247275435:layer:LambdaInsightsExtension:72
Canadá (centro)	arn:aws:lambda:ca-central-1:580247275435:layer:LambdaInsightsExtension:44
Oeste de Canadá (Calgary	arn:aws:lambda:ca-west-1:946466191631:layer:LambdaInsightsExtension:4
China (Pekín)	arn:aws-cn:lambda:cn-north-1:488211338238:layer:LambdaInsightsExtension:36

Región	ARN
China (Ningxia);	arn:aws-cn:lambda:cn-northwest-1:488211338238:layer:LambdaInsightsExtension:36
Europa (Fráncfort)	arn:aws:lambda:eu-central-1:580247275435:layer:LambdaInsightsExtension:45
Europa (Irlanda)	arn:aws:lambda:eu-west-1:580247275435:layer:LambdaInsightsExtension:45
Europa (Londres)	arn:aws:lambda:eu-west-2:580247275435:layer:LambdaInsightsExtension:45
Europa (Milán)	arn:aws:lambda:eu-south-1:339249233099:layer:LambdaInsightsExtension:35
Europa (París)	arn:aws:lambda:eu-west-3:580247275435:layer:LambdaInsightsExtension:44
Europa (España)	arn:aws:lambda:eu-south-2:352183217350:layer:LambdaInsightsExtension:19
Europa (Estocolmo)	arn:aws:lambda:eu-north-1:580247275435:layer:LambdaInsightsExtension:42
Europa (Zúrich)	arn:aws:lambda:eu-central-2:033019950311:layer:LambdaInsightsExtension:17
Israel (Tel Aviv)	arn:aws:lambda:il-central-1:459530977127:layer:LambdaInsightsExtension:12
Medio Oriente (Baréin)	arn:aws:lambda:me-south-1:285320876703:layer:LambdaInsightsExtension:35
Medio Oriente (EAU)	arn:aws:lambda:me-central-1:732604637566:layer:LambdaInsightsExtension:18
América del Sur (São Paulo)	arn:aws:lambda:sa-east-1:580247275435:layer:LambdaInsightsExtension:44

1.0.229.0

La versión 1.0.229.0 incluye importantes correcciones de errores para todos los tiempos de ejecución compatibles y añade compatibilidad con la región de Israel (Tel Aviv).

ARN para la versión 1.0.229.0

En la siguiente tabla se muestran los ARN que se deben utilizar para esta versión de la extensión en cada Región de AWS en la que está disponible.

Región	ARN
Este de EE. UU. (Norte de Virginia)	`arn:aws:lambda:us-east-1:580247275435:layer:LambdaInsightsExtension:38`
Este de EE. UU. (Ohio)	`arn:aws:lambda:us-east-2:580247275435:layer:LambdaInsightsExtension:38`
Oeste de EE. UU. (Norte de California)	`arn:aws:lambda:us-west-1:580247275435:layer:LambdaInsightsExtension:38`
Oeste de EE. UU. (Oregón)	`arn:aws:lambda:us-west-2:580247275435:layer:LambdaInsightsExtension:38`
África (Ciudad del Cabo)	`arn:aws:lambda:af-south-1:012438385374:layer:LambdaInsightsExtension:28`
Asia-Pacífico (Hong Kong)	`arn:aws:lambda:ap-east-1:519774774795:layer:LambdaInsightsExtension:28`
Asia-Pacífico (Hyderabad)	`arn:aws:lambda:ap-south-2:891564319516:layer:LambdaInsightsExtension:10`
Asia-Pacífico (Yakarta)	`arn:aws:lambda:ap-southeast-3:439286490199:layer:LambdaInsightsExtension:14`
Asia-Pacífico (Melbourne)	`arn:aws:lambda:ap-southeast-4:158895979263:layer:LambdaInsightsExtension:5`

Región	ARN
Asia-Pacífico (Bombay)	arn:aws:lambda:ap-south-1:580247275435:layer:LambdaInsightsExtension:36
Asia-Pacífico (Osaka)	arn:aws:lambda:ap-northeast-3:194566237122:layer:LambdaInsightsExtension:19
Asia-Pacífico (Seúl)	arn:aws:lambda:ap-northeast-2:580247275435:layer:LambdaInsightsExtension:37
Asia-Pacífico (Singapur)	arn:aws:lambda:ap-southeast-1:580247275435:layer:LambdaInsightsExtension:38
Asia-Pacífico (Sídney)	arn:aws:lambda:ap-southeast-2:580247275435:layer:LambdaInsightsExtension:38
Asia-Pacífico (Tokio)	arn:aws:lambda:ap-northeast-1:580247275435:layer:LambdaInsightsExtension:60
Canadá (centro)	arn:aws:lambda:ca-central-1:580247275435:layer:LambdaInsightsExtension:37
China (Pekín)	arn:aws-cn:lambda:cn-north-1:488211338238:layer:LambdaInsightsExtension:29
China (Ningxia);	arn:aws-cn:lambda:cn-northwest-1:488211338238:layer:LambdaInsightsExtension:29
Europa (Fráncfort)	arn:aws:lambda:eu-central-1:580247275435:layer:LambdaInsightsExtension:38
Europa (Irlanda)	arn:aws:lambda:eu-west-1:580247275435:layer:LambdaInsightsExtension:38
Europa (Londres)	arn:aws:lambda:eu-west-2:580247275435:layer:LambdaInsightsExtension:38
Europa (Milán)	arn:aws:lambda:eu-south-1:339249233099:layer:LambdaInsightsExtension:28

Región	ARN
Europa (París)	arn:aws:lambda:eu-west-3:580247275435:layer:LambdaInsightsExtension:37
Europa (España)	arn:aws:lambda:eu-south-2:352183217350:layer:LambdaInsightsExtension:12
Europa (Estocolmo)	arn:aws:lambda:eu-north-1:580247275435:layer:LambdaInsightsExtension:35
Europa (Zúrich)	arn:aws:lambda:eu-central-2:033019950311:layer:LambdaInsightsExtension:11
Israel (Tel Aviv)	arn:aws:lambda:il-central-1:459530977127:layer:LambdaInsightsExtension:5
Medio Oriente (Baréin)	arn:aws:lambda:me-south-1:285320876703:layer:LambdaInsightsExtension:28
Medio Oriente (EAU)	arn:aws:lambda:me-central-1:732604637566:layer:LambdaInsightsExtension:11
América del Sur (São Paulo)	arn:aws:lambda:sa-east-1:580247275435:layer:LambdaInsightsExtension:37

1.0.178.0

La versión 1.0.178.0 agrega compatibilidad con las siguientes regiones de AWS.

- Asia-Pacífico (Hyderabad)
- Asia-Pacífico (Yakarta)
- Europa (España)
- Europa (Zúrich)
- Medio Oriente (EAU)

ARN para la versión 1.0.178.0

En la siguiente tabla se muestran los ARN que se deben utilizar para esta versión de la extensión en cada Región de AWS en la que está disponible.

Región	ARN
Este de EE. UU. (Norte de Virginia)	`arn:aws:lambda:us-east-1:580247275435:layer:LambdaInsightsExtension:35`
Este de EE. UU. (Ohio)	`arn:aws:lambda:us-east-2:580247275435:layer:LambdaInsightsExtension:33`
Oeste de EE. UU. (Norte de California)	`arn:aws:lambda:us-west-1:580247275435:layer:LambdaInsightsExtension:33`
Oeste de EE. UU. (Oregón)	`arn:aws:lambda:us-west-2:580247275435:layer:LambdaInsightsExtension:33`
África (Ciudad del Cabo)	`arn:aws:lambda:af-south-1:012438385374:layer:LambdaInsightsExtension:25`
Asia-Pacífico (Hong Kong)	`arn:aws:lambda:ap-east-1:519774774795:layer:LambdaInsightsExtension:25`
Asia-Pacífico (Hyderabad)	`arn:aws:lambda:ap-south-2:891564319516:layer:LambdaInsightsExtension:8`
Asia-Pacífico (Yakarta)	`arn:aws:lambda:ap-southeast-3:439286490199:layer:LambdaInsightsExtension:11`
Asia-Pacífico (Bombay)	`arn:aws:lambda:ap-south-1:580247275435:layer:LambdaInsightsExtension:31`
Asia-Pacífico (Osaka)	`arn:aws:lambda:ap-northeast-3:194566237122:layer:LambdaInsightsExtension:2`
Asia-Pacífico (Seúl)	`arn:aws:lambda:ap-northeast-2:580247275435:layer:LambdaInsightsExtension:32`

Región	ARN
Asia-Pacífico (Singapur)	arn:aws:lambda:ap-southeast-1:580247275435:layer:LambdaInsightsExtension:33
Asia-Pacífico (Sídney)	arn:aws:lambda:ap-southeast-2:580247275435:layer:LambdaInsightsExtension:33
Asia-Pacífico (Tokio)	arn:aws:lambda:ap-northeast-1:580247275435:layer:LambdaInsightsExtension:50
Canadá (centro)	arn:aws:lambda:ca-central-1:580247275435:layer:LambdaInsightsExtension:32
China (Pekín)	arn:aws-cn:lambda:cn-north-1:488211338238:layer:LambdaInsightsExtension:26
China (Ningxia);	arn:aws-cn:lambda:cn-northwest-1:488211338238:layer:LambdaInsightsExtension:26
Europa (Fráncfort)	arn:aws:lambda:eu-central-1:580247275435:layer:LambdaInsightsExtension:35
Europa (Irlanda)	arn:aws:lambda:eu-west-1:580247275435:layer:LambdaInsightsExtension:33
Europa (Londres)	arn:aws:lambda:eu-west-2:580247275435:layer:LambdaInsightsExtension:33
Europa (Milán)	arn:aws:lambda:eu-south-1:339249233099:layer:LambdaInsightsExtension:25
Europa (París)	arn:aws:lambda:eu-west-3:580247275435:layer:LambdaInsightsExtension:32
Europa (España)	arn:aws:lambda:eu-south-2:352183217350:layer:LambdaInsightsExtension:10
Europa (Estocolmo)	arn:aws:lambda:eu-north-1:580247275435:layer:LambdaInsightsExtension:30

Región	ARN
Europa (Zúrich)	`arn:aws:lambda:eu-central-2:033019950311:layer:LambdaInsightsExtension:7`
Medio Oriente (Baréin)	`arn:aws:lambda:me-south-1:285320876703:layer:LambdaInsightsExtension:25`
Medio Oriente (EAU)	`arn:aws:lambda:me-central-1:732604637566:layer:LambdaInsightsExtension:9`
América del Sur (São Paulo)	`arn:aws:lambda:sa-east-1:580247275435:layer:LambdaInsightsExtension:32`

1.0.143.0

La versión 1.0.143.0 incluye correcciones de errores de compatibilidad con Python 3.7 y Go 1.x. El tiempo de ejecución de Python 3.6 Lambda está en desuso. Para obtener más información, consulte Tiempos de ejecución de Lambda.

ARN para la versión 1.0.143.0

En la siguiente tabla se muestran los ARN que se deben utilizar para esta versión de la extensión en cada Región de AWS en la que está disponible.

Región	ARN
Este de EE. UU. (Norte de Virginia)	`arn:aws:lambda:us-east-1:580247275435:layer:LambdaInsightsExtension:21`
Este de EE. UU. (Ohio)	`arn:aws:lambda:us-east-2:580247275435:layer:LambdaInsightsExtension:21`
Oeste de EE. UU. (Norte de California)	`arn:aws:lambda:us-west-1:580247275435:layer:LambdaInsightsExtension:20`
Oeste de EE. UU. (Oregón)	`arn:aws:lambda:us-west-2:580247275435:layer:LambdaInsightsExtension:21`

Región	ARN
África (Ciudad del Cabo)	arn:aws:lambda:af-south-1:012438385374:layer:LambdaInsightsExtension:13
Asia-Pacífico (Hong Kong)	arn:aws:lambda:ap-east-1:519774774795:layer:LambdaInsightsExtension:13
Asia-Pacífico (Bombay)	arn:aws:lambda:ap-south-1:580247275435:layer:LambdaInsightsExtension:21
Asia-Pacífico (Osaka)	arn:aws:lambda:ap-northeast-3:194566237122:layer:LambdaInsightsExtension:2
Asia-Pacífico (Seúl)	arn:aws:lambda:ap-northeast-2:580247275435:layer:LambdaInsightsExtension:20
Asia-Pacífico (Singapur)	arn:aws:lambda:ap-southeast-1:580247275435:layer:LambdaInsightsExtension:21
Asia-Pacífico (Sídney)	arn:aws:lambda:ap-southeast-2:580247275435:layer:LambdaInsightsExtension:21
Asia-Pacífico (Tokio)	arn:aws:lambda:ap-northeast-1:580247275435:layer:LambdaInsightsExtension:32
Canadá (centro)	arn:aws:lambda:ca-central-1:580247275435:layer:LambdaInsightsExtension:20
China (Pekín)	arn:aws-cn:lambda:cn-north-1:488211338238:layer:LambdaInsightsExtension:14
China (Ningxia);	arn:aws-cn:lambda:cn-northwest-1:488211338238:layer:LambdaInsightsExtension:14
Europa (Fráncfort)	arn:aws:lambda:eu-central-1:580247275435:layer:LambdaInsightsExtension:21
Europa (Irlanda)	arn:aws:lambda:eu-west-1:580247275435:layer:LambdaInsightsExtension:21

Región	ARN
Europa (Londres)	`arn:aws:lambda:eu-west-2:580247275435:layer:LambdaInsightsExtension:21`
Europa (Milán)	`arn:aws:lambda:eu-south-1:339249233099:layer:LambdaInsightsExtension:13`
Europa (París)	`arn:aws:lambda:eu-west-3:580247275435:layer:LambdaInsightsExtension:20`
Europa (Estocolmo)	`arn:aws:lambda:eu-north-1:580247275435:layer:LambdaInsightsExtension:20`
Medio Oriente (Baréin)	`arn:aws:lambda:me-south-1:285320876703:layer:LambdaInsightsExtension:13`
América del Sur (São Paulo)	`arn:aws:lambda:sa-east-1:580247275435:layer:LambdaInsightsExtension:20`

1.0.135.0

La versión 1.0.135.0 incluye correcciones de errores sobre la forma en que Lambda Insights recopila e informa datos sobre el uso del descriptor de discos y archivos. En versiones anteriores de la extensión, la métrica `tmp_free` informó el espacio libre máximo en el directorio `/tmp` mientras se ejecuta una función. Esta versión cambia la métrica para que informe, en cambio, el valor mínimo, lo que la hace más útil a la hora de evaluar el uso del disco. Para obtener más información sobre las cuotas de almacenamiento del directorio `tmp`, consulte Cuotas de Lambda.

La versión 1.0.135.0 también informa ahora el uso del descriptor de archivos (`fd_use` y `fd_max`) como el valor máximo en todos los procesos, en lugar de informar el nivel del sistema operativo.

ARN para la versión 1.0.135.0

En la siguiente tabla se muestran los ARN que se deben utilizar para esta versión de la extensión en cada Región de AWS en la que está disponible.

Región	ARN
Este de EE. UU. (Norte de Virginia)	`arn:aws:lambda:us-east-1:580247275435:layer:LambdaInsightsExtension:18`
Este de EE. UU. (Ohio)	`arn:aws:lambda:us-east-2:580247275435:layer:LambdaInsightsExtension:18`
Oeste de EE. UU. (Norte de California)	`arn:aws:lambda:us-west-1:580247275435:layer:LambdaInsightsExtension:18`
Oeste de EE. UU. (Oregón)	`arn:aws:lambda:us-west-2:580247275435:layer:LambdaInsightsExtension:18`
África (Ciudad del Cabo)	`arn:aws:lambda:af-south-1:012438385374:layer:LambdaInsightsExtension:11`
Asia-Pacífico (Hong Kong)	`arn:aws:lambda:ap-east-1:519774774795:layer:LambdaInsightsExtension:11`
Asia-Pacífico (Bombay)	`arn:aws:lambda:ap-south-1:580247275435:layer:LambdaInsightsExtension:18`
Asia-Pacífico (Osaka)	`arn:aws:lambda:ap-northeast-3:194566237122:layer:LambdaInsightsExtension:1`
Asia-Pacífico (Seúl)	`arn:aws:lambda:ap-northeast-2:580247275435:layer:LambdaInsightsExtension:18`
Asia-Pacífico (Singapur)	`arn:aws:lambda:ap-southeast-1:580247275435:layer:LambdaInsightsExtension:18`
Asia-Pacífico (Sídney)	`arn:aws:lambda:ap-southeast-2:580247275435:layer:LambdaInsightsExtension:18`
Asia-Pacífico (Tokio)	`arn:aws:lambda:ap-northeast-1:580247275435:layer:LambdaInsightsExtension:25`

Región	ARN
Canadá (centro)	arn:aws:lambda:ca-central-1:580247275435:layer:LambdaInsightsExtension:18
China (Pekín)	arn:aws-cn:lambda:cn-north-1:488211338238:layer:LambdaInsightsExtension:11
China (Ningxia);	arn:aws-cn:lambda:cn-northwest-1:488211338238:layer:LambdaInsightsExtension:11
Europa (Fráncfort)	arn:aws:lambda:eu-central-1:580247275435:layer:LambdaInsightsExtension:18
Europa (Irlanda)	arn:aws:lambda:eu-west-1:580247275435:layer:LambdaInsightsExtension:18
Europa (Londres)	arn:aws:lambda:eu-west-2:580247275435:layer:LambdaInsightsExtension:18
Europa (Milán)	arn:aws:lambda:eu-south-1:339249233099:layer:LambdaInsightsExtension:11
Europa (París)	arn:aws:lambda:eu-west-3:580247275435:layer:LambdaInsightsExtension:18
Europa (Estocolmo)	arn:aws:lambda:eu-north-1:580247275435:layer:LambdaInsightsExtension:18
Medio Oriente (Baréin)	arn:aws:lambda:me-south-1:285320876703:layer:LambdaInsightsExtension:11
América del Sur (São Paulo)	arn:aws:lambda:sa-east-1:580247275435:layer:LambdaInsightsExtension:18

1.0.119.0

ARN para la versión 1.0.119.0

En la siguiente tabla se muestran los ARN que se deben utilizar para esta versión de la extensión en cada Región de AWS en la que está disponible.

Región	ARN
Este de EE. UU. (Norte de Virginia)	`arn:aws:lambda:us-east-1:580247275435:layer:LambdaInsightsExtension:16`
Este de EE. UU. (Ohio)	`arn:aws:lambda:us-east-2:580247275435:layer:LambdaInsightsExtension:16`
Oeste de EE. UU. (Norte de California)	`arn:aws:lambda:us-west-1:580247275435:layer:LambdaInsightsExtension:16`
Oeste de EE. UU. (Oregón)	`arn:aws:lambda:us-west-2:580247275435:layer:LambdaInsightsExtension:16`
África (Ciudad del Cabo)	`arn:aws:lambda:af-south-1:012438385374:layer:LambdaInsightsExtension:9`
Asia-Pacífico (Hong Kong)	`arn:aws:lambda:ap-east-1:519774774795:layer:LambdaInsightsExtension:9`
Asia-Pacífico (Bombay)	`arn:aws:lambda:ap-south-1:580247275435:layer:LambdaInsightsExtension:16`
Asia-Pacífico (Seúl)	`arn:aws:lambda:ap-northeast-2:580247275435:layer:LambdaInsightsExtension:16`
Asia-Pacífico (Singapur)	`arn:aws:lambda:ap-southeast-1:580247275435:layer:LambdaInsightsExtension:16`
Asia-Pacífico (Sídney)	`arn:aws:lambda:ap-southeast-2:580247275435:layer:LambdaInsightsExtension:16`
Asia-Pacífico (Tokio)	`arn:aws:lambda:ap-northeast-1:580247275435:layer:LambdaInsightsExtension:23`

Región	ARN
Canadá (centro)	arn:aws:lambda:ca-central-1:580247275435:layer:LambdaInsightsExtension:16
China (Pekín)	arn:aws-cn:lambda:cn-north-1:488211338238:layer:LambdaInsightsExtension:9
China (Ningxia);	arn:aws-cn:lambda:cn-northwest-1:488211338238:layer:LambdaInsightsExtension:9
Europa (Fráncfort)	arn:aws:lambda:eu-central-1:580247275435:layer:LambdaInsightsExtension:16
Europa (Irlanda)	arn:aws:lambda:eu-west-1:580247275435:layer:LambdaInsightsExtension:16
Europa (Londres)	arn:aws:lambda:eu-west-2:580247275435:layer:LambdaInsightsExtension:16
Europa (Milán)	arn:aws:lambda:eu-south-1:339249233099:layer:LambdaInsightsExtension:9
Europa (París)	arn:aws:lambda:eu-west-3:580247275435:layer:LambdaInsightsExtension:16
Europa (Estocolmo)	arn:aws:lambda:eu-north-1:580247275435:layer:LambdaInsightsExtension:16
Medio Oriente (Baréin)	arn:aws:lambda:me-south-1:285320876703:layer:LambdaInsightsExtension:9
América del Sur (São Paulo)	arn:aws:lambda:sa-east-1:580247275435:layer:LambdaInsightsExtension:16

1.0.98.0

En esta versión se elimina el registro innecesario y también se soluciona un problema con las invocaciones locales de la CLI de AWS Serverless Application Model. Para obtener más información

acerca de este problema, consulte Adding results in timeout with 'sam local invoke' (La adición de LambdaInsightsExtension da como resultado tiempo de espera con 'invocación local sam').

ARN para la versión 1.0.98.0

En la siguiente tabla se muestran los ARN que se deben utilizar para esta versión de la extensión en cada Región de AWS en la que está disponible.

Región	ARN
Este de EE. UU. (Norte de Virginia)	`arn:aws:lambda:us-east-1:580247275435:layer:LambdaInsightsExtension:14`
Este de EE. UU. (Ohio)	`arn:aws:lambda:us-east-2:580247275435:layer:LambdaInsightsExtension:14`
Oeste de EE. UU. (Norte de California)	`arn:aws:lambda:us-west-1:580247275435:layer:LambdaInsightsExtension:14`
Oeste de EE. UU. (Oregón)	`arn:aws:lambda:us-west-2:580247275435:layer:LambdaInsightsExtension:14`
África (Ciudad del Cabo)	`arn:aws:lambda:af-south-1:012438385374:layer:LambdaInsightsExtension:8`
Asia-Pacífico (Hong Kong)	`arn:aws:lambda:ap-east-1:519774774795:layer:LambdaInsightsExtension:8`
Asia-Pacífico (Bombay)	`arn:aws:lambda:ap-south-1:580247275435:layer:LambdaInsightsExtension:14`
Asia-Pacífico (Seúl)	`arn:aws:lambda:ap-northeast-2:580247275435:layer:LambdaInsightsExtension:14`
Asia-Pacífico (Singapur)	`arn:aws:lambda:ap-southeast-1:580247275435:layer:LambdaInsightsExtension:14`
Asia-Pacífico (Sídney)	`arn:aws:lambda:ap-southeast-2:580247275435:layer:LambdaInsightsExtension:14`

Región	ARN
Asia-Pacífico (Tokio)	`arn:aws:lambda:ap-northeast-1:580247275435:layer:LambdaInsightsExtension:14`
Canadá (centro)	`arn:aws:lambda:ca-central-1:580247275435:layer:LambdaInsightsExtension:14`
China (Pekín)	`arn:aws-cn:lambda:cn-north-1:488211338238:layer:LambdaInsightsExtension:8`
China (Ningxia);	`arn:aws-cn:lambda:cn-northwest-1:488211338238:layer:LambdaInsightsExtension:8`
Europa (Fráncfort)	`arn:aws:lambda:eu-central-1:580247275435:layer:LambdaInsightsExtension:14`
Europa (Irlanda)	`arn:aws:lambda:eu-west-1:580247275435:layer:LambdaInsightsExtension:14`
Europa (Londres)	`arn:aws:lambda:eu-west-2:580247275435:layer:LambdaInsightsExtension:14`
Europa (Milán)	`arn:aws:lambda:eu-south-1:339249233099:layer:LambdaInsightsExtension:8`
Europa (París)	`arn:aws:lambda:eu-west-3:580247275435:layer:LambdaInsightsExtension:14`
Europa (Estocolmo)	`arn:aws:lambda:eu-north-1:580247275435:layer:LambdaInsightsExtension:14`
Medio Oriente (Baréin)	`arn:aws:lambda:me-south-1:285320876703:layer:LambdaInsightsExtension:8`
América del Sur (São Paulo)	`arn:aws:lambda:sa-east-1:580247275435:layer:LambdaInsightsExtension:14`

1.0.89.0

Esta versión corrige la marca de tiempo del evento de rendimiento para representar siempre el comienzo de la invocación de la función.

ARN para la versión 1.0.89.0

En la siguiente tabla se muestran los ARN que se deben utilizar para esta versión de la extensión en cada Región de AWS en la que está disponible.

Región	ARN
Este de EE. UU. (Norte de Virginia)	`arn:aws:lambda:us-east-1:580247275435:layer:LambdaInsightsExtension:12`
Este de EE. UU. (Ohio)	`arn:aws:lambda:us-east-2:580247275435:layer:LambdaInsightsExtension:12`
Oeste de EE. UU. (Norte de California)	`arn:aws:lambda:us-west-1:580247275435:layer:LambdaInsightsExtension:12`
Oeste de EE. UU. (Oregón)	`arn:aws:lambda:us-west-2:580247275435:layer:LambdaInsightsExtension:12`
Asia-Pacífico (Bombay)	`arn:aws:lambda:ap-south-1:580247275435:layer:LambdaInsightsExtension:12`
Asia-Pacífico (Seúl)	`arn:aws:lambda:ap-northeast-2:580247275435:layer:LambdaInsightsExtension:12`
Asia-Pacífico (Singapur)	`arn:aws:lambda:ap-southeast-1:580247275435:layer:LambdaInsightsExtension:12`
Asia-Pacífico (Sídney)	`arn:aws:lambda:ap-southeast-2:580247275435:layer:LambdaInsightsExtension:12`
Asia-Pacífico (Tokio)	`arn:aws:lambda:ap-northeast-1:580247275435:layer:LambdaInsightsExtension:12`

Región	ARN
Canadá (centro)	`arn:aws:lambda:ca-central-1:580247275435:layer:LambdaInsightsExtension:12`
Europa (Fráncfort)	`arn:aws:lambda:eu-central-1:580247275435:layer:LambdaInsightsExtension:12`
Europa (Irlanda)	`arn:aws:lambda:eu-west-1:580247275435:layer:LambdaInsightsExtension:12`
Europa (Londres)	`arn:aws:lambda:eu-west-2:580247275435:layer:LambdaInsightsExtension:12`
Europa (París)	`arn:aws:lambda:eu-west-3:580247275435:layer:LambdaInsightsExtension:12`
Europa (Estocolmo)	`arn:aws:lambda:eu-north-1:580247275435:layer:LambdaInsightsExtension:12`
América del Sur (São Paulo)	`arn:aws:lambda:sa-east-1:580247275435:layer:LambdaInsightsExtension:12`

1.0.86.0

Con la versión 1.0.54.0 de la extensión, algunas veces las métricas de la memoria se notificaron incorrectamente y, otras veces, eran superiores al 100 %. La versión 1.0.86.0 corrige el problema de medición de la memoria al utilizar los mismos datos de eventos que las métricas de la plataforma de Lambda. Esto significa que puede ver un cambio drástico en los valores registrados de la métrica de memoria. Esto se logra mediante el uso de la nueva API de Lambda Logs. Esto proporciona una medición más precisa del uso de la memoria de entorno de pruebas de Lambda. Sin embargo, hay que tener en cuenta que la API de Lambda Logs no puede entregar eventos de informes de la plataforma si el tiempo de espera de la función entorno de pruebas se agota y, por lo tanto, se cierra. En este caso. Lambda Insights no puede registrar las métricas de invocación. Para obtener más información acerca de la API de Lambda Logs, consulte AWSLambda Logs API.

Características nuevas en la versión 1.0.86.0

- Utiliza la API de Lambda Logs para corregir la métrica de la memoria. Esto resuelve el problema anterior en el que las estadísticas de la memoria eran superiores al 100 %.
- Presenta `Init Duration` como una nueva métrica de CloudWatch.
- Utiliza los ARN de invocación para agregar una dimensión version (versión) para alias y versiones invocadas. Si utiliza alias o versiones de Lambda para lograr implementaciones progresivas (como implementaciones azul-verde), puede ver las métricas mediante el alias invocado. La dimensión version (versión) no se aplica si la función no utiliza un alias o una versión. Para obtener más información, consulte Alias de la función de Lambda.
- Agrega un `billed_mb_ms field` a los eventos de rendimiento para mostrar el costo por invocación. Esto no tiene en cuenta ningún costo asociado con la simultaneidad aprovisionada.
- Agrega los campos `billed_duration` y `duration` a los eventos de rendimiento.

ARN para la versión 1.0.86.0

En la siguiente tabla se muestran los ARN que se deben utilizar para esta versión de la extensión en cada Región de AWS en la que está disponible.

Región	ARN
Este de EE. UU. (Norte de Virginia)	`arn:aws:lambda:us-east-1:580247275435:layer:LambdaInsightsExtension:11`
Este de EE. UU. (Ohio)	`arn:aws:lambda:us-east-2:580247275435:layer:LambdaInsightsExtension:11`
Oeste de EE. UU. (Norte de California)	`arn:aws:lambda:us-west-1:580247275435:layer:LambdaInsightsExtension:11`
Oeste de EE. UU. (Oregón)	`arn:aws:lambda:us-west-2:580247275435:layer:LambdaInsightsExtension:11`
Asia-Pacífico (Bombay)	`arn:aws:lambda:ap-south-1:580247275435:layer:LambdaInsightsExtension:11`
Asia-Pacífico (Seúl)	`arn:aws:lambda:ap-northeast-2:580247275435:layer:LambdaInsightsExtension:11`

Región	ARN
Asia-Pacífico (Singapur)	`arn:aws:lambda:ap-southeast-1:580247275435:layer:LambdaInsightsExtension:11`
Asia-Pacífico (Sídney)	`arn:aws:lambda:ap-southeast-2:580247275435:layer:LambdaInsightsExtension:11`
Asia-Pacífico (Tokio)	`arn:aws:lambda:ap-northeast-1:580247275435:layer:LambdaInsightsExtension:11`
Canadá (centro)	`arn:aws:lambda:ca-central-1:580247275435:layer:LambdaInsightsExtension:11`
Europa (Fráncfort)	`arn:aws:lambda:eu-central-1:580247275435:layer:LambdaInsightsExtension:11`
Europa (Irlanda)	`arn:aws:lambda:eu-west-1:580247275435:layer:LambdaInsightsExtension:11`
Europa (Londres)	`arn:aws:lambda:eu-west-2:580247275435:layer:LambdaInsightsExtension:11`
Europa (París)	`arn:aws:lambda:eu-west-3:580247275435:layer:LambdaInsightsExtension:11`
Europa (Estocolmo)	`arn:aws:lambda:eu-north-1:580247275435:layer:LambdaInsightsExtension:11`
América del Sur (São Paulo)	`arn:aws:lambda:sa-east-1:580247275435:layer:LambdaInsightsExtension:11`

1.0.54.0

La versión 1.0.54.0 fue la versión inicial de la extensión de Lambda Insights.

ARN para la versión 1.0.54.0

En la siguiente tabla se muestran los ARN que se deben utilizar para esta versión de la extensión en cada Región de AWS en la que está disponible.

Región	ARN
Este de EE. UU. (Norte de Virginia)	`arn:aws:lambda:us-east-1:580247275435:layer:LambdaInsightsExtension:2`
Este de EE. UU. (Ohio)	`arn:aws:lambda:us-east-2:580247275435:layer:LambdaInsightsExtension:2`
Oeste de EE. UU. (Norte de California)	`arn:aws:lambda:us-west-1:580247275435:layer:LambdaInsightsExtension:2`
Oeste de EE. UU. (Oregón)	`arn:aws:lambda:us-west-2:580247275435:layer:LambdaInsightsExtension:2`
Asia-Pacífico (Bombay)	`arn:aws:lambda:ap-south-1:580247275435:layer:LambdaInsightsExtension:2`
Asia-Pacífico (Seúl)	`arn:aws:lambda:ap-northeast-2:580247275435:layer:LambdaInsightsExtension:2`
Asia-Pacífico (Singapur)	`arn:aws:lambda:ap-southeast-1:580247275435:layer:LambdaInsightsExtension:2`
Asia-Pacífico (Sídney)	`arn:aws:lambda:ap-southeast-2:580247275435:layer:LambdaInsightsExtension:2`
Asia-Pacífico (Tokio)	`arn:aws:lambda:ap-northeast-1:580247275435:layer:LambdaInsightsExtension:2`
Canadá (centro)	`arn:aws:lambda:ca-central-1:580247275435:layer:LambdaInsightsExtension:2`
Europa (Fráncfort)	`arn:aws:lambda:eu-central-1:580247275435:layer:LambdaInsightsExtension:2`
Europa (Irlanda)	`arn:aws:lambda:eu-west-1:580247275435:layer:LambdaInsightsExtension:2`

Región	ARN
Europa (Londres)	`arn:aws:lambda:eu-west-2:580247275435:layer:LambdaInsightsExtension:2`
Europa (París)	`arn:aws:lambda:eu-west-3:580247275435:layer:LambdaInsightsExtension:2`
Europa (Estocolmo)	`arn:aws:lambda:eu-north-1:580247275435:layer:LambdaInsightsExtension:2`
América del Sur (São Paulo)	`arn:aws:lambda:sa-east-1:580247275435:layer:LambdaInsightsExtension:2`

Plataformas ARM64

En esta sección se enumeran las versiones de la extensión de Lambda Insights para plataformas ARM64 y los ARN que se utilizan para estas extensiones en cada región de AWS.

> **Important**
>
> Las extensiones 1.0.317.0 y posteriores de Lambda Insights no son compatibles con Amazon Linux 1.

1.0.317.0

En la versión 1.0.317.0 se retira la compatibilidad con la plataforma Amazon Linux 1 y se corrigen errores. También se admiten las regiones de AWS GovCloud (US).

Región	ARN
Este de EE. UU. (Norte de Virginia)	`arn:aws:lambda:us-east-1:580247275435:layer:LambdaInsightsExtension-Arm64:19`
Este de EE. UU. (Ohio)	`arn:aws:lambda:us-east-2:580247275435:layer:LambdaInsightsExtension-Arm64:21`

Región	ARN
Oeste de EE. UU. (Norte de California)	arn:aws:lambda:us-west-1:580247275435:layer:LambdaInsightsExtension-Arm64:17
Oeste de EE. UU. (Oregón)	arn:aws:lambda:us-west-2:580247275435:layer:LambdaInsightsExtension-Arm64:19
África (Ciudad del Cabo)	arn:aws:lambda:af-south-1:012438385374:layer:LambdaInsightsExtension-Arm64:17
Asia-Pacífico (Hong Kong)	arn:aws:lambda:ap-east-1:519774774795:layer:LambdaInsightsExtension-Arm64:17
Asia-Pacífico (Hyderabad)	arn:aws:lambda:ap-south-2:891564319516:layer:LambdaInsightsExtension-Arm64:5
Asia-Pacífico (Yakarta)	arn:aws:lambda:ap-southeast-3:439286490199:layer:LambdaInsightsExtension-Arm64:17
Asia-Pacífico (Bombay)	arn:aws:lambda:ap-south-1:580247275435:layer:LambdaInsightsExtension-Arm64:21
Asia-Pacífico (Osaka)	arn:aws:lambda:ap-northeast-3:194566237122:layer:LambdaInsightsExtension-Arm64:16
Asia-Pacífico (Seúl)	arn:aws:lambda:ap-northeast-2:580247275435:layer:LambdaInsightsExtension-Arm64:18
Asia-Pacífico (Singapur)	arn:aws:lambda:ap-southeast-1:580247275435:layer:LambdaInsightsExtension-Arm64:19
Asia-Pacífico (Sídney)	arn:aws:lambda:ap-southeast-2:580247275435:layer:LambdaInsightsExtension-Arm64:19
Asia-Pacífico (Tokio)	arn:aws:lambda:ap-northeast-1:580247275435:layer:LambdaInsightsExtension-Arm64:30
Canadá (centro)	arn:aws:lambda:ca-central-1:580247275435:layer:LambdaInsightsExtension-Arm64:17

Región	ARN
Europa (Fráncfort)	arn:aws:lambda:eu-central-1:580247275435:layer:LambdaInsightsExtension-Arm64:19
Europa (Irlanda)	arn:aws:lambda:eu-west-1:580247275435:layer:LambdaInsightsExtension-Arm64:19
Europa (Londres)	arn:aws:lambda:eu-west-2:580247275435:layer:LambdaInsightsExtension-Arm64:19
Europa (Milán)	arn:aws:lambda:eu-south-1:339249233099:layer:LambdaInsightsExtension-Arm64:17
Europa (París)	arn:aws:lambda:eu-west-3:580247275435:layer:LambdaInsightsExtension-Arm64:17
Europa (España)	arn:aws:lambda:eu-south-2:352183217350:layer:LambdaInsightsExtension-Arm64:5
Europa (Estocolmo)	arn:aws:lambda:eu-north-1:580247275435:layer:LambdaInsightsExtension-Arm64:17
Medio Oriente (Baréin)	arn:aws:lambda:me-south-1:285320876703:layer:LambdaInsightsExtension-Arm64:17
América del Sur (São Paulo)	arn:aws:lambda:sa-east-1:580247275435:layer:LambdaInsightsExtension-Arm64:17
AWS GovCloud (Este de EE. UU.)	arn:aws-us-gov:lambda:us-gov-east-1:122132214140:layer:LambdaInsightsExtension-Arm64:1
AWS GovCloud (Oeste de EE. UU.)	arn:aws-us-gov:lambda:us-gov-west-1:751350123760:layer:LambdaInsightsExtension-Arm64:1

1.0.295.0

La versión 1.0.295.0 incluye actualizaciones de dependencias para todos los tiempos de ejecución compatibles.

Región	ARN
Este de EE. UU. (Norte de Virginia)	`arn:aws:lambda:us-east-1:580247275435:layer:LambdaInsightsExtension-Arm64:18`
Este de EE. UU. (Ohio)	`arn:aws:lambda:us-east-2:580247275435:layer:LambdaInsightsExtension-Arm64:20`
Oeste de EE. UU. (Norte de California)	`arn:aws:lambda:us-west-1:580247275435:layer:LambdaInsightsExtension-Arm64:16`
Oeste de EE. UU. (Oregón)	`arn:aws:lambda:us-west-2:580247275435:layer:LambdaInsightsExtension-Arm64:18`
África (Ciudad del Cabo)	`arn:aws:lambda:af-south-1:012438385374:layer:LambdaInsightsExtension-Arm64:16`
Asia-Pacífico (Hong Kong)	`arn:aws:lambda:ap-east-1:519774774795:layer:LambdaInsightsExtension-Arm64:16`
Asia-Pacífico (Hyderabad)	`arn:aws:lambda:ap-south-2:891564319516:layer:LambdaInsightsExtension-Arm64:4`
Asia-Pacífico (Yakarta)	`arn:aws:lambda:ap-southeast-3:439286490199:layer:LambdaInsightsExtension-Arm64:16`
Asia-Pacífico (Bombay)	`arn:aws:lambda:ap-south-1:580247275435:layer:LambdaInsightsExtension-Arm64:20`
Asia-Pacífico (Osaka)	`arn:aws:lambda:ap-northeast-3:194566237122:layer:LambdaInsightsExtension-Arm64:15`
Asia-Pacífico (Seúl)	`arn:aws:lambda:ap-northeast-2:580247275435:layer:LambdaInsightsExtension-Arm64:17`
Asia-Pacífico (Singapur)	`arn:aws:lambda:ap-southeast-1:580247275435:layer:LambdaInsightsExtension-Arm64:18`

Región	ARN
Asia-Pacífico (Sídney)	arn:aws:lambda:ap-southeast-2:580247275435:layer:LambdaInsightsExtension-Arm64:18
Asia-Pacífico (Tokio)	arn:aws:lambda:ap-northeast-1:580247275435:layer:LambdaInsightsExtension-Arm64:29
Canadá (centro)	arn:aws:lambda:ca-central-1:580247275435:layer:LambdaInsightsExtension-Arm64:16
Europa (Fráncfort)	arn:aws:lambda:eu-central-1:580247275435:layer:LambdaInsightsExtension-Arm64:18
Europa (Irlanda)	arn:aws:lambda:eu-west-1:580247275435:layer:LambdaInsightsExtension-Arm64:18
Europa (Londres)	arn:aws:lambda:eu-west-2:580247275435:layer:LambdaInsightsExtension-Arm64:18
Europa (Milán)	arn:aws:lambda:eu-south-1:339249233099:layer:LambdaInsightsExtension-Arm64:16
Europa (París)	arn:aws:lambda:eu-west-3:580247275435:layer:LambdaInsightsExtension-Arm64:16
Europa (España)	arn:aws:lambda:eu-south-2:352183217350:layer:LambdaInsightsExtension-Arm64:4
Europa (Estocolmo)	arn:aws:lambda:eu-north-1:580247275435:layer:LambdaInsightsExtension-Arm64:16
Medio Oriente (Baréin)	arn:aws:lambda:me-south-1:285320876703:layer:LambdaInsightsExtension-Arm64:16
América del Sur (São Paulo)	arn:aws:lambda:sa-east-1:580247275435:layer:LambdaInsightsExtension-Arm64:16

1.0.275.0

La versión 1.0.275.0 incluye correcciones de errores para todos los tiempos de ejecución compatibles y compatibilidad con las regiones de Europa (España) y Asia Pacífico (Hyderabad).

Región	ARN
Este de EE. UU. (Norte de Virginia)	`arn:aws:lambda:us-east-1:580247275435:layer:LambdaInsightsExtension-Arm64:16`
Este de EE. UU. (Ohio)	`arn:aws:lambda:us-east-2:580247275435:layer:LambdaInsightsExtension-Arm64:18`
Oeste de EE. UU. (Norte de California)	`arn:aws:lambda:us-west-1:580247275435:layer:LambdaInsightsExtension-Arm64:14`
Oeste de EE. UU. (Oregón)	`arn:aws:lambda:us-west-2:580247275435:layer:LambdaInsightsExtension-Arm64:16`
África (Ciudad del Cabo)	`arn:aws:lambda:af-south-1:012438385374:layer:LambdaInsightsExtension-Arm64:14`
Asia-Pacífico (Hong Kong)	`arn:aws:lambda:ap-east-1:519774774795:layer:LambdaInsightsExtension-Arm64:14`
Asia-Pacífico (Hyderabad)	`arn:aws:lambda:ap-south-2:891564319516:layer:LambdaInsightsExtension-Arm64:2`
Asia-Pacífico (Yakarta)	`arn:aws:lambda:ap-southeast-3:439286490199:layer:LambdaInsightsExtension-Arm64:14`
Asia-Pacífico (Bombay)	`arn:aws:lambda:ap-south-1:580247275435:layer:LambdaInsightsExtension-Arm64:18`
Asia-Pacífico (Osaka)	`arn:aws:lambda:ap-northeast-3:194566237122:layer:LambdaInsightsExtension-Arm64:13`
Asia-Pacífico (Seúl)	`arn:aws:lambda:ap-northeast-2:580247275435:layer:LambdaInsightsExtension-Arm64:15`

Región	ARN
Asia-Pacífico (Singapur)	arn:aws:lambda:ap-southeast-1:580247275435:layer:LambdaInsightsExtension-Arm64:16
Asia-Pacífico (Sídney)	arn:aws:lambda:ap-southeast-2:580247275435:layer:LambdaInsightsExtension-Arm64:16
Asia-Pacífico (Tokio)	arn:aws:lambda:ap-northeast-1:580247275435:layer:LambdaInsightsExtension-Arm64:27
Canadá (centro)	arn:aws:lambda:ca-central-1:580247275435:layer:LambdaInsightsExtension-Arm64:14
Europa (Fráncfort)	arn:aws:lambda:eu-central-1:580247275435:layer:LambdaInsightsExtension-Arm64:16
Europa (Irlanda)	arn:aws:lambda:eu-west-1:580247275435:layer:LambdaInsightsExtension-Arm64:16
Europa (Londres)	arn:aws:lambda:eu-west-2:580247275435:layer:LambdaInsightsExtension-Arm64:16
Europa (Milán)	arn:aws:lambda:eu-south-1:339249233099:layer:LambdaInsightsExtension-Arm64:14
Europa (París)	arn:aws:lambda:eu-west-3:580247275435:layer:LambdaInsightsExtension-Arm64:14
Europa (España)	arn:aws:lambda:eu-south-2:352183217350:layer:LambdaInsightsExtension-Arm64:2
Europa (Estocolmo)	arn:aws:lambda:eu-north-1:580247275435:layer:LambdaInsightsExtension-Arm64:14
Medio Oriente (Baréin)	arn:aws:lambda:me-south-1:285320876703:layer:LambdaInsightsExtension-Arm64:14
América del Sur (São Paulo)	arn:aws:lambda:sa-east-1:580247275435:layer:LambdaInsightsExtension-Arm64:14

1.0.273.0

La versión 1.0.273.0 incluye correcciones de errores para todos los tiempos de ejecución compatibles.

Región	ARN
Este de EE. UU. (Norte de Virginia)	`arn:aws:lambda:us-east-1:580247275435:layer:LambdaInsightsExtension-Arm64:12`
Este de EE. UU. (Ohio)	`arn:aws:lambda:us-east-2:580247275435:layer:LambdaInsightsExtension-Arm64:14`
Oeste de EE. UU. (Norte de California)	`arn:aws:lambda:us-west-1:580247275435:layer:LambdaInsightsExtension-Arm64:9`
Oeste de EE. UU. (Oregón)	`arn:aws:lambda:us-west-2:580247275435:layer:LambdaInsightsExtension-Arm64:12`
África (Ciudad del Cabo)	`arn:aws:lambda:af-south-1:012438385374:layer:LambdaInsightsExtension-Arm64:9`
Asia-Pacífico (Hong Kong)	`arn:aws:lambda:ap-east-1:519774774795:layer:LambdaInsightsExtension-Arm64:9`
Asia-Pacífico (Yakarta)	`arn:aws:lambda:ap-southeast-3:439286490199:layer:LambdaInsightsExtension-Arm64:9`
Asia-Pacífico (Bombay)	`arn:aws:lambda:ap-south-1:580247275435:layer:LambdaInsightsExtension-Arm64:14`
Asia-Pacífico (Osaka)	`arn:aws:lambda:ap-northeast-3:194566237122:layer:LambdaInsightsExtension-Arm64:9`
Asia-Pacífico (Seúl)	`arn:aws:lambda:ap-northeast-2:580247275435:layer:LambdaInsightsExtension-Arm64:11`
Asia-Pacífico (Singapur)	`arn:aws:lambda:ap-southeast-1:580247275435:layer:LambdaInsightsExtension-Arm64:12`

Región	ARN
Asia-Pacífico (Sídney)	arn:aws:lambda:ap-southeast-2:580247275435:layer:LambdaInsightsExtension-Arm64:12
Asia-Pacífico (Tokio)	arn:aws:lambda:ap-northeast-1:580247275435:layer:LambdaInsightsExtension-Arm64:23
Canadá (centro)	arn:aws:lambda:ca-central-1:580247275435:layer:LambdaInsightsExtension-Arm64:10
Europa (Fráncfort)	arn:aws:lambda:eu-central-1:580247275435:layer:LambdaInsightsExtension-Arm64:12
Europa (Irlanda)	arn:aws:lambda:eu-west-1:580247275435:layer:LambdaInsightsExtension-Arm64:12
Europa (Londres)	arn:aws:lambda:eu-west-2:580247275435:layer:LambdaInsightsExtension-Arm64:12
Europa (Milán)	arn:aws:lambda:eu-south-1:339249233099:layer:LambdaInsightsExtension-Arm64:9
Europa (París)	arn:aws:lambda:eu-west-3:580247275435:layer:LambdaInsightsExtension-Arm64:10
Europa (Estocolmo)	arn:aws:lambda:eu-north-1:580247275435:layer:LambdaInsightsExtension-Arm64:10
Medio Oriente (Baréin)	arn:aws:lambda:me-south-1:285320876703:layer:LambdaInsightsExtension-Arm64:9
América del Sur (São Paulo)	arn:aws:lambda:sa-east-1:580247275435:layer:LambdaInsightsExtension-Arm64:10

1.0.229.0

La versión 1.0.229.0 incluye correcciones de errores para todos los tiempos de ejecución compatibles. También añade compatibilidad con las siguientes regiones:

- Oeste de EE. UU. (Norte de California)
- África (Ciudad del Cabo)
- Asia-Pacífico (Hong Kong)
- Asia-Pacífico (Yakarta)
- Asia-Pacífico (Osaka)
- Asia-Pacífico (Seúl)
- Canadá (Centro)
- Europa (Milán)
- Europa (París)
- Europa (Estocolmo)
- Medio Oriente (Baréin)
- América del Sur (São Paulo)

Región	ARN
Este de EE. UU. (Norte de Virginia)	`arn:aws:lambda:us-east-1:580247275435:layer:LambdaInsightsExtension-Arm64:5`
Este de EE. UU. (Ohio)	`arn:aws:lambda:us-east-2:580247275435:layer:LambdaInsightsExtension-Arm64:7`
Oeste de EE. UU. (Norte de California)	`arn:aws:lambda:us-west-1:580247275435:layer:LambdaInsightsExtension-Arm64:3`
Oeste de EE. UU. (Oregón)	`arn:aws:lambda:us-west-2:580247275435:layer:LambdaInsightsExtension-Arm64:5`
África (Ciudad del Cabo)	`arn:aws:lambda:af-south-1:012438385374:layer:LambdaInsightsExtension-Arm64:2`
Asia-Pacífico (Hong Kong)	`arn:aws:lambda:ap-east-1:519774774795:layer:LambdaInsightsExtension-Arm64:2`
Asia-Pacífico (Yakarta)	`arn:aws:lambda:ap-southeast-3:439286490199:layer:LambdaInsightsExtension-Arm64:2`

Región	ARN
Asia-Pacífico (Bombay)	arn:aws:lambda:ap-south-1:580247275435:layer:LambdaInsightsExtension-Arm64:7
Asia-Pacífico (Osaka)	arn:aws:lambda:ap-northeast-3:194566237122:layer:LambdaInsightsExtension-Arm64:2
Asia-Pacífico (Seúl)	arn:aws:lambda:ap-northeast-2:580247275435:layer:LambdaInsightsExtension-Arm64:4
Asia-Pacífico (Singapur)	arn:aws:lambda:ap-southeast-1:580247275435:layer:LambdaInsightsExtension-Arm64:5
Asia-Pacífico (Sídney)	arn:aws:lambda:ap-southeast-2:580247275435:layer:LambdaInsightsExtension-Arm64:5
Asia-Pacífico (Tokio)	arn:aws:lambda:ap-northeast-1:580247275435:layer:LambdaInsightsExtension-Arm64:11
Canadá (centro)	arn:aws:lambda:ca-central-1:580247275435:layer:LambdaInsightsExtension-Arm64:3
Europa (Fráncfort)	arn:aws:lambda:eu-central-1:580247275435:layer:LambdaInsightsExtension-Arm64:5
Europa (Irlanda)	arn:aws:lambda:eu-west-1:580247275435:layer:LambdaInsightsExtension-Arm64:5
Europa (Londres)	arn:aws:lambda:eu-west-2:580247275435:layer:LambdaInsightsExtension-Arm64:5
Europa (España)	arn:aws:lambda:eu-south-1:339249233099:layer:LambdaInsightsExtension-Arm64:2
Europa (París)	arn:aws:lambda:eu-west-3:580247275435:layer:LambdaInsightsExtension-Arm64:3
Europa (Estocolmo)	arn:aws:lambda:eu-north-1:580247275435:layer:LambdaInsightsExtension-Arm64:3

Región	ARN
Medio Oriente (Baréin)	`arn:aws:lambda:me-south-1:285320876703:layer:LambdaInsightsExtension-Arm64:2`
América del Sur (São Paulo)	`arn:aws:lambda:sa-east-1:580247275435:layer:LambdaInsightsExtension-Arm64:3`

1.0.135.0

La versión 1.0.135.0 incluye correcciones de errores sobre la forma en que Lambda Insights recopila e informa datos sobre el uso del descriptor de discos y archivos. En versiones anteriores de la extensión, la métrica `tmp_free` informó el espacio libre máximo en el directorio `/tmp` mientras se ejecuta una función. Esta versión cambia la métrica para que informe, en cambio, el valor mínimo, lo que la hace más útil a la hora de evaluar el uso del disco. Para obtener más información sobre las cuotas de almacenamiento del directorio `tmp`, consulte Cuotas de Lambda.

La versión 1.0.135.0 también informa ahora el uso del descriptor de archivos (`fd_use` y `fd_max`) como el valor máximo en todos los procesos, en lugar de informar el nivel del sistema operativo.

Región	ARN
Este de EE. UU. (Norte de Virginia)	`arn:aws:lambda:us-east-1:580247275435:layer:LambdaInsightsExtension-Arm64:2`
Este de EE. UU. (Ohio)	`arn:aws:lambda:us-east-2:580247275435:layer:LambdaInsightsExtension-Arm64:2`
Oeste de EE. UU. (Oregón)	`arn:aws:lambda:us-west-2:580247275435:layer:LambdaInsightsExtension-Arm64:2`
Asia-Pacífico (Bombay)	`arn:aws:lambda:ap-south-1:580247275435:layer:LambdaInsightsExtension-Arm64:2`
Asia-Pacífico (Singapur)	`arn:aws:lambda:ap-southeast-1:580247275435:layer:LambdaInsightsExtension-Arm64:2`

Región	ARN
Asia-Pacífico (Sídney)	arn:aws:lambda:ap-southeast-2:580247275435:layer:LambdaInsightsExtension-Arm64:2
Asia-Pacífico (Tokio)	arn:aws:lambda:ap-northeast-1:580247275435:layer:LambdaInsightsExtension-Arm64:2
Europa (Fráncfort)	arn:aws:lambda:eu-central-1:580247275435:layer:LambdaInsightsExtension-Arm64:2
Europa (Irlanda)	arn:aws:lambda:eu-west-1:580247275435:layer:LambdaInsightsExtension-Arm64:2
Europa (Londres)	arn:aws:lambda:eu-west-2:580247275435:layer:LambdaInsightsExtension-Arm64:2

1.0.119.0

Región	ARN
Este de EE. UU. (Norte de Virginia)	arn:aws:lambda:us-east-1:580247275435:layer:LambdaInsightsExtension-Arm64:1
Este de EE. UU. (Ohio)	arn:aws:lambda:us-east-2:580247275435:layer:LambdaInsightsExtension-Arm64:1
Oeste de EE. UU. (Oregón)	arn:aws:lambda:us-west-2:580247275435:layer:LambdaInsightsExtension-Arm64:1
Asia-Pacífico (Bombay)	arn:aws:lambda:ap-south-1:580247275435:layer:LambdaInsightsExtension-Arm64:1
Asia-Pacífico (Singapur)	arn:aws:lambda:ap-southeast-1:580247275435:layer:LambdaInsightsExtension-Arm64:1
Asia-Pacífico (Sídney)	arn:aws:lambda:ap-southeast-2:580247275435:layer:LambdaInsightsExtension-Arm64:1

Región	ARN
Asia-Pacífico (Tokio)	`arn:aws:lambda:ap-northeast-1:580247275435:layer:LambdaInsightsExtension-Arm64:1`
Europa (Fráncfort)	`arn:aws:lambda:eu-central-1:580247275435:layer:LambdaInsightsExtension-Arm64:1`
Europa (Irlanda)	`arn:aws:lambda:eu-west-1:580247275435:layer:LambdaInsightsExtension-Arm64:1`
Europa (Londres)	`arn:aws:lambda:eu-west-2:580247275435:layer:LambdaInsightsExtension-Arm64:1`

Uso de la consola para habilitar Lambda Insights en una función de Lambda existente

Siga estos pasos en la consola de Lambda para habilitar Lambda Insights en una función de Lambda existente.

Cómo habilitar Lambda Insights en una función de Lambda

1. Abra la consola de AWS Lambda en https://console.aws.amazon.com/lambda/.
2. Elija el nombre de una función y, a continuación, seleccione la pestaña Configuration (Configuración) en la siguiente pantalla.
3. En la pestaña Configuración, elija Herramientas de supervisión y operaciones en el menú de navegación izquierdo y, a continuación, elija Editar.

   Esto lo llevará a una pantalla en la que puede editar las herramientas de supervisión.
4. En Supervisión mejorada de Lambda Insights, elija Editar.
5. En la sección CloudWatch Lambda Insights, habilite Supervisión mejorada y, a continuación, elija Guardar.

Uso de AWS CLI para habilitar Lambda Insights en una función de Lambda existente

Siga estos pasos para utilizar la AWS CLI para habilitar Lambda Insights en una función de Lambda existente.

Paso 1: actualice los permisos de la función

Para actualizar los permisos de la función

- Adjunte la politíca de IAM administrada CloudWatchLambdaInsightsExecutionRolePolicy al rol de ejecución de la función mediante el ingreso del siguiente comando.

```
aws iam attach-role-policy \
--role-name function-execution-role \
--policy-arn "arn:aws:iam::aws:policy/CloudWatchLambdaInsightsExecutionRolePolicy"
```

Paso 2: instale la extensión de Lambda

Instale la extensión de Lambda con el siguiente comando. Reemplace el valor ARN para el parámetro `layers` con el ARN que concuerde con su Región y la versión de extensión que desea utilizar. Para obtener más información, consulte Versiones disponibles de la extensión de Lambda Insights.

```
aws lambda update-function-configuration \
   --function-name function-name \
   --layers "arn:aws:lambda:us-west-1:580247275435:layer:LambdaInsightsExtension:14"
```

Paso 3: habilite el punto de conexión de VPC de CloudWatch Logs

Este paso sólo es necesario para las funciones que se ejecutan en una subred privada sin acceso a Internet y si aún no ha configurado un punto de enlace de la virtual private cloud (VPC) de CloudWatch Logs.

Si necesita realizar este paso, ingrese el siguiente comando, lo que reemplaza los espacios disponibles con información para la VPC.

Para obtener más información, consulte Using CloudWatch Logs with Interface VPC Endpoints (Uso de CloudWatch Logs con puntos de enlace de la VPC de tipo interfaz).

```
aws ec2 create-vpc-endpoint \
--vpc-id vpcId \
--vpc-endpoint-type Interface \
--service-name com.amazonaws.region.logs \
--subnet-id subnetId
--security-group-id securitygroupId
```

Uso de la CLI de AWS SAM para habilitar Lambda Insights en una función de Lambda existente

Siga estos pasos para utilizar la AWS CLI de AWS SAM para habilitar Lambda Insights en una función de Lambda existente.

Si aún no instaló la última versión de la CLI de AWS SAM, primero debe instalarla o actualizarla. Para obtener más información, consulte Instalación de la CLI de AWS SAM.

Paso 1: instale la capa

Para que la extensión Lambda Insights esté disponible para todas las funciones de Lambda, agregue una propiedad `Layers` a la sección `Globals` de la plantilla SAM con el ARN de la capa de Lambda Insights. En el siguiente ejemplo se utiliza la capa para la versión inicial de Lambda Insights. Para obtener la versión más reciente de la capa de extensión de Lambda Insights, consulte Versiones disponibles de la extensión de Lambda Insights.

```
Globals:
  Function:
    Layers:
      - !Sub "arn:aws:lambda:
${AWS::Region}:580247275435:layer:LambdaInsightsExtension:14"
```

Para habilitar esta capa para una sola función, agregue la propiedad `Layers` a la función como se muestra en este ejemplo.

```
Resources:
  MyFunction:
    Type: AWS::Serverless::Function
    Properties:
      Layers:
        - !Sub "arn:aws:lambda:
${AWS::Region}:580247275435:layer:LambdaInsightsExtension:14"
```

Paso 2: agregue la política administrada

Para cada función, agregue la política de IAM CloudWatchLambdaInsightsExecutionRolePolicy.

AWS SAM no es compatible con políticas globales, por lo que debe habilitar las de cada función de manera individual, como se muestra en este ejemplo. Para obtener más información acerca de las globales, consulte Globals Section (Sección de globales).

```
Resources:
  MyFunction:
    Type: AWS::Serverless::Function
    Properties:
      Policies:
        - CloudWatchLambdaInsightsExecutionRolePolicy
```

Invocaciones locales

La CLI de AWS SAM admite extensiones Lambda. Sin embargo, cada invocación ejecutada localmente restablece el entorno en tiempo de ejecución. Los datos de Lambda Insights no estarán disponibles desde las invocaciones locales porque el tiempo de ejecución se reinicia sin un evento de cierre. Para obtener más información, consulte Versión 1.6.0: agregar compatibilidad con las pruebas locales de las extensiones de AWS Lambda.

Solución de problemas

Para solucionar problemas con la instalación de Lambda Insights, agregue la siguiente variable de entorno a la función de Lambda para habilitar el registro de depuración.

```
Resources:
  MyFunction:
    Type: AWS::Serverless::Function
    Properties:
      Environment:
        Variables:
          LAMBDA_INSIGHTS_LOG_LEVEL: info
```

Uso de AWS CloudFormation para habilitar Lambda Insights en una función de Lambda existente

Siga estos pasos para utilizar AWS CloudFormation para habilitar Lambda Insights en una función de Lambda existente.

Paso 1: instale la capa

Agregue la capa de Lambda Insights a la propiedad `Layers` dentro del ARN de la capa de Lambda Insights. En el siguiente ejemplo se utiliza la capa para la versión inicial de Lambda Insights. Para obtener la versión más reciente de la capa de extensión de Lambda Insights, consulte Versiones disponibles de la extensión de Lambda Insights.

```
Resources:
  MyFunction:
    Type: AWS::Lambda::Function
    Properties:
      Layers:
        - !Sub "arn:aws:lambda:
${AWS::Region}:580247275435:layer:LambdaInsightsExtension:14"
```

Paso 2: agregue la política administrada

Agregue la política de IAM CloudWatchLambdaInsightsExecutionRolePolicy al rol de ejecución de la función.

```
Resources:
  MyFunctionExecutionRole:
    Type: 'AWS::IAM::Role'
    Properties:
      ManagedPolicyArns:
        - 'arn:aws:iam::aws:policy/CloudWatchLambdaInsightsExecutionRolePolicy'
```

Paso 3: (opcional) Agregue el punto de conexión de VPC

Este paso sólo es necesario para las funciones que se ejecutan en una subred privada sin acceso a Internet y si aún no ha configurado un punto de enlace de la virtual private cloud (VPC) de CloudWatch Logs. Para obtener más información, consulte Using CloudWatch Logs with Interface VPC Endpoints (Uso de CloudWatch Logs con puntos de enlace de la VPC de tipo interfaz).

```
Resources:
  CloudWatchLogsVpcPrivateEndpoint:
    Type: AWS::EC2::VPCEndpoint
    Properties:
      PrivateDnsEnabled: 'true'
      VpcEndpointType: Interface
      VpcId: !Ref: VPC
      ServiceName: !Sub com.amazonaws.${AWS::Region}.logs
      SecurityGroupIds:
        - !Ref InterfaceVpcEndpointSecurityGroup
      SubnetIds:
        - !Ref PublicSubnet01
        - !Ref PublicSubnet02
        - !Ref PublicSubnet03
```

Uso de AWS CDK para habilitar Lambda Insights en una función de Lambda existente

Siga estos pasos para utilizar AWS CDK para habilitar Lambda Insights en una función de Lambda existente. Para seguir estos pasos, ya debe estar usando AWS CDK para administrar los recursos.

Los comandos de esta sección se encuentran en TypeScript.

Primero, actualice los permisos de la función.

```
executionRole.addManagedPolicy(
 ManagedPolicy.fromAwsManagedPolicyName('CloudWatchLambdaInsightsExecutionRolePolicy')
);
```

A continuación, instale la extensión en la función de Lambda. Reemplace el valor ARN para el parámetro `layerArn` con el ARN que concuerde con la Región y la versión de extensión que desea utilizar. Para obtener más información, consulte Versiones disponibles de la extensión de Lambda Insights.

```
import lambda = require('@aws-cdk/aws-lambda');
const layerArn = 'arn:aws:lambda:us-
west-1:580247275435:layer:LambdaInsightsExtension:14';
const layer = lambda.LayerVersion.fromLayerVersionArn(this, 'LayerFromArn', layerArn);
```

Si es necesario, habilite el punto de enlace de la virtual private cloud (VPC) para CloudWatch Logs. Este paso sólo es necesario para las funciones que se ejecutan en una subred privada sin acceso a Internet y si aún no ha configurado un punto de enlace de la VPC de CloudWatch Logs.

```
const cloudWatchLogsEndpoint = vpc.addInterfaceEndpoint('cwl-gateway', {
  service: InterfaceVpcEndpointAwsService.CLOUDWATCH_LOGS,
});

cloudWatchLogsEndpoint.connections.allowDefaultPortFromAnyIpv4();
```

Uso de Serverless Framework para habilitar Lambda Insights en una función de Lambda existente

Siga estos pasos para utilizar Serverless Framework para habilitar Lambda Insights en una función de Lambda existente. Para obtener más información acerca de Serverless Framework, consulte serverless.com.

Esto se hace a través de un complemento Lambda Insights para Serverless. Para obtener más información, consulte serverless-plugin-lambda-insights.

Si aún no tiene instalada la versión más reciente de la interfaz de línea de comandos Serverless, primero debe instalarla o actualizarla. Para obtener más información, consulte Introducción al código abierto del marco sin servidor y AWS.

Para utilizar Serverless Framework para habilitar Lambda Insights en una función de Lambda

1. Ejecute el siguiente comando en su directorio Serverless para instalar el complemento Serverless para Lambda Insights:

```
npm install --save-dev serverless-plugin-lambda-insights
```

2. En el archivo `serverless.yml`, agregue el complemento en la sección `plugins` como se muestra:

```
provider:
  name: aws
plugins:
  - serverless-plugin-lambda-insights
```

3. Habilitación de Lambda Insights.
   - Puede habilitar Lambda Insights individualmente por función al agregar la siguiente propiedad al archivo serverless.yml

```
functions:
  myLambdaFunction:
    handler: src/app/index.handler
    lambdaInsights: true #enables Lambda Insights for this function
```

   - Puede habilitar Lambda Insights para todas las funciones dentro del archivo `serverless.yml` si agrega la siguiente sección personalizada:

```
custom:
  lambdaInsights:
    defaultLambdaInsights: true #enables Lambda Insights for all functions
```

4. Vuelva a implementar el servicio Serverless con el siguiente comando:

```
serverless deploy
```

Esto vuelve a implementar todas las funciones y habilita Lambda Insights para las funciones que haya especificado. Habilita Lambda Insights al agregar la capa de Lambda Insights y al adjuntar los permisos necesarios mediante la política de IAM `arn:aws:iam::aws:policy/CloudWatchLambdaInsightsExecutionRolePolicy`.

Habilitación de Lambda Insights en una implementación de imágenes de contenedor de Lambda

Para habilitar Lambda Insights en una función de Lambda que se implementa como una imagen de contenedor, agregue líneas en el Dockerfile. Estas líneas instalan el agente Lambda Insights como una extensión en la imagen de contenedor. Las líneas que se van a añadir son diferentes para los contenedores x86-64 y los contenedores ARM64.

> **Note**
>
> El agente Lambda Insights solo se admite en los tiempos de ejecución de Lambda que utilizan Amazon Linux 2.

Temas

- Implementación de imágenes de contenedores x86-64
- Implementación de imágenes de contenedores ARM64

Implementación de imágenes de contenedores x86-64

Para habilitar Lambda Insights en una función de Lambda que se implementa como una imagen de contenedor que se ejecuta en un contenedor x86-64, agregue las siguientes líneas en el Dockerfile. Estas líneas instalan el agente Lambda Insights como una extensión en la imagen de contenedor.

```
RUN curl -O https://lambda-insights-extension.s3-ap-northeast-1.amazonaws.com/
amazon_linux/lambda-insights-extension.rpm && \
    rpm -U lambda-insights-extension.rpm && \
    rm -f lambda-insights-extension.rpm
```

Después de crear la función de Lambda, asigne la política de IAM CloudWatchLambdaInsightsExecutionRolePolicy al rol de ejecución de la función; Lambda Insights se habilita en la función de Lambda basada en imágenes de contenedor.

> **Note**
>
> Para utilizar una versión anterior de la extensión de Lambda Insights, reemplace la URL de los comandos anteriores por esta URL: `https://lambda-insights-extension.s3-ap-northeast-1.amazonaws.com/amazon_linux/lambda-insights-extension.1.0.111.0.rpm`. Actualmente, solo están disponibles la versión 1.0.111.0 y las versiones posteriores de Lambda Insights. Para obtener más información, consulte Versiones disponibles de la extensión de Lambda Insights.

Para verificar la firma del paquete de agente de Lambda Insights en un servidor Linux

1. Ingrese el siguiente comando para descargar la clave pública.

```
shell$ wget https://lambda-insights-extension.s3-ap-northeast-1.amazonaws.com/
lambda-insights-extension.gpg
```

2. Ingrese el siguiente comando para importar la clave pública en el archivo de claves.

```
shell$ gpg --import lambda-insights-extension.gpg
```

   El resultado será similar al siguiente. Tome nota del valor de `key`, ya que lo necesitará en el siguiente paso. En este resultado de ejemplo, el valor de clave es `848ABDC8`.

```
gpg: key 848ABDC8: public key "Amazon Lambda Insights Extension" imported
gpg: Total number processed: 1
gpg: imported: 1  (RSA: 1)
```

3. Compruebe la huella digital mediante el siguiente comando. Reemplace `key-value` con el valor de clave del paso anterior.

```
shell$  gpg --fingerprint key-value
```

La cadena de huellas digitales en el resultado de este comando debe ser `E0AF FA11 FFF3 5BD7 349E E222 479C 97A1 848A BDC8`. Si la cadena de huellas digitales no concuerda, no instale el agente y contáctese con AWS.

4. Después de haber verificado la huella digital, puede utilizarla para verificar la firma del paquete del agente de Lambda Insights. Ingrese el siguiente comando para descargar el paquete de archivo SIGNATURE.

```
shell$  wget https://lambda-insights-extension.s3-ap-northeast-1.amazonaws.com/
amazon_linux/lambda-insights-extension.rpm.sig
```

5. Para verificar la firma, ejecute el siguiente comando:

```
shell$ gpg --verify lambda-insights-extension.rpm.sig lambda-insights-extension.rpm
```

El resultado debe tener el siguiente aspecto:

```
gpg: Signature made Thu 08 Apr 2021 06:41:00 PM UTC using RSA key ID 848ABDC8
gpg: Good signature from "Amazon Lambda Insights Extension"
gpg: WARNING: This key is not certified with a trusted signature!
gpg:          There is no indication that the signature belongs to the owner.
Primary key fingerprint: E0AF FA11 FFF3 5BD7 349E  E222 479C 97A1 848A BDC8
```

En el resultado esperado, puede haber una advertencia sobre una firma de confianza. Una clave solo es de confianza si la ha firmado usted o alguien en quien confíe. Esto no significa que la firma no sea válida, solo que no ha verificado la clave pública.

Si el resultado contiene `BAD signature`, verifique si ha realizado los pasos correctamente. Si sigue recibiendo una respuesta `BAD signature`, contáctese con AWS y evite usar el archivo descargado.

Ejemplo de x86-64

Esta sección incluye un ejemplo de habilitación de Lambda Insights en una función Python de Lambda basada en imágenes de contenedor.

Un ejemplo de habilitación de Lambda Insights en una imagen de contenedor de Lambda

1. Cree un archivo Dockerfile similar al que se muestra a continuación:

```
FROM public.ecr.aws/lambda/python:3.8

// extra lines to install the agent here
RUN curl -O https://lambda-insights-extension.s3-ap-northeast-1.amazonaws.com/
amazon_linux/lambda-insights-extension.rpm && \
    rpm -U lambda-insights-extension.rpm && \
    rm -f lambda-insights-extension.rpm

COPY index.py ${LAMBDA_TASK_ROOT}
CMD [ "index.handler" ]
```

2. Cree un archivo Python denominado `index.py` que es similar al que se muestra a continuación:

```
def handler(event, context):
  return {
    'message': 'Hello World!'
  }
```

3. Coloque el archivo Dockerfile y `index.py` en el mismo directorio. A continuación, en ese directorio, ejecute los siguientes pasos para crear la imagen de Docker y cargarla en Amazon ECR.

```
// create an ECR repository
aws ecr create-repository --repository-name test-repository
// build the docker image
docker build -t test-image .
// sign in to AWS
aws ecr get-login-password | docker login --username AWS --password-stdin
 "${ACCOUNT_ID}".dkr.ecr."${REGION}".amazonaws.com
// tag the image
docker tag test-image:latest "${ACCOUNT_ID}".dkr.ecr."${REGION}".amazonaws.com/
test-repository:latest
// push the image to ECR
docker push "${ACCOUNT_ID}".dkr.ecr."${REGION}".amazonaws.com/test-
repository:latest
```

4. Utilice la imagen de Amazon ECR que acaba de crear para crear la función de Lambda.
5. Asigne la política de IAM CloudWatchLambdaInsightsExecutionRolePolicy para el rol de ejecución de la función.

Implementación de imágenes de contenedores ARM64

Para habilitar Lambda Insights en una función de Lambda que se implementa como una imagen de contenedor que se ejecuta en un contenedor AL2_aarch64 (que utiliza arquitectura ARM64), añada las siguientes líneas en el Dockerfile. Estas líneas instalan el agente Lambda Insights como una extensión en la imagen de contenedor.

```
RUN curl -O https://lambda-insights-extension-arm64.s3-ap-northeast-1.amazonaws.com/
amazon_linux/lambda-insights-extension-arm64.rpm && \
    rpm -U lambda-insights-extension-arm64.rpm && \
    rm -f lambda-insights-extension-arm64.rpm
```

Después de crear la función de Lambda, asigne la política de IAM CloudWatchLambdaInsightsExecutionRolePolicy al rol de ejecución de la función; Lambda Insights se habilita en la función de Lambda basada en imágenes de contenedor.

> **Note**
>
> Para utilizar una versión anterior de la extensión de Lambda Insights, reemplace la URL de los comandos anteriores por esta URL: `https://lambda-insights-extension-arm64.s3-ap-northeast-1.amazonaws.com/amazon_linux/lambda-insights-extension-arm64.1.0.229.0.rpm`. Actualmente, solo están disponibles la versión 1.0.229.0 y las versiones posteriores de Lambda Insights. Para obtener más información, consulte Versiones disponibles de la extensión de Lambda Insights.

Para verificar la firma del paquete de agente de Lambda Insights en un servidor Linux

1. Ingrese el siguiente comando para descargar la clave pública.

   ```
   shell$ wget https://lambda-insights-extension-arm64.s3-ap-
   northeast-1.amazonaws.com/lambda-insights-extension.gpg
   ```

2. Ingrese el siguiente comando para importar la clave pública en el archivo de claves.

   ```
   shell$ gpg --import lambda-insights-extension.gpg
   ```

   El resultado será similar al siguiente. Tome nota del valor de `key`, ya que lo necesitará en el siguiente paso. En este resultado de ejemplo, el valor de clave es `848ABDC8`.

```
gpg: key 848ABDC8: public key "Amazon Lambda Insights Extension" imported
gpg: Total number processed: 1
gpg: imported: 1  (RSA: 1)
```

3. Compruebe la huella digital mediante el siguiente comando. Reemplace `key-value` con el valor de clave del paso anterior.

```
shell$  gpg --fingerprint key-value
```

   La cadena de huellas digitales en el resultado de este comando debe ser `E0AF FA11 FFF3 5BD7 349E E222 479C 97A1 848A BDC8`. Si la cadena de huellas digitales no concuerda, no instale el agente y contáctese con AWS.

4. Después de haber verificado la huella digital, puede utilizarla para verificar la firma del paquete del agente de Lambda Insights. Ingrese el siguiente comando para descargar el paquete de archivo SIGNATURE.

```
shell$  wget https://lambda-insights-extension-arm64.s3-ap-
northeast-1.amazonaws.com/amazon_linux/lambda-insights-extension-arm64.rpm.sig
```

5. Para verificar la firma, ejecute el siguiente comando:

```
shell$ gpg --verify lambda-insights-extension-arm64.rpm.sig lambda-insights-
extension-arm64.rpm
```

   El resultado debe tener el siguiente aspecto:

```
gpg: Signature made Thu 08 Apr 2021 06:41:00 PM UTC using RSA key ID 848ABDC8
gpg: Good signature from "Amazon Lambda Insights Extension"
gpg: WARNING: This key is not certified with a trusted signature!
gpg:          There is no indication that the signature belongs to the owner.
Primary key fingerprint: E0AF FA11 FFF3 5BD7 349E  E222 479C 97A1 848A BDC8
```

   En el resultado esperado, puede haber una advertencia sobre una firma de confianza. Una clave solo es de confianza si la ha firmado usted o alguien en quien confíe. Esto no significa que la firma no sea válida, solo que no ha verificado la clave pública.

Si el resultado contiene `BAD signature`, verifique si ha realizado los pasos correctamente. Si sigue recibiendo una respuesta `BAD signature`, contáctese con AWS y evite usar el archivo descargado.

Ejemplo de ARM64

Esta sección incluye un ejemplo de habilitación de Lambda Insights en una función Python de Lambda basada en imágenes de contenedor.

Un ejemplo de habilitación de Lambda Insights en una imagen de contenedor de Lambda

1. Cree un archivo Dockerfile similar al que se muestra a continuación:

```
FROM public.ecr.aws/lambda/python:3.8
// extra lines to install the agent here
RUN curl -O https://lambda-insights-extension-arm64.s3-ap-
northeast-1.amazonaws.com/amazon_linux/lambda-insights-extension-arm64.rpm && \
    rpm -U lambda-insights-extension-arm64.rpm && \
    rm -f lambda-insights-extension-arm64.rpm

COPY index.py ${LAMBDA_TASK_ROOT}
CMD [ "index.handler" ]
```

2. Cree un archivo Python denominado `index.py` que es similar al que se muestra a continuación:

```
def handler(event, context):
  return {
    'message': 'Hello World!'
  }
```

3. Coloque el archivo Dockerfile y `index.py` en el mismo directorio. A continuación, en ese directorio, ejecute los siguientes pasos para crear la imagen de Docker y cargarla en Amazon ECR.

```
// create an ECR repository
aws ecr create-repository --repository-name test-repository
// build the docker image
docker build -t test-image .
// sign in to AWS
aws ecr get-login-password | docker login --username AWS --password-stdin
 "${ACCOUNT_ID}".dkr.ecr."${REGION}".amazonaws.com
```

```
// tag the image
docker tag test-image:latest "${ACCOUNT_ID}".dkr.ecr."${REGION}".amazonaws.com/
test-repository:latest
// push the image to ECR
docker push "${ACCOUNT_ID}".dkr.ecr."${REGION}".amazonaws.com/test-
repository:latest
```

4. Utilice la imagen de Amazon ECR que acaba de crear para crear la función de Lambda.
5. Asigne la política de IAM CloudWatchLambdaInsightsExecutionRolePolicy para el rol de ejecución de la función.

Visualización de las métricas de Lambda Insights

Después de instalar la extensión de Lambda Insights en una función de Lambda que se ha invocado, puede utilizar la consola CloudWatch para ver las métricas. Puede ver la información general multifunción o centrarse en una sola función.

Para obtener una lista de las métricas de Lambda Insights, consulte Métricas que Lambda Insights recopila.

Para ver la información general multifunción de las métricas de Lambda Insights

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación izquierdo, en Lambda Insights, elija Multi-function (Multifunción).

   En la parte superior de la página se muestran gráficos con métricas agregadas de todas las funciones de Lambda en la Región que tiene Lambda Insights habilitada. En la parte inferior de la página se muestra una tabla que enumera las funciones.
3. Para filtrar las funciones por nombre y reducir el número de funciones que se muestran, escriba parte del nombre de la función en el cuadro que está situado cerca de la parte superior de la página.
4. Para agregar esta vista a un panel como widget, elija Add to dashboard (Agregar al panel).

Para ver las métricas de una sola función

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación izquierdo, en Lambda Insights, elija Single-function (Ver solo una función).

En la parte superior de la página se muestran gráficos con métricas para la función seleccionada.

3. Si tiene habilitado X-Ray, puede elegir un único ID de rastreo. Esto abre la página del Mapa de seguimiento de X-Ray para esa invocación, y desde allí puede alejarse para ver el rastro distribuido y los otros servicios involucrados en el manejo de esa transacción específica. Para obtener más información sobre el Mapa de seguimiento de X-Ray, consulte Uso del Mapa de seguimiento de X-Ray.
4. Para abrir CloudWatch Logs Insights y ampliar un error específico, elija View logs (Ver registros) en la tabla situada en la parte inferior de la página.
5. Para agregar esta vista a un panel como widget, elija Add to dashboard Agregar al panel.

Integración con Información de aplicaciones

Información de aplicaciones de Amazon CloudWatch ayuda a supervisar las aplicaciones e identifica y configura las métricas clave, los registros y las alarmas entre los recursos de aplicaciones y la pila de tecnología. Para obtener más información, consulte Información de aplicaciones de Amazon CloudWatch.

Puede habilitar Información de aplicaciones para recopilar datos adicionales de las funciones Lambda. Si aún no lo ha hecho, puede habilitarlo al seleccionar Configuración automática de Información de aplicaciones en la pestaña Información de aplicaciones debajo de la vista de rendimiento en el panel de Información de contenedores.

Si ya ha configurado Información de aplicaciones de CloudWatch para supervisar las aplicaciones en contenedores, el panel de Información de aplicaciones aparece debajo del panel de Información de contenedores.

Métricas que Lambda Insights recopila

Lambda Insights recopila algunas métricas de las funciones de Lambda en donde está instalada. Algunas de estas métricas están disponibles como datos de serie temporal agregados en CloudWatch Metrics. Otras métricas no se agregan en datos de serie temporal, pero se pueden encontrar en las entradas de registro con formato de métricas integradas mediante CloudWatch Logs Insights.

Las siguientes métricas están disponibles como datos de serie temporal agregados en CloudWatch Metrics en el espacio de nombres `LambdaInsights`.

Nombre de métrica	Dimensiones	Descripción
`cpu_total_time`	function_name function_name, versión	Suma de `cpu_system_time` y `cpu_user_time`. Unidad: milisegundos
`init_duration`	function_name function_name, versión	La cantidad de tiempo empleado en la fase `init` del ciclo de vida del entorno de ejecución de Lambda. Unidad: milisegundos
`memory_utilization`	function_name function_name, versión	La memoria máxima medida como porcentaje de la memoria asignada a la función. Unidad: porcentaje
`rx_bytes`	function_name function_name, versión	Cantidad de bytes que la función recibe. Unidades: bytes
`tmp_used`		La cantidad de espacio utilizado en el directorio `/tmp`. Unidades: bytes
`tx_bytes`	function_name function_name, versión	Cantidad de bytes que la función envía. Unidades: bytes

Nombre de métrica	Dimensiones	Descripción
`total_memory`	function_name function_name, versión	La cantidad de memoria que se asigna a la función de Lambda. Es lo mismo que el tamaño de la memoria de la función. Unidades: megabytes
`total_network`	function_name function_name, versión	Suma de `rx_bytes` y `tx_bytes`. Incluso para las funciones que no realizan tareas E/S, este valor suele ser mayor que cero debido a las llamadas a la red que el tiempo de ejecución de Lambda realiza. Unidades: bytes
`used_memory_max`	function_name function_name, versión	La memoria medida del entorno de pruebas de la función. Unidades: megabytes

Puede encontrar las siguientes métricas en las entradas de registro con formato de métricas integradas mediante CloudWatch Logs Insights. Para obtener más información acerca de CloudWatch Logs Insights, consulte Analyzing Log Data with CloudWatch Logs Insights (Análisis de datos de registro con CloudWatch Logs Insights).

Para obtener más información sobre el formato de métrica integrado, consulte Incrustar métricas en los registros.

Nombre de métrica	Descripción
`cpu_system_time`	La cantidad de tiempo que la CPU dedicó a ejecutar el código del kernel. Unidad: milisegundos
`cpu_total_time`	Suma de `cpu_system_time` y `cpu_user_time` . Unidad: milisegundos
`cpu_user_time`	La cantidad de tiempo que la CPU dedicó a ejecutar el código de usuario. Unidad: milisegundos
`fd_max`	El número máximo de descriptores de archivo disponibles. Unidad: recuento
`fd_use`	El número máximo de descriptores de archivo en uso. Unidad: recuento
`memory_utilization`	La memoria máxima medida como un porcentaje de la memoria asignada a la función. Unidad: porcentaje
`rx_bytes`	El número de bytes que la función recibe. Unidades: bytes
`tx_bytes`	El número de bytes que la función envía. Unidades: bytes

Nombre de métrica	Descripción
`threads_max`	El número de subprocesos que el proceso de la función utiliza. Como autor de funciones, usted no controla el número inicial de subprocesos que el tiempo de ejecución crea. Unidad: recuento
`tmp_max`	La cantidad de espacio disponible en el directorio `/tmp`. Unidades: bytes
`total_memory`	La cantidad de memoria que se asigna a la función de Lambda. Es lo mismo que el tamaño de la memoria de la función. Unidades: megabytes
`total_network`	Suma de `rx_bytes` y `tx_bytes`. Incluso para las funciones que no realizan tareas E/S, este valor suele ser mayor que cero debido a las llamadas a la red que el tiempo de ejecución de Lambda realiza. Unidades: bytes
`used_memory_max`	La memoria medida del entorno de pruebas de la función. Unidades: bytes

Solución de errores y problemas conocidos

El primer paso para solucionar cualquier error es habilitar el registro de depuración en la extensión de Lambda Insights. Para ello, configure la siguiente variable de entorno en la función de Lambda: `LAMBDA_INSIGHTS_LOG_LEVEL=info`. Para obtener más información, consulte Uso de variables de entorno de AWS Lambda.

La extensión emite registros en el mismo grupo de registros que la función (`/aws/lambda/function-name`). Revise esos registros para ver si el error está relacionado con un problema de instalación.

No se ve ninguna métrica desde Lambda Insights

Si no se muestran las métricas de Lambda Insights que espera ver, verifique las siguientes posibilidades:

- Es posible que las métricas se retrasen: si la función aún no se ha invocado o los datos aún no se han vaciado, no verá las métricas en CloudWatch. Para obtener más información, consulte Known Issues (Problemas conocidos) más adelante en esta sección.
- Verifique que la función de Lambda tenga los permisos que correspondenAsegúrese de que la política de IAM CloudWatchLambdaInsightsExecutionRolePolicy se asigne al rol de ejecución de la función.
- Verifique el tiempo de ejecución de Lambda: Lambda Insights solo admite ciertos tiempos de ejecución de Lambda. Para obtener una lista de los tiempos de ejecución admitidos, consulte Lambda Insights.

  Por ejemplo, para usar Lambda Insights en Java 8, debe usar el tiempo de ejecución `java8.al2`, no el tiempo de ejecución `java8`.
- Verifique el acceso a la red: la función de Lambda puede estar en una subred privada de la VPC sin acceso a Internet y usted no tiene un punto de enlace de la VPC configurado para CloudWatch Logs. Para ayudar a depurar este problema, puede establecer la variable de entorno `LAMBDA_INSIGHTS_LOG_LEVEL=info`.

Problemas conocidos

El retraso de los datos puede ser de hasta 20 minutos. Cuando se completa un controlador de funciones, Lambda congela el entorno de prueba, que también congela la extensión de Lambda Insights. Mientras la función se está ejecutando, se utiliza una estrategia de procesamiento por lotes adaptable basada en la función TPS para generar datos. Sin embargo, si la función deja de invocarse durante un período prolongado y todavía hay datos de eventos en el búfer, estos datos pueden retrasarse hasta que Lambda cierre el entorno de pruebas. Cuando Lambda cierra el entorno de pruebas, vaciamos los datos almacenados en el búfer.

Evento de telemetría de ejemplo

Cada invocación de una función de Lambda que tiene Lambda Insights habilitado registra un único evento de registro en el grupo de registro `/aws/lambda-insights`. Cada evento de registro contiene métricas en formato de métricas integradas. Para obtener más información sobre el formato de métrica integrado, consulte Incrustar métricas en los registros.

Para analizar los eventos de registro, puede utilizar los siguientes métodos:

- La sección Lambda Insights de la consola CloudWatch, como se explica en Visualización de las métricas de Lambda Insights.
- Consultas de eventos de registro mediante CloudWatch Logs Insights. Para obtener más información, consulte Analizar datos de registro con CloudWatch Logs Insights.
- Métricas recopiladas en el espacio de nombres `LambdaInsights`, que se grafican mediante las métricas de CloudWatch.

A continuación se muestra un ejemplo de un evento de registro de Lambda Insights con el formato de métricas integradas.

```
{
    "_aws": {
        "Timestamp": 1605034324256,
        "CloudWatchMetrics": [
            {
                "Namespace": "LambdaInsights",
                "Dimensions": [
                    [ "function_name" ],
                    [ "function_name", "version" ]
                ],
                "Metrics": [
                    { "Name": "memory_utilization", "Unit": "Percent" },
                    { "Name": "total_memory", "Unit": "Megabytes" },
                    { "Name": "used_memory_max", "Unit": "Megabytes" },
                    { "Name": "cpu_total_time", "Unit": "Milliseconds" },
                    { "Name": "tx_bytes", "Unit": "Bytes" },
                    { "Name": "rx_bytes", "Unit": "Bytes" },
                    { "Name": "total_network", "Unit": "Bytes" },
                    { "Name": "init_duration", "Unit": "Milliseconds" }
                ]
            }
```

```
        ],
        "LambdaInsights": {
            "ShareTelemetry": true
        }
    },
    "event_type": "performance",
    "function_name": "cpu-intensive",
    "version": "Blue",
    "request_id": "12345678-8bcc-42f7-b1de-123456789012",
    "trace_id": "1-5faae118-12345678901234567890",
    "duration": 45191,
    "billed_duration": 45200,
    "billed_mb_ms": 11571200,
    "cold_start": true,
    "init_duration": 130,
    "tmp_free": 538329088,
    "tmp_max": 551346176,
    "threads_max": 11,
    "used_memory_max": 63,
    "total_memory": 256,
    "memory_utilization": 24,
    "cpu_user_time": 6640,
    "cpu_system_time": 50,
    "cpu_total_time": 6690,
    "fd_use": 416,
    "fd_max": 32642,
    "tx_bytes": 4434,
    "rx_bytes": 6911,
    "timeout": true,
    "shutdown_reason": "Timeout",
    "total_network": 11345,
    "agent_version": "1.0.72.0",
    "agent_memory_avg": 10,
    "agent_memory_max": 10
}
```

Uso de Información de colaboradores para analizar datos de alta cardinalidad

Puede utilizar Contributor Insights para analizar datos de registro y crear series temporales que muestren datos de colaboradores. Puede ver métricas acerca de los colaboradores Top-N, el número total de colaboradores únicos y su uso. Esto le ayuda a encontrar a los hablantes más activos y

a comprender quién o qué afecta al rendimiento del sistema. Por ejemplo, puede encontrar hosts incorrectos, identificar a los usuarios de red más frecuentes o encontrar las URL que más errores generan.

Puede crear reglas desde cero y, cuando utilice la AWS Management Console, también puede usar reglas de ejemplo creadas por AWS. Las reglas definen los campos de registro que desea utilizar para definir colaboradores, como `IpAddress`. También puede filtrar los datos de registro para buscar y analizar el comportamiento de los colaboradores individuales.

CloudWatch también proporciona reglas integradas que se pueden utilizar para analizar métricas de otros servicios de AWS.

Todas las reglas analizan los datos de entrada en tiempo real.

Si ha iniciado sesión en una cuenta que está configurada como cuenta de monitoreo en la observabilidad entre cuentas de CloudWatch, puede crear reglas de Información de colaboradores en esa cuenta de monitoreo que analicen los grupos de registro en las cuentas de origen y en la cuenta de monitoreo. También puede crear una regla única que analice los grupos de registro de varias cuentas. Para obtener más información, consulte Observabilidad entre cuentas de CloudWatch.

> **Note**
>
> Si utiliza Contributor Insights, se le cobra por cada aparición de un evento de registro que coincida con una regla. Para obtener más información, consulte Precios de Amazon CloudWatch.

Temas

- Creación de una regla de Información de colaboradores
- Sintaxis de regla de Contributor Insights
- Ejemplos de regla de Contributor Insights
- Visualización de informes de Contributor Insights
- Representación gráfica de métricas generadas por reglas
- Uso de reglas integradas de Contributor Insights

Creación de una regla de Información de colaboradores

Puede crear reglas para analizar los datos de registro. Se pueden evaluar todos los registros en JSON o Common Log Format (CLF). Esto incluye los registros personalizados que siguen uno de estos formatos y los registros de AWS como los registros de flujo de Amazon VPC, los registros de consultas de DNS de Amazon Route 53, los registros de contenedores de Amazon ECS y los registros de AWS CloudTrail, Amazon SageMaker, Amazon RDS, AWS AppSync y API Gateway.

En una regla, al especificar valores o nombres de campo, todas las coincidencias distinguen entre mayúsculas y minúsculas.

Al crear una regla, puede utilizar reglas de ejemplo integradas o bien crear su propia regla desde cero. Contributor Insights incluye reglas de muestra para los siguientes tipos de registros:

- Registros de Amazon API Gateway
- Registros de consultas de DNS públicas de Amazon Route 53
- Registros de consultas de Amazon Route 53 Resolver
- Registros de CloudWatch Container Insights
- Logs de flujo de VPC

Si ha iniciado sesión en una cuenta que está configurada como cuenta de monitoreo en la observabilidad entre cuentas de CloudWatch, puede crear reglas de Información de colaboradores para los grupos de registro de las cuentas de origen que estén vinculadas a la cuenta de monitoreo, además de crear reglas para los grupos de registro de la cuenta de monitoreo. También puede configurar una sola regla que supervise los grupos de registro en diferentes cuentas. Para obtener más información, consulte Observabilidad entre cuentas de CloudWatch.

> **Important**
>
> Cuando concede a un usuario el permiso `cloudwatch:PutInsightRule`, de forma predeterminada, ese usuario puede crear una regla que evalúe cualquier grupo de registros en CloudWatch Logs. Puede agregar condiciones de políticas de IAM que limiten estos permisos para que un usuario incluya y excluya grupos de registros específicos. Para obtener más información, consulte Uso de claves de condición para limitar el acceso de los usuarios de Contributor Insights a los grupos de registro.

Para crear una regla con una regla de ejemplo integrada

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Insights y, luego, Contributor Insights.
3. Elija Crear regla.
4. En Select log group(s), seleccione los grupos de registro que quiera que la regla supervise. Puede seleccionar hasta 20 grupos de registro. Si ha iniciado sesión en una cuenta de monitoreo que esté configurada para la observabilidad entre cuentas de CloudWatch, puede seleccionar grupos de registro en las cuentas de origen y también puede configurar una regla única para analizar los grupos de registro de diferentes cuentas.
   - (Opcional) Para seleccionar todos los grupos de registros que tienen nombres que empiezan por una cadena específica, elija el menú desplegable Select by prefix match (Seleccionar por concordancia de prefijo) y, a continuación, introduzca el prefijo. Si se trata de una cuenta de monitoreo, tiene la opción de seleccionar las cuentas en las que quiera buscar; de lo contrario, se seleccionarán todas las cuentas.

   > **Note**
   >
   > Incurrirá en cargos por cada evento de registro que coincida con su regla. Si elige el menú desplegable Select by prefix match (Seleccionar por concordancia de prefijo), tenga en cuenta cuántos grupos de registros puede coincidir el prefijo. Si busca más grupos de registro de los que pretendía, puede incurrir en cargos inesperados. Para obtener más información, consulte Precios de Amazon CloudWatch.
5. En Rule type (Tipo de regla), elija Sample rule (Regla de ejemplo). A continuación, elija Select sample rule (Seleccionar regla de ejemplo) y seleccione la regla.
6. La regla de ejemplo ha rellenado los campos Log format (Formato de registro), Contribution (Contribución), Filters (Filtros) y Aggregate on (Agregar según). Si lo desea, puede ajustar los valores.
7. Elija Siguiente.
8. Escriba un nombre en Rule name (Nombre de la regla). Los caracteres válidos son A-Z, a-z, 0-9, (guion), (guion bajo) y (punto).

9. Elija si desea crear una regla deshabilitada o habilitada. Si elige habilitarla, la regla comenzará a analizar sus datos inmediatamente. Cuando se ejecutan reglas habilitadas, se incurre en costos. Para obtener más información, consulte Precios de Amazon CloudWatch.

   Contributor Insights analiza solo eventos de registro nuevos una vez creada una regla. Una regla no puede procesar eventos de registros que CloudWatch Logs haya procesado anteriormente.

10. (Opcional) En Tags (Etiquetas), agregue uno o más pares de clave-valor como etiquetas para esta regla. Las etiquetas pueden ayudarle a identificar y organizar sus recursos de AWS y a realizar un seguimiento de sus costos de AWS. Para obtener más información, consulte Etiquetado de los recursos de Amazon CloudWatch.
11. Seleccione Crear.

Para crear una regla desde cero

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Contributor Insights.
3. Elija Crear regla.

4. En Select log group(s), seleccione los grupos de registro que quiera que la regla supervise. Puede seleccionar hasta 20 grupos de registro. Si ha iniciado sesión en una cuenta de monitoreo que esté configurada para la observabilidad entre cuentas de CloudWatch, puede seleccionar grupos de registro en las cuentas de origen y también puede configurar una regla única para analizar los grupos de registro de diferentes cuentas.

   - (Opcional) Para seleccionar todos los grupos de registros que tienen nombres que empiezan por una cadena específica, elija el menú desplegable Select by prefix match (Seleccionar por concordancia de prefijo) y, a continuación, introduzca el prefijo.

> **Note**
>
> Incurrirá en cargos por cada evento de registro que coincida con su regla. Si elige el menú desplegable Select by prefix match (Seleccionar por concordancia de prefijo), tenga en cuenta cuántos grupos de registros puede coincidir el prefijo. Si busca más grupos de registro de los que pretendía, puede incurrir en cargos inesperados. Para obtener más información, consulte Precios de Amazon CloudWatch.

5. En Rule type, elija Custom Rule (Regla personalizada).
6. En Log format (Formato de registro), elija JSON o CLF.
7. Para terminar de crear la regla, utilice el asistente o elija la pestaña Syntax (Sintaxis) y especifique la sintaxis de la regla manualmente.

   Para seguir utilizando el asistente, haga lo siguiente:

   a. En Contribution (Contribución) y Key (Clave), especifique un tipo de colaborador del que desee informar. En el informe se muestran los N valores principales de este tipo de colaborador.

      Las entradas válidas son cualquier campo de registro que tenga valores. Entre los ejemplos se incluyen **`requestId`**, **`sourceIPaddress`** y **`containerID`**.

      Para obtener información acerca de cómo encontrar los nombres de los campos de registro para los registros de un determinado grupo de estos, consulte Buscar campos de registro.

      Las claves superiores a 1 KB se truncan en 1 KB.

   b. (Opcional) Seleccione Add new key (Agregar nueva clave) para agregar más claves. Puede incluir hasta cuatro claves en una regla. Si especifica más de una clave, las combinaciones de valores únicas de las claves definen a los colaboradores del informe. Por ejemplo, si especifica tres claves, cada combinación única de valores de las tres claves se contará como un colaborador único.

   c. (Opcional) Si desea agregar un filtro que limite los resultados, elija Add filter (Agregar filtro). En Match (Coincidencia), ingrese el nombre del campo de registro en el que desea aplicar el filtro. En Condition (Condición), elija el operador de comparación e ingrese un valor por el cual desee filtrar este campo.

      Puede añadir hasta cuatro filtros en una regla. La lógica AND une varios filtros, por lo que solo se evalúan los eventos de registro que coincidan con todos los filtros.

      > **Note**
      >
      > Las matrices que siguen los operadores de comparación, tales como `In`, `NotIn` o `StartsWith`, pueden incluir hasta 10 valores de cadena. Para obtener más información acerca de la sintaxis de las reglas de Contributor Insights, consulte Sintaxis de regla de Contributor Insights.

   d. En Aggregate on (Agregar según), elija Count (Recuento) o Sum (Suma). Si se elige Count (Recuento), la clasificación de los colaboradores se basará en el número de apariciones. Si se elige Sum (Suma), la clasificación se basará en la suma añadida de los valores del campo que especifique para Contribution (Contribución) y Value (Valor).

8. Para especificar su regla como objeto JSON en lugar de utilizar el asistente, haga lo siguiente:

   a. Elija la pestaña Syntax (Sintaxis).

   b. En Rule body (Cuerpo de reglas), especifique el objeto JSON para su regla. Para obtener información sobre la sintaxis de regla, consulte Sintaxis de regla de Contributor Insights.

9. Elija Siguiente.

10. Escriba un nombre en Rule name (Nombre de la regla). Los caracteres válidos son A-Z, a-z, 0-9, "-", "_' y ".".

11. Elija si desea crear una regla deshabilitada o habilitada. Si elige habilitarla, la regla comenzará a analizar sus datos inmediatamente. Cuando se ejecutan reglas habilitadas, se incurre en costos. Para obtener más información, consulte Precios de Amazon CloudWatch.

    Contributor Insights analiza solo eventos de registro nuevos una vez creada una regla. Una regla no puede procesar eventos de registros que CloudWatch Logs haya procesado anteriormente.

12. (Opcional) En Tags (Etiquetas), agregue uno o más pares de clave-valor como etiquetas para esta regla. Las etiquetas pueden ayudarle a identificar y organizar sus recursos de AWS y a realizar un seguimiento de sus costos de AWS. Para obtener más información, consulte Etiquetado de los recursos de Amazon CloudWatch.

13. Elija Siguiente.

14. Confirme la configuración que haya introducido y seleccione Create rule (Crear regla).

Puede deshabilitar, habilitar o eliminar las reglas que ha creado.

Para habilitar, deshabilitar o eliminar una regla en Contributor Insights

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.

2. En el panel de navegación, elija Contributor Insights.

3. En la lista de reglas, seleccione la casilla situada junto a una sola regla.

   Los servicios de AWS crean las reglas integradas, que no se pueden editar, deshabilitar ni eliminar.

4. Elija Actions (Acciones) y, a continuación, elija la opción que desee.

Búsqueda de campos de registro

Al crear una regla, debe conocer los nombres de los campos de las entradas de registro de un grupo de registros.

Para buscar los campos de registro en un grupo de registros

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, en Logs (Registros), elija Insights (Conocimientos).
3. Encima del editor de consultas, seleccione uno o varios grupos de registros que va a consultar.

   Al seleccionar un grupo de registros, CloudWatch Logs Insights detecta automáticamente los campos de los datos en el grupo de registros y los muestra en el panel derecho en Discovered fields (Campos detectados).

Sintaxis de regla de Contributor Insights

En esta sección se explica la sintaxis para las reglas de Contributor Insights. Utilice esta sintaxis solo cuando vaya a crear una regla introduciendo un bloque JSON. Si utiliza el asistente para crear una regla, no necesita conocer la sintaxis. Para obtener más información acerca de cómo crear reglas mediante el asistente, consulte Creación de una regla de Información de colaboradores.

Todas las coincidencias de reglas para registrar valores y nombres de campo de evento distinguen entre mayúsculas y minúsculas.

En el ejemplo siguiente se ilustra la sintaxis de los registros JSON.

```
{
    "Schema": {
        "Name": "CloudWatchLogRule",
        "Version": 1
    },
    "LogGroupNames": [
        "API-Gateway-Access-Logs*",
        "Log-group-name2"
    ],
```

```
    "LogFormat": "JSON",
    "Contribution": {
        "Keys": [
            "$.ip"
        ],
        "ValueOf": "$.requestBytes",
        "Filters": [
            {
                "Match": "$.httpMethod",
                "In": [
                    "PUT"
                ]
            }
        ]
    },
    "AggregateOn": "Sum"
}
```

Campos en las reglas de Contributor Insights

Esquema

En el valor de `Schema` para una regla que analiza datos de CloudWatch Logs siempre debe ser `{"Name": "CloudWatchLogRule", "Version": 1}`

LogGroupNames

Una matriz de cadenas. Para cada elemento de la matriz, puede usar opcionalmente * al final de una cadena para incluir todos los grupos de registros con nombres que empiecen por ese prefijo.

Tenga cuidado con el uso de comodines con los nombres de los grupos de registro. Incurrirá en cargos por cada evento de registro que coincida con una regla. Si busca accidentalmente más grupos de registro de los que pretendía, puede incurrir en cargos inesperados. Para obtener más información, consulte Precios de Amazon CloudWatch.

LogGroupARNs

Si está creando esta regla en una cuenta de monitoreo de observabilidad entre cuentas de CloudWatch, puede usar `LogGroupARNs` para especificar grupos de registro en las cuentas de origen que están vinculadas a la cuenta de monitoreo y para especificar grupos de registro en la propia cuenta de monitoreo. Debe especificar `LogGroupNames` o `LogGroupARNs` en la regla, pero no ambos.

`LogGroupARNs` es una matriz de cadenas. Para cada elemento de la matriz, tiene la opción de usar `*` como comodín en determinadas situaciones. Por ejemplo, puede indicar `arn:aws:logs:us-west-1:*:log-group/MyLogGroupName2` para especificar los grupos de registro denominados `MyLogGroupName2` en todas las cuentas de origen y en la cuenta de monitoreo, en la región Oeste de EE. UU. (Norte de California). También puede indicar `arn:aws:logs:us-west-1:111122223333:log-group/GroupNamePrefix*` para especificar todos los grupos de registro del Oeste de EE. UU. (Norte de California) en 111122223333 que tengan nombres que comiencen por `GroupNamePrefix`.

Recuerde que no puede especificar un ID de cuenta parcial (AWS) como prefijo con un comodín.

Use con cuidado los comodines en los ARN de los grupos de registro. Incurrirá en cargos por cada evento de registro que coincida con una regla. Si busca accidentalmente más grupos de registro de los que pretendía, puede incurrir en cargos inesperados. Para obtener más información, consulte Precios de Amazon CloudWatch.

LogFormat

Los valores válidos son `JSON` y `CLF`.

Contribución

Este objeto incluye una matriz `Keys` con hasta cuatro miembros, una solo `ValueOf` de forma opcional y, también opcionalmente, una matriz de hasta cuatro `Filters`.

Claves

Una matriz de hasta cuatro campos de registro que se utilizan como dimensiones para clasificar los colaboradores. Si especifica más de una clave, cada combinación única de valores de las claves se cuenta como un colaborador único. Los campos deben especificarse utilizando la notación de formato de propiedad JSON.

ValueOf

(Opcional) Especifique esto solo cuando especifique `Sum` como valor de `AggregateOn`. `ValueOf` especifica un campo de registro con valores numéricos. En este tipo de regla, los colaboradores se clasifican según su suma del valor de este campo, en lugar de su número de apariciones en las entradas de registro. Por ejemplo, si desea ordenar a los colaboradores por sus valores de `BytesSent` totales durante un período, establezca `ValueOf` en `BytesSent` y especifique `Sum` para `AggregateOn`.

Filtros

(Opcional) Especifica una matriz de hasta cuatro filtros para restringir los eventos de registro que se incluyen en el informe. Si especifica varios filtros, Contributor Insights los evalúa con un operador AND lógico. Puede utilizar esto para filtrar eventos de registro irrelevantes en su búsqueda, o bien para seleccionar un solo colaborador a fin de analizar su comportamiento.

Cada miembro de la matriz debe incluir un campo `Match` y un campo que indique el tipo de operador coincidente que se debe usar.

El campo `Match` especifica un campo de registro para evaluar en el filtro. El campo de registro se especifica mediante la notación de formato de propiedad JSON.

El campo de operador coincidente debe ser uno de los siguientes: `In`, `NotIn`, `StartsWith`, `GreaterThan`, `LessThan`, `EqualTo`, `NotEqualTo` o `IsPresent`. Si el campo de operador es `In`, `NotIn` o `StartsWith`, va seguido de una matriz de valores de cadena que comprobar. Contributor Insights evalúa la matriz de valores de cadena con un operador OR. La matriz puede incluir hasta 10 valores de cadena.

Si el campo de operador es `GreaterThan`, `LessThan`, `EqualTo` o `NotEqualTo`, va seguido de un único valor numérico con el que comparar.

Si el campo de operador es `IsPresent`, va seguido de `true` o `false`. Este operador coincide con los eventos de registro en función de si el campo de registro especificado está presente en el evento de registro. `isPresent` funciona solo con valores en el nodo hijo de propiedades JSON. Por ejemplo, un filtro que busca coincidencias con `c-count` no evalúa un evento de registro con un valor de `details.c-count.c1`.

Vea los cuatro ejemplos de filtro siguientes:

```
{"Match": "$.httpMethod", "In": [ "PUT", ] }
{"Match": "$.StatusCode", "EqualTo": 200 }
{"Match": "$.BytesReceived", "GreaterThan": 10000}
{"Match": "$.eventSource", "StartsWith": [ "ec2", "ecs" ] }
```

AggregateOn

Los valores válidos son `Count` y `Sum`. Especifica si se debe agregar el informe en función de un recuento de apariciones o una suma de los valores del campo que se especifica en el campo `ValueOf`.

Notación de formato de propiedad JSON

Los campos `Keys`, `ValueOf` y `Match` siguen el formato de propiedad JSON con la notación de puntos, donde $ representa la raíz del objeto JSON. Esto va seguido de un punto y, a continuación, de una cadena alfanumérica con el nombre de la subpropiedad. Se admiten varios niveles de propiedad.

El primer carácter de la cadena solo puede ser A-Z o a-z. Los siguientes caracteres de la cadena pueden ser A-Z, a-z o 0-9.

En la lista siguiente se muestran ejemplos válidos de formato de propiedad JSON:

```
$.userAgent
$.endpoints[0]
$.users[1].name
$.requestParameters.instanceId
```

Campo adicional en las reglas de los registros para CLF

Los eventos de registro de Common Log Format (CLF) no tienen nombres para los campos como JSON. Para proporcionar los campos que se utilizarán para las reglas de Contributor Insights, un evento de registro CLF se puede tratar como matriz con un índice a partir de 1. Puede especificar el primer campo como **"1"**, el segundo como **"2"**, y así sucesivamente.

Para facilitar la lectura de una regla de un registro CLF, puede utilizar `Fields`. Esto le permite proporcionar un alias de nombre para las ubicaciones de campo CLF. Por ejemplo, puede especificar que la ubicación "4" es una dirección IP. Una vez especificada, se puede usar `IpAddress` como propiedad en `Keys`, `ValueOf` y `Filters` en la regla.

A continuación, se muestra un ejemplo de una regla de un registro CLF que utiliza el campo `Fields`.

```
{
    "Schema": {
        "Name": "CloudWatchLogRule",
        "Version": 1
    },
    "LogGroupNames": [
        "API-Gateway-Access-Logs*"
    ],
    "LogFormat": "CLF",
    "Fields": {
```

```
        "4": "IpAddress",
        "7": "StatusCode"
    },
    "Contribution": {
        "Keys": [
            "IpAddress"
        ],
        "Filters": [
            {
                "Match": "StatusCode",
                "EqualTo": 200
            }
        ]
    },
    "AggregateOn": "Count"
}
```

Ejemplos de regla de Contributor Insights

Esta sección contiene ejemplos que ilustran casos de uso de las reglas de Contributor Insights.

Registros de flujo de VPC: transferencias de bytes por dirección IP de origen IP y destino

```
{
    "Schema": {
        "Name": "CloudWatchLogRule",
        "Version": 1
    },
    "LogGroupNames": [
        "/aws/containerinsights/sample-cluster-name/flowlogs"
    ],
    "LogFormat": "CLF",
    "Fields": {
        "4": "srcaddr",
        "5": "dstaddr",
        "10": "bytes"
    },
    "Contribution": {
        "Keys": [
            "srcaddr",
            "dstaddr"
        ],
        "ValueOf": "bytes",
```

```
        "Filters": []
    },
    "AggregateOn": "Sum"
}
```

Registros de flujo de VPC: número más alto de solicitudes HTTPS

```
{
    "Schema": {
        "Name": "CloudWatchLogRule",
        "Version": 1
    },
    "LogGroupNames": [
        "/aws/containerinsights/sample-cluster-name/flowlogs"
    ],
    "LogFormat": "CLF",
    "Fields": {
        "5": "destination address",
        "7": "destination port",
        "9": "packet count"
    },
    "Contribution": {
        "Keys": [
            "destination address"
        ],
        "ValueOf": "packet count",
        "Filters": [
            {
                "Match": "destination port",
                "EqualTo": 443
            }
        ]
    },
    "AggregateOn": "Sum"
}
```

Registros de flujo de VPC: conexiones TCP rechazadas

```
{
    "Schema": {
        "Name": "CloudWatchLogRule",
        "Version": 1
    },
```

```
    "LogGroupNames": [
        "/aws/containerinsights/sample-cluster-name/flowlogs"
    ],
    "LogFormat": "CLF",
    "Fields": {
        "3": "interfaceID",
        "4": "sourceAddress",
        "8": "protocol",
        "13": "action"
    },
    "Contribution": {
        "Keys": [
            "interfaceID",
            "sourceAddress"
        ],
        "Filters": [
            {
                "Match": "protocol",
                "EqualTo": 6
            },
            {
                "Match": "action",
                "In": [
                    "REJECT"
                ]
            }
        ]
    },
    "AggregateOn": "Sum"
}
```

Respuestas de Route 53 NxDomain por dirección de origen

```
{
    "Schema": {
        "Name": "CloudWatchLogRule",
        "Version": 1
    },
    "AggregateOn": "Count",
    "Contribution": {
        "Filters": [
            {
                "Match": "$.rcode",
```

```
                "StartsWith": [
                    "NXDOMAIN"
                ]
            }
        ],
        "Keys": [
            "$.srcaddr"
        ]
    },
    "LogFormat": "JSON",
    "LogGroupNames": [
        "<loggroupname>"
    ]
}
```

Consultas de resolución de Route 53 por nombre de dominio

```
{
    "Schema": {
        "Name": "CloudWatchLogRule",
        "Version": 1
    },
    "AggregateOn": "Count",
    "Contribution": {
        "Filters": [],
        "Keys": [
            "$.query_name"
        ]
    },
    "LogFormat": "JSON",
    "LogGroupNames": [
        "<loggroupname>"
    ]
}
```

Consultas de resolución de Route 53 por tipo de consulta y dirección de origen

```
{
    "Schema": {
        "Name": "CloudWatchLogRule",
        "Version": 1
    },
    "AggregateOn": "Count",
```

```
    "Contribution": {
        "Filters": [],
        "Keys": [
            "$.query_type",
            "$.srcaddr"
        ]
    },
    "LogFormat": "JSON",
    "LogGroupNames": [
        "<loggroupname>"
    ]
}
```

Visualización de informes de Contributor Insights

Para ver gráficos de datos de informe y una lista clasificada de colaboradores que se encuentran mediante sus reglas, siga estos pasos.

Para ver los informes de las reglas

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Contributor Insights.
3. En la lista de reglas, elija el nombre de una regla.

   En el gráfico se muestran los resultados de la regla durante las últimas tres horas. En la tabla situada bajo el gráfico se muestran los 10 colaboradores principales.
4. Para cambiar el número de colaboradores que se muestran en la tabla, elija Top 10 contributors (10 colaboradores principales) en la parte superior del gráfico.
5. Para filtrar el gráfico a fin de mostrar solo los resultados de un único colaborador, elija a ese colaborador en la leyenda de la tabla. Para volver a mostrar todos los colaboradores, elija de nuevo a ese mismo colaborador en la leyenda.
6. Para cambiar el intervalo de tiempo que se muestra en el informe, elija 15 min, 30 min, 1 h, 2 h, 3 h o custom (personalizado) en la parte superior del gráfico.

   El intervalo de tiempo máximo del informe es de 24 horas, pero puede elegir una ventana de 24 horas que se produjo hace 15 días como máximo. Para elegir una ventana de tiempo en el pasado, elija custom (personalizada), absolute (absoluta) y, a continuación, especifique la ventana de tiempo.

7. Para cambiar la duración del período de tiempo empleado para la incorporación y clasificación de los colaboradores, elija period (período) en la parte superior del gráfico. La visualización de un período de tiempo más largo suele mostrar un informe más sencillo con pocos picos. Es más probable que se muestren picos si se elige un período de tiempo más corto.
8. Para agregar este gráfico a un panel de CloudWatch, elija Add to dashboard (Agregar al panel).
9. Para abrir la ventana de consulta de CloudWatch Logs Insights, con los grupos de registros de este informe ya cargados en el cuadro de consulta, elija View logs (Ver registros).
10. Para exportar los datos de informe al portapapeles o a un archivo CSV, elija Export (Exportar).

Representación gráfica de métricas generadas por reglas

Contributor Insights proporciona una función de cálculo de métricas, `INSIGHT_RULE_METRIC`. Puede utilizar esta función para agregar datos de un informe de Contributor Insights a un gráfico en la pestaña Metrcis (Métricas) de la consola de CloudWatch. También puede establecer una alarma basada en esta función matemática. Para obtener más información acerca de las funciones de cálculo de métricas, consulte Uso de la calculadora de métricas.

Para utilizar esta función matemática de métricas, debe haber iniciado sesión en una cuenta que tenga los permisos `cloudwatch:GetMetricData` y `cloudwatch:GetInsightRuleReport`.

La sintaxis es `INSIGHT_RULE_METRIC(`*`ruleName`*`, `*`metricName`*`)`. *`ruleName`* es el nombre de una regla de Contributor Insights y *`metricName`* es uno de los valores de la lista siguiente. El valor de *`metricName`* determina qué tipo de datos devuelve la función matemática.

- `UniqueContributors`: el número de colaboradores únicos para cada punto de datos.
- `MaxContributorValue`: el valor del colaborador principal para cada punto de datos. La identidad del colaborador puede cambiar en cada punto de datos del gráfico.

  Si la regla agrega por `Count`, el colaborador principal de cada punto de datos es el colaborador que más veces aparece en ese período. Si la regla agrega por `Sum`, el colaborador principal es aquel con la suma más alta en el campo de registro especificado por el `Value` de la regla durante ese período.
- `SampleCount`: el número de puntos de datos que concuerdan con la regla.
- `Sum`: la suma de los valores de todos los colaboradores durante el periodo de tiempo representado por ese punto de datos.

- `Minimum`: el valor mínimo de una sola observación durante el período de tiempo representado por ese punto de datos.
- `Maximum`: el valor máximo de una sola observación durante el período de tiempo representado por ese punto de datos.
- `Average`: el valor promedio de todos los colaboradores durante el período de tiempo representado por ese punto de datos.

Configuración de una alarma en los datos de métricas de Contributor Insights

Puede configurar alarmas para las métricas generadas por Contributor Insights mediante la función `INSIGHT_RULE_METRIC`. Por ejemplo, podría crear una alarma en función del porcentaje de conexiones de protocolo de control de transmisión (TCP) que se han rechazado. Para empezar a utilizar este tipo de alarma, puede crear reglas como las que se muestran en los dos ejemplos siguientes:

Regla de ejemplo: “RejectedConnectionsRule”

```
{
    "Schema": {
        "Name": "CloudWatchLogRule",
        "Version": 1
    },
    "LogGroupNames": [
        "/aws/containerinsights/sample-cluster-name/flowlogs"
    ],
    "LogFormat": "CLF",
    "Fields": {
        "3": "interfaceID",
        "4": "sourceAddress",
        "8": "protocol",
        "13": "action"
    },
    "Contribution": {
        "Keys": [
            "interfaceID",
            "sourceAddress"
        ],
        "Filters": [
            {
                "Match": "protocol",
```

```
                "EqualTo": 6
            },
            {
                "Match": "action",
                "In": [
                    "REJECT"
                ]
            }
        ]
    },
    "AggregateOn": "Sum"
}
```

Regla de ejemplo: “TotalConnectionsRule”

```
{
    "Schema": {
        "Name": "CloudWatchLogRule",
        "Version": 1
    },
    "LogGroupNames": [
        "/aws/containerinsights/sample-cluster-name/flowlogs"
    ],
    "LogFormat": "CLF",
    "Fields": {
        "3": "interfaceID",
        "4": "sourceAddress",
        "8": "protocol",
        "13": "action"
    },
    "Contribution": {
        "Keys": [
            "interfaceID",
            "sourceAddress"
        ],
        "Filters": [
            {
                "Match": "protocol",
                "EqualTo": 6
            }
    "AggregateOn": "Sum"
}
```

Después de crear las reglas, puede seleccionar la pestaña Metrics (Métricas) en la consola de CloudWatch, donde puede utilizar las siguientes expresiones matemáticas de métricas de ejemplo para graficar los datos que informa Contributor Insights:

Ejemplo: expresiones matemáticas de métricas

```
e1 INSIGHT_RULE_METRIC("RejectedConnectionsRule", "Sum")
e2 INSIGHT_RULE_METRIC("TotalConnectionsRule", "Sum")
e3 (e1/e2)*100
```

En el ejemplo, la expresión matemática de métrica `e3` devuelve todas las conexiones TCP rechazadas. Si desea recibir una notificación cuando se rechace el 20 por ciento de las conexiones TCP, puede modificar la expresión cambiando el umbral de `100` a `20`.

> **Note**
>
> Puede configurar una alarma para una métrica que esté supervisando desde la sección Metrics (Métricas). Mientras está en la pestaña Graphed metrics (Métricas en gráficos), puede seleccionar el icono Create alarm (Crear alarma) en la columna Actions (Acciones). El icono Create alarm (Crear alarma) parece una campana.

Para obtener más información acerca de la representación gráfica de métricas y el uso de funciones matemáticas de métricas, consulte la sección Añadir una expresión matemática a un gráfico de CloudWatch.

Uso de reglas integradas de Contributor Insights

Puede utilizar reglas integradas de Contributor Insights para analizar métricas de otros servicios de AWS. Los siguientes servicios admiten las reglas integradas:

- Contributor Insights para Amazon DynamoDB en la Guía para desarrolladores de Amazon DynamoDB.
- Utilice las reglas integradas de Contributor Insights en la Guía AWS PrivateLink.

Información de aplicaciones de Amazon CloudWatch

Información de aplicaciones de Amazon CloudWatch permite la observabilidad para sus aplicaciones y recursos subyacentes de AWS. Puede ayudarle a configurar los mejores monitoreos para la los recursos de aplicaciones con el fin de analizar de forma continua los datos en busca de señales que indiquen problemas con las aplicaciones. Información de aplicaciones, que emplea tecnología de Sagemaker y otras tecnologías de AWS, proporciona paneles automatizados que muestran posibles problemas con las aplicaciones monitoreadas, lo que le ayuda a aislar rápidamente los problemas en curso con sus aplicaciones e infraestructura. La visibilidad mejorada sobre el estado de las aplicaciones que proporciona Información de aplicaciones ayuda a reducir el tiempo promedio de reparación (MTTR) para solucionar los problemas de la aplicación.

Cuando agrega las aplicaciones a Información de aplicaciones de Amazon CloudWatch, este servicio analiza los recursos en las aplicaciones, y recomienda y configura las métricas y los registros en CloudWatch para los componentes de la aplicación. Los componentes de la aplicación de muestra incluyen la base de datos backend de SQL Server y los IIS de Microsoft o las capas web. Información de aplicaciones analiza los patrones de las métricas mediante datos históricos para detectar anomalías, y detecta de forma continua errores y excepciones en los registros de las aplicaciones, del sistema operativo y de la infraestructura. Relaciona estas observaciones mediante una combinación de algoritmos de clasificación y reglas integradas. A continuación, crea automáticamente paneles que muestran las observaciones pertinentes e información sobre la severidad del problema que le ayudarán a priorizar sus acciones. Para problemas comunes en pilas de aplicaciones .NET y SQL, como la latencia de la aplicación, los errores de copias de seguridad de SQL Server, pérdidas de memoria, solicitudes HTTP grandes y operaciones de E/S canceladas, proporciona información adicional que indica una posible causa raíz y los pasos para la resolución. La integración incorporada con AWS SSM OpsCenter le permite resolver problemas mediante la ejecución del documento correspondiente de automatización de Systems Manager.

Secciones

- Tutorial: configuración de la supervisión para SAP ASE
- Tutorial: Configuración de la supervisión para SAP HANA
- Tutorial: Configuración de la supervisión para SAP NetWeaver
- Visualización y solución de los problemas que Información de aplicaciones de Amazon CloudWatch haya detectado
- Registros y métricas que Información de aplicaciones de Amazon CloudWatch admite

¿Qué es Información de aplicaciones de Amazon CloudWatch?

Información de aplicaciones de CloudWatch le permite supervisar las aplicaciones que utilicen instancias de Amazon EC2 junto con otros recursos de aplicaciones. Identifica y configura métricas clave, registros y alarmas para los recursos y la pila de tecnología de la aplicación (como la base de datos Microsoft SQL Server, los servidores web (IIS) y de aplicaciones, el sistema operativo, los balanceadores de carga y las colas). Monitorea continuamente las métricas y los registros para detectar y relacionar anomalías y errores. Cuando se detectan errores y anomalías, Información de aplicaciones genera eventos de CloudWatch que puede utilizar para configurar notificaciones o realizar acciones. Para ayudar con la solución de problemas, crea paneles automatizados para los problemas detectados, que incluyen anomalías de métricas y errores de registro relacionados, además de información adicional que indica la posible causa raíz. Los paneles automatizados ayudan a adoptar medidas correctivas para mantener las aplicaciones en buen estado y para evitar que los usuarios finales de la aplicación se vean afectados. También crea OpsItems para que pueda resolver los problemas con AWS SSM OpsCenter.

Puede configurar contadores importantes, como la transacción de escritura replicada por segundo, la longitud de cola de recuperación y el retraso de transacción como también los registros de eventos de Windows en CloudWatch. Cuando sucede un evento o problema de conmutación por error con la carga de trabajo de alta disponibilidad de SQL, como un acceso restringido para consultar una base de datos de destino, Información de aplicaciones de CloudWatch proporciona información automatizada.

Información de aplicaciones de CloudWatch se integra con AWS Launch Wizard para proporcionar una experiencia de configuración de supervisión con un solo clic para implementar cargas de trabajo de alta disponibilidad de SQL Server en AWS. Cuando selecciona la opción para configurar la supervisión y la información con Información de aplicaciones en la consola de Launch Wizard, Información de aplicaciones de CloudWatch configura automáticamente las métricas, los registros y las alarmas relevantes en CloudWatch y comienza a supervisar las cargas de trabajo recientemente

implementadas. Puede ver la información automatizada y los problemas detectados, junto con el estado de las cargas de trabajo de alta disponibilidad de SQL Server, en la consola de CloudWatch.

Contenido

Características

Información de aplicaciones ofrece las siguientes características.

Configuración automática de monitores de recursos de la aplicación

Información de aplicaciones de CloudWatch reduce el tiempo que se tarda en configurar la supervisión de las aplicaciones. Para ello, realiza el análisis de los recursos de la aplicación, que proporciona una lista de las métricas y registros recomendados personalizables y los configura en CloudWatch para ofrecer la visibilidad necesaria sobre los recursos de la aplicación, como Amazon EC2 y de Elastic Load Balancer (ELB). También configura alarmas dinámicas para métricas monitorizadas. Las alarmas se actualizan automáticamente en función de las anomalías detectadas en las últimas dos semanas.

Detección y notificación de problemas

Información de aplicaciones de CloudWatch detecta señales que indiquen posibles problemas con la aplicación, como anomalías de métricas y errores de registro. Relaciona estas observaciones para mostrar los posibles problemas de su aplicación. A continuación, genera eventos de CloudWatch Events, que se pueden configurar para recibir notificaciones o realizar las medidas oportunas. De esta manera, ya no tendrá que crear alarmas para métricas individuales o errores de registro.

Resolución de problemas

Información de aplicaciones de CloudWatch crea paneles automatizados de CloudWatch para los problemas detectados. Los paneles muestran información sobre el problema, incluidas las anomalías de las métricas y los errores de registro asociados, para ayudarle a solucionar los problemas.

También proporciona información adicional que indica posibles causas de las anomalías y los errores.

Conceptos

Los siguientes conceptos son importantes para comprender cómo Información de aplicaciones monitoriza la aplicación.

Componente

Un conjunto agrupado automáticamente, independiente o personalizado de recursos similares que conforman una aplicación. Es recomendable agrupar los recursos similares en componentes personalizados para mejorar la monitorización.

Observación

Un evento individual (anomalía de métricas, error de registro o excepción) que se detecta con una aplicación o un recurso de la aplicación.

Problema

Los problemas se detectan relacionando, clasificando y agrupando las observaciones relacionadas.

Para conocer las definiciones de otros conceptos importantes de Información de aplicaciones de CloudWatch, consulte Conceptos de Amazon CloudWatch.

Precios

Información de aplicaciones de CloudWatch configura métricas y registros recomendados para recursos de la aplicación seleccionados mediante las métricas de CloudWatch, registros y eventos de las notificaciones sobre los problemas detectados. Estas características se facturan en su cuenta de AWS de acuerdo con los Precios de CloudWatch. En el caso de los problemas detectados, Información de aplicaciones también crea SSM OpsItems para notificarle los problemas. Además, Información de aplicaciones crea parámetros del almacén de parámetros de SSM para configurar los agentes de CloudWatch en sus instancias. Las características de Amazon EC2 Systems Manager se cobran según los precios de SSM. No se le aplicará ningún cargo por la ayuda con la configuración, la supervisión del análisis de datos ni la detección de problemas.

Costos de Información de aplicaciones de CloudWatch

Los costos de Amazon EC2 incluyen el uso de las siguientes características:

- Agente de CloudWatch
  - Grupos de registros del agente de CloudWatch
  - Métricas del agente de CloudWatch
  - Grupos de registro de Prometheus (para cargas de trabajo de JMX)

Los costos de todos los recursos incluyen el uso de las siguientes características:

- Alarmas de CloudWatch (la mayoría del costo)
- SSM OpsItems (costo mínimo)

Ejemplo de cálculo de costos

Los costos de este ejemplo se calculan de acuerdo con el siguiente escenario.

Creó un grupo de recursos que incluye lo siguiente:

- Una instancia de Amazon EC2 con SQL Server instalado.
- Un volumen de Amazon EBS adjunto.

Al incorporar este grupo de recursos a Información de aplicaciones de CloudWatch, se detecta la carga de trabajo de SQL Server instalada en la instancia de Amazon EC2. A continuación, Información de aplicaciones de CloudWatch comienza a supervisar las siguientes métricas.

Las siguientes métricas se monitorean para la instancia de SQL Server:

- CPUUtilization
- StatusCheckFailed
- Porcentaje de bytes confirmados en uso de memoria
- Mbytes disponibles de memoria
- Total de bytes de interfaz de red/segundo
- Porcentaje de uso de archivo de paginación
- Porcentaje de tiempo de disco del disco físico
- Porcentaje de tiempo de procesador del procesador
- SQLServer: tasa de aciertos de la caché del administrador del búfer
- SQLServer: expectativa de vida del administrador del búfer

- SQLServer: estadísticas generales: procesos bloqueados
- SQLServer: estadísticas generales: conexiones de usuario
- SQLServer: bloqueos: número de interbloqueos/segundo
- SQLServer: estadísticas de SQL: solicitudes por lotes/segundo
- Longitud de cola del procesador del sistema

Las siguientes métricas se monitorean para los volúmenes adjuntos a la instancia de SQL Server:

- VolumeReadBytes
- VolumeWriteBytes
- VolumeReadOps
- VolumeWriteOps
- VolumeTotalReadTime
- VolumeTotalWriteTime
- VolumeIdleTime
- VolumeQueuelength
- VolumeThroughputPercentage
- VolumenConsumedReadWriteOps
- BurstBalance

Para este escenario, los costos se calculan según la página de precios de CloudWatch y la página de precios de SSM:

- Métricas personalizadas

  Para este escenario, 13 de las métricas anteriores se emiten a CloudWatch mediante el agente de CloudWatch. Estas métricas se tratan como métricas personalizadas. El costo de cada métrica personalizada es de 0,3 USD al mes. El costo total de estas métricas personalizadas es de 13 · 0,3 USD = 3,90 USD al mes.
- Alarmas

  Para este escenario, Información de aplicaciones de CloudWatch monitorea 26 métricas en total, lo que crea 26 alarmas. El costo de cada alarma es de 0,1 USD al mes. El costo total de las alarmas es de 26 * 0,1 USD = 2,60 USD al mes.

- Ingesta de datos y registros de errores

  El costo de la ingesta de datos es de 0,05 USD/GB y el almacenamiento del registro de errores de SQL Server es de 0,03 USD/GB. El coste total de la ingesta de datos y el registro de errores es de 0,05 USD/GB + 0,03 USD/GB = 0,08 USD/GB.

- Amazon EC2 Systems Manager OpsItems

  Se crea un elemento SSM OpsItem por cada problema que detecte Información de aplicaciones de CloudWatch. Para un número n de problemas en su solicitud, el costo total es de 0,00267 USD · n al mes.

Servicios relacionados

Los siguientes servicios se utilizan junto con Información de aplicaciones de CloudWatch:

Servicios de AWS relacionados

- Con Amazon CloudWatch, podrá ver la utilización de recursos, el rendimiento de las aplicaciones y el estado de funcionamiento de todo el sistema. Recopila y realiza un seguimiento de las métricas, envía notificaciones de alarma, actualiza automáticamente los recursos que monitoriza en función de las reglas que haya definido y le permite monitorizar sus propias métricas personalizadas. Información de aplicaciones de CloudWatch se inicia a través de CloudWatch (en concreto, dentro de los paneles operativos predeterminados de CloudWatch). Para más información, consulte la Guía del usuario de Amazon CloudWatch.
- Información de contenedores de CloudWatch recopila, agrega y resume métricas y registros de las aplicaciones y microservicios en contenedores. Puede utilizar Información de contenedores para monitorear las plataformas Amazon ECS, Amazon Elastic Kubernetes Service y Kubernetes en Amazon EC2. Cuando Información de aplicaciones está habilitado en las consolas Información de contenedores o Información de aplicaciones, Información de aplicaciones muestra los problemas detectados en el panel de Información de contenedores. Para obtener más información, consulte Información de contenedores.
- Amazon DynamoDB es un servicio de base de datos NoSQL completamente administrado que le permite delegar las cargas administrativas que supone tener que utilizar y escalar bases de datos distribuidas, para que no tenga que preocuparse del aprovisionamiento, la instalación ni la configuración del hardware, ni tampoco de las tareas de replicación, aplicación de parches de software o escalado de clústeres. DynamoDB también ofrece el cifrado en reposo, que elimina la carga y la complejidad operativa que conlleva la protección de información confidencial.

- Amazon EC2 proporciona capacidad informática escalable en la nube de AWS. Puede usar Amazon EC2 para lanzar tantos servidores virtuales como necesite, configurar la seguridad y las redes, y administrar el almacenamiento. Puede escalar hacia arriba o hacia abajo para controlar los cambios en los requisitos o los picos de popularidad, con lo que se reduce la necesidad de prever el tráfico. Para obtener más información, consulte la Guía del usuario de Amazon EC2 para instancias de Linux o la Guía de Amazon EC2 para instancias de Windows.
- Amazon Elastic Block Store (Amazon EBS) proporciona volúmenes de almacenamiento de nivel de bloque para su uso con instancias de Amazon EC2. Los volúmenes de Amazon EBS se comportan como dispositivos de bloques sin formatear y sin procesar. Puede montar estos volúmenes como dispositivos en sus instancias. Los volúmenes de Amazon EBS que están adjuntados a una instancia se exponen como volúmenes de almacenamiento que persisten independientemente de la duración de la instancia. Puede crear un sistema de archivos sobre estos volúmenes o utilizarlos de cualquier modo en el que utilizaría un dispositivo de bloques (como un disco duro). Puede cambiar dinámicamente la configuración de un volumen adjunto a una instancia. Para obtener más información, consulte la Amazon EBS User Guide (Guía del usuario de Amazon EBS).
- Amazon EC2 Auto Scaling lo ayuda a garantizar que cuenta con la cantidad correcta de instancias EC2 disponibles para controlar la carga de su aplicación. Para obtener más información, consulte la Guía del usuario de Amazon EC2 Auto Scaling.
- Elastic Load Balancing distribuye el tráfico entrante de red o de la aplicación entre varios destinos, por ejemplo, instancias EC2, contenedores y direcciones IP en varias zonas de disponibilidad. Para obtener más información, consulte la Guía del usuario de Elastic Load Balancing.
- IAM es un servicio web que ayuda a controlar de forma segura el acceso de los usuarios a los recursos de AWS. Utilice IAM para controlar quién puede usar los recursos de AWS (autenticación), los recursos que pueden usar y cómo pueden usarlos (autorización). Para obtener más información, consulte Autenticación y control de acceso de Amazon CloudWatch.
- AWS Lambda le permite crear aplicaciones sin servidor compuestas por funciones que se activan por eventos y se implementan automáticamente con CodePipeline y CodeBuild de AWS. Para obtener más información, consulte Aplicaciones de AWS Lambda.
- AWS Launch Wizard for SQL Server reduce el tiempo que se tarda en implementar la solución de alta disponibilidad de SQL Server en la nube. Introduzca los requisitos de la aplicación, incluido el rendimiento, el número de nodos y la conectividad en la consola de servicio, y AWS Launch Wizard identificará los recursos adecuados de AWS para implementar y ejecutar la aplicación Always On de SQL Server.
- Resource Groups de AWS le ayudan a organizar los recursos que componen la aplicación. Con los Resource Groups, puede administrar y automatizar tareas en un gran número de recursos al

mismo tiempo. Solo se puede registrar un grupo de recursos para una aplicación. Para obtener más información, consulte la AWS Resource Groups User Guide (Guía del usuario de Resource Groups).

- Amazon SQS ofrece una cola alojada segura, duradera y disponible que le permite integrar y desacoplar sistemas y componentes de software distribuidos. Para obtener más información, consulte la Guía del usuario de Amazon SQS.
- AWS Step Functions es un compositor de funciones sin servidor que le permite secuenciar una variedad de servicios y recursos de AWS, incluidas las funciones de AWS Lambda, en flujos de trabajo estructurados y visuales. Para obtener más información, consulte la Guía del usuario de AWS Step Functions.
- SSM OpsCenter de AWS acumula y estandariza OpsItems en todos los servicios a la vez que proporciona datos de investigación contextual sobre cada OpsItem, OpsItems relacionados y recursos relacionados. OpsCenter también proporciona documentos de Systems Manager Automation (runbooks) que puede utilizar para resolver problemas rápidamente. Puede especificar datos que se pueden buscar y personalizar para cada OpsItem. También puede ver informes de resumen generados automáticamente sobre OpsItem por estado y origen. Para obtener más información, consulte la Guía del usuario de AWS Systems Manager.
- Amazon API Gateway es un servicio de AWS para la creación, la publicación, el mantenimiento, el monitoreo y la protección de las API REST, HTTP y de WebSocket a cualquier escala. Los desarrolladores de la API pueden crear API que obtengan acceso a AWS o a otros servicios web, así como los datos almacenados en la nube de AWS. Para obtener más información, consulte Amazon API Gateway User Guide (Guía del usuario de Amazon API Gateway).

  > **Note**
  >
  > Información de aplicaciones solo admite protocolos de la API de REST (v1 del servicio API Gateway).

- Amazon Elastic Container Service (Amazon ECS) es un servicio de orquestación de contenedores completamente administrado. Puede utilizar Amazon ECS para ejecutar sus aplicaciones más confidenciales y de misión crítica. Para obtener más información, consulte Amazon Elastic Container Service Developer Guide (Guía para desarrolladores de Amazon Elastic Container Service).
- Amazon Elastic Kubernetes Service (Amazon EKS) es un servicio administrado que puede usar para ejecutar Kubernetes en AWS sin tener que instalar, operar y mantener su propio plano de control o nodos de Kubernetes. Kubernetes es un sistema de código abierto para automatizar la

implementación, escalado y administración de las aplicaciones en contenedores. Para obtener más información, consulte la Amazon EKS User Guide (Guía del usuario de Amazon EKS).

- Kubernetes en Amazon EC2. Kubernetes es un software de código abierto que le ayuda a implementar y administrar aplicaciones en contenedores a escala. Kubernetes administra clústeres de instancias informáticas de Amazon EC2 y ejecuta contenedores en esas instancias con procesos de implementación, mantenimiento y escalado. Con Kubernetes puede ejecutar cualquier tipo de aplicación en contenedores con el mismo conjunto de herramientas en las instalaciones y en la nube. Para obtener más información, consulte Documentación: Introducción a Kubernetes.
- Amazon FSx ayuda a lanzar y ejecutar sistemas de archivos populares que están completamente administrados por AWS. Con Amazon FSx, puede aprovechar los conjuntos de características y el rendimiento de los sistemas de archivos comunes de código abierto y con licencia comercial para evitar tareas administrativas que consumen mucho tiempo. Para obtener más información, consulte la Amazon FSx Documentation (Documentación de Amazon FSx).
- Amazon Simple Notification Service (SNS) es un servicio de mensajería completamente administrado para la comunicación de aplicación a aplicación y de aplicación a persona. Puede configurar Amazon SNS para el monitoreo mediante Información de aplicaciones. Cuando se configura Amazon SNS como recurso para monitorear, Información de aplicaciones realiza un seguimiento de las métricas de SNS para ayudar a determinar el motivo de los problemas o fallas de los mensajes de SNS.
- Amazon Elastic File System (Amazon EFS) es un sistema de archivos NFS elástico completamente administrado que se utiliza con servicios de Nube de AWS y recursos en las instalaciones. Está diseñado para escalar a petabytes bajo demanda sin interrumpir las aplicaciones. Aumenta y disminuye automáticamente a medida que se agregan o eliminan archivos, lo que elimina la necesidad de aprovisionar y administrar la capacidad para adaptarse al crecimiento. Para obtener más información, consulte la documentación del producto de Amazon Elastic File System.

Servicios de terceros relacionados

- Para algunas cargas de trabajo y aplicaciones monitoreadas en Información de aplicaciones, JMX Exporter de Prometheus se instala mediante AWS Systems Manager para que Información de aplicaciones de CloudWatch pueda recuperar métricas específicas de Java. Cuando decide monitorear una aplicación Java, Información de aplicaciones instala automáticamente JMX Exporter de Prometheus.

Componentes de aplicaciones admitidos

Información de aplicaciones de CloudWatch analiza su grupo de recursos para identificar los componentes de la aplicación. Los componentes pueden ser independientes, estar agrupados automáticamente (como las instancias de un grupo de escalado automático o detrás de un equilibrador de carga) o personalizados (mediante la agrupación de instancias de Amazon EC2 distintas).

Información de aplicaciones de CloudWatch admite los siguientes componentes:

Componentes de AWS

- Amazon EC2
- Amazon EBS
- Amazon RDS
- Elastic Load Balancing: Application Load Balancer y Classic Load Balancer (todas las instancias de destino de estos balanceadores de carga se identifican y configuran).
- Grupos de Amazon EC2 Auto Scaling: escalado automático de AWS (los grupos de escalado automático se configuran de forma dinámica para todas las instancias de destino; si la aplicación escala verticalmente, Información de aplicaciones de CloudWatch configura automáticamente las nuevas instancias). Los grupos de escalado automático no son compatibles con los grupos de recursos basados en pilas de CloudFormation.
- AWS Lambda
- Amazon Simple Queue Service (Amazon SQS)
- La tabla de Amazon DynamoDB.
- Las Métricas de buckets de Amazon S3
- AWS Step Functions
- Etapas de la API REST de Amazon API Gateway
- Amazon Elastic Container Service (Amazon ECS): clúster, servicio y tarea
- Amazon Elastic Kubernetes Service (Amazon EKS): clúster
- Kubernetes en Amazon EC2: clúster de Kubernetes que se ejecuta en EC2
- Tema de Amazon SNS

Información de aplicaciones de CloudWatch no realiza un seguimiento de ningún otro tipo de recursos de componentes. Si un tipo de componente admitido no aparece en su aplicación de

Información de aplicaciones, es posible que ya se haya registrado y que otra aplicación que esté monitorizada por Información de aplicaciones lo esté administrando.

Pilas de tecnología admitidas

Puede utilizar Información de aplicaciones de CloudWatch para supervisar las aplicaciones que se ejecuten en sistemas operativos Windows Server y Linux si selecciona la opción de menú desplegable de nivel de aplicación para una de las siguientes tecnologías:

- Front-end: servidor web de Microsoft Internet Information Services (IIS)
- Nivel de empleado:
  - .NET Framework
  - .NET Core
- Aplicaciones:
  - Java
  - Implementaciones estándar, distribuidas y de alta disponibilidad de SAP NetWeaver
- Active Directory
- SharePoint
- Bases de datos:
  - Microsoft SQL Server que se ejecuta en Amazon RDS o Amazon EC2 (incluidas las configuraciones de alta disponibilidad de SQL Server. Consulte, Ejemplos de configuración del componente)
  - MySQL que se ejecuta en Amazon RDS, Amazon Aurora o Amazon EC2
  - PostgreSQL que se ejecuta en Amazon RDS o Amazon EC2
  - La tabla de Amazon DynamoDB.
  - Oracle que se ejecuta en Amazon RDS o Amazon EC2
  - Base de datos SAP HANA en una única instancia de Amazon EC2 y en varias instancias de Amazon EC2.
  - Configuración de alta disponibilidad dela base de datos Cross-AZ SAP HANA
  - Base de datos SAP Sybase ASE en una única instancia de Amazon EC2
  - Configuración de alta disponibilidad de la base de datos Cross-AZ SAP Sybase ASE

Si ninguna de las pilas de tecnología enumeradas anteriormente se aplica a los recursos de la aplicación, puede monitorear la pila de aplicaciones eligiendo Custom (Personalizada) en el menú desplegable de nivel de aplicación de la página Manage monitoring (Administrar monitoreo).

Descubra cómo funciona Información de aplicaciones de Amazon CloudWatch

Esta sección contiene información sobre cómo funciona CloudWatch Application e incluye:

- Cómo Información de aplicaciones monitorea las aplicaciones
- Retención de datos
- Cuotas
- Paquetes de Systems Manager (SSM) de AWS que Información de aplicaciones de CloudWatch utiliza
- Documentos de Systems Manager (SSM) de AWS utilizados por Información de aplicaciones de CloudWatch

Cómo Información de aplicaciones monitorea las aplicaciones

Información de aplicaciones monitorea las aplicaciones como se indica a continuación.

Detección y configuración de aplicaciones

La primera vez que una aplicación se agrega a Información de aplicaciones de CloudWatch, se examinan los componentes de la aplicación para recomendar las métricas clave, los registros y otros orígenes de datos para supervisar la aplicación. A continuación, puede configurar la aplicación en función de estas recomendaciones.

Preprocesamiento de datos

Información de aplicaciones de CloudWatch analiza continuamente los orígenes de datos que se supervisen en los recursos de la aplicación para detectar anomalías de métricas y errores de registro (observaciones).

Detección inteligente de problemas

El motor de Información de aplicaciones de CloudWatch detecta los problemas de la aplicación; para ello, relaciona las observaciones con algoritmos de clasificación y reglas integradas. Para ayudarlo

a solucionar problemas, crea paneles de CloudWatch automatizados, que incluyen información contextual acerca de los problemas.

Alerta y acción

Cuando Información de aplicaciones de CloudWatch detecta un problema con la aplicación, genera eventos de CloudWatch para informarle del problema. ConsulteInformación de aplicaciones CloudWatch Events y notificaciones para problemas detectados para obtener más información acerca de cómo configurar estos eventos.

Escenario de ejemplo

Tiene una aplicación ASP .NET respaldada por una base de datos SQL Server. De repente, la base de datos comienza a funcionar incorrectamente debido a un alto consumo de memoria. Esto merma el rendimiento de la aplicación y se producen errores HTTP 500 en los servidores web y el balanceador de carga.

Con Información de aplicaciones de CloudWatch y los análisis inteligentes, puede identificar la capa de la aplicación que esté causando el problema mediante la verificación del panel creado de manera dinámica que muestra las métricas relacionadas y fragmentos de los archivos de registro. En este caso, el problema podría estar en la capa de la base de datos SQL.

Retención de datos

Información de aplicaciones de CloudWatch conserva los problemas durante 55 días y las observaciones durante 60 días.

Cuotas

Para conocer las cuotas predeterminadas para Información de aplicaciones de CloudWatch, consulte Información de aplicaciones de Amazon CloudWatch endpoints and quotas (Puntos de enlace y cuotas de Información de aplicaciones de Amazon CloudWatch). A menos que se indique lo contrario, cada cuota es por cada Región de AWS. Comuníquese con AWS Support para solicitar un aumento en su cuota de servicio. Muchos servicios contienen cuotas que no se pueden cambiar. Para obtener más información acerca de las cuotas de un servicio específico, consulte la documentación correspondiente a dicho servicio.

Paquetes de Systems Manager (SSM) de AWS que Información de aplicaciones de CloudWatch utiliza

Información de aplicaciones utiliza los paquetes que se enumeran en esta sección y se pueden administrar e implementar de forma independiente con Distributor de AWS Systems Manager. Para obtener más información acerca de SSM Distributor, consulte AWS Systems Manager Distributor en la Guía del usuario de Systems Manager de AWS.

Paquetes:

- AWSObservabilityExporter-JMXExporterInstallAndConfigure
- AWSObservabilityExporter-SAP-HANADBExporterInstallAndConfigure
- AWSObservabilityExporter-HAClusterExporterInstallAndConfigure
- AWSObservabilityExporter-SAP-SAPHostExporterInstallAndConfigure
- AWSObservabilityExporter-SQLExporterInstallAndConfigure

`AWSObservabilityExporter-JMXExporterInstallAndConfigure`

Puede recuperar métricas Java específicas de la carga de trabajo desde Prometheus JMX exporter para Información de aplicaciones a fin de configurar y monitorear las alarmas. En la consola de Información de aplicaciones, en la página Administrar monitoreo, seleccione Aplicación JAVA desde el menú desplegable Nivel de aplicación. Luego, en JAVA Prometheus exporter configuration (Configuración del exportador JAVA Prometheus), seleccione el Collection method (Método de recopilación) y JMX port number (Número de puerto de JMX).

Para utilizar Distributor de AWS Systems Manager para instalar y configurar el paquete JMX Exporter de Prometheus que proporciona AWS independientemente de Información de aplicaciones, siga los pasos que se describen a continuación.

Prerrequisitos para usar el paquete SSM del JMX Exporter de Prometheus

- Debe contar con la versión 2.3.1550.0 de SSM Agent instalada o una posterior
- La variable de entorno JAVA_HOME debe estar establecida

Instale y configure el paquete **`AWSObservabilityExporter-JMXExporterInstallAndConfigure`**

El paquete `AWSObservabilityExporter-JMXExporterInstallAndConfigure` es un paquete de SSM Distributor que puede usar para instalar y configurar Prometheus JMX Exporter. Cuando el JMX Exporter de Prometheus envía métricas de Java, el agente de CloudWatch se puede configurar para recuperar las métricas del servicio CloudWatch.

1. En función de sus preferencias, prepare el Prometheus JMX Exporter YAML configuration file (Archivo de configuración YAML del exportador JMX de Prometheus) ubicado en el repositorio Prometheus de GitHub. Utilice la configuración de ejemplo y las descripciones de opciones a modo de guía.
2. Copie el archivo de configuración YAML del JMX Exporter de Prometheus codificado como Base64 a un nuevo parámetro SSM en el SSM Parameter Store (Almacén de parámetros de SSM).
3. Diríjase a la consola SSM Distributor y abra la pestaña Owned by Amazon (Perteneciente a Amazon). Seleccione AWSObservabilityExporter-JMXExporterInstallAndConfigure y elija Install one time (Instalar una vez).
4. Actualice el parámetro de SSM que creó en el primer paso mediante el reemplazo de ‘Argumentos adicionales’ por lo siguiente:

```
{
  "SSM_EXPORTER_CONFIGURATION": "{{ssm:<SSM_PARAMETER_STORE_NAME>}}",
  "SSM_EXPOSITION_PORT": "9404"
}
```

> **Note**
>
> El puerto 9404 es el puerto predeterminado que se utiliza para enviar métricas de JMX de Prometheus. Puede actualizar este puerto.

Ejemplo: Configurar el agente de CloudWatch para recuperar métricas de Java

1. Instale JMX Exporter de Prometheus, como se describe en el procedimiento anterior. A continuación, verifique el estado del puerto para comprobar que se haya instalado correctamente en la instancia.

   Ejemplo de instalación exitosa en la instancia de Windows

   ```
   PS C:\> curl http://localhost:9404 (http://localhost:9404/)
   ```

```
StatusCode : 200
StatusDescription : OK
Content : # HELP jvm_info JVM version info
```

Ejemplo de instalación exitosa en la instancia de Linux

```
$ curl localhost:9404
# HELP jmx_config_reload_failure_total Number of times configuration have failed to
 be reloaded.
# TYPE jmx_config_reload_failure_total counter
jmx_config_reload_failure_total 0.0
```

2. Cree el archivo YAML de detección del servicio Prometheus. El siguiente ejemplo de archivo de detección de servicios realiza lo siguiente:

   - Especifica el puerto del host del JMX Exporter de Prometheus como `localhost: 9404`.
   - Adjunta etiquetas (`Application`, `ComponentName`, y `InstanceId`) a las métricas, que se pueden establecer como dimensiones métricas de CloudWatch.

```
$ cat prometheus_sd_jmx.yaml
- targets:
  - 127.0.0.1:9404
  labels:
    Application: myApp
    ComponentName: arn:aws:elasticloadbalancing:us-
east-1:123456789012:loadbalancer/app/sampl-Appli-MMZW8E3GH4H2/aac36d7fea2a6e5b
    InstanceId: i-12345678901234567
```

3. Cree el archivo YAML de configuración del JMX Exporter de Prometheus. El siguiente ejemplo de archivo de configuración especifica lo siguiente:

   - El intervalo de trabajo de recuperación de métricas y el período de tiempo de espera.
   - Los trabajos de recuperación de métricas (`jmx` y `sap`), también conocidos como “raspado”, que incluyen el nombre del trabajo, la serie temporal máxima devuelta a la vez y la ruta del archivo de detección de servicios.

```
$ cat prometheus.yaml
global:
  scrape_interval: 1m
```

```
  scrape_timeout: 10s
scrape_configs:
  - job_name: jmx
    sample_limit: 10000
    file_sd_configs:
      - files: ["/tmp/prometheus_sd_jmx.yaml"]
  - job_name: sap
    sample_limit: 10000
    file_sd_configs:
      - files: ["/tmp/prometheus_sd_sap.yaml"]
```

4. Verifique que el agente de CloudWatch esté instalado en la instancia de Amazon EC2 y que la versión sea 1.247346.1b249759 o una posterior. Para instalar el agente de CloudWatch en la instancia EC2, consulte Installing the CloudWatch Agent (Instalación del agente de CloudWatch). Para verificar la versión, consulte Finding information about CloudWatch agent versions (Búsqueda de información acerca de las versiones del agente CloudWatch).
5. Configure el agente de CloudWatch. Para obtener más información acerca de cómo se configurar el archivo de configuración del agente de CloudWatch, consulte Manually create or edit the CloudWatch agent configuration file (Crear o editar manualmente el archivo de configuración del agente de CloudWatch). En el siguiente ejemplo de archivo de configuración del agente de CloudWatch se realiza lo siguiente:

   - Se especifica la ruta del archivo de configuración del JMX Exporter de Prometheus.
   - Se especifica el grupo de registros de destino en el que se publicarán los registros de las métricas EMF.
   - Se especifican dos conjuntos de dimensiones para cada nombre de métrica.
   - Se envían 8 métricas de CloudWatch (4 nombres de métricas y 2 conjuntos de dimensiones por nombre de métrica).

```
{
   "logs":{
      "logs_collected":{
         ....
      },
      "metrics_collected":{
         "prometheus":{
            "cluster_name":"prometheus-test-cluster",
            "log_group_name":"prometheus-test",
            "prometheus_config_path":"/tmp/prometheus.yaml",
```

```
            "emf_processor":{
               "metric_declaration_dedup":true,
               "metric_namespace":"CWAgent",
               "metric_unit":{
                  "jvm_threads_current":"Count",
                  "jvm_gc_collection_seconds_sum":"Second",
                  "jvm_memory_bytes_used":"Bytes"
               },
               "metric_declaration":[
                  {
                     "source_labels":[
                        "job"
                     ],
                     "label_matcher":"^jmx$",
                     "dimensions":[
                        [
                           "InstanceId",
                           "ComponentName"
                        ],
                        [
                           "ComponentName"
                        ]
                     ],
                     "metric_selectors":[
                        "^java_lang_threading_threadcount$",
                        "^java_lang_memory_heapmemoryusage_used$",
                        "^java_lang_memory_heapmemoryusage_committed$"
                     ]
                  }
               ]
            }
         }
      }
   },
   "metrics":{
      ....
   }
}
```

AWSObservabilityExporter-SAP-HANADBExporterInstallAndConfigure

Puede recuperar métricas de SAP HANA específicas de la carga de trabajo desde el exportador de base de datos HANA de Prometheus para que Información de aplicaciones configure y monitoree las alarmas. Para obtener más información, consulte la sección Configuración de la base de datos SAP HANA para la supervisión de esta guía.

Para utilizar Distributor de AWS Systems Manager para armar, instalar y configurar el paquete exportador de base de datos HANA de Prometheus que proporciona AWS independientemente de Información de aplicaciones, siga los pasos que se describen a continuación.

Requisitos previos para usar el paquete SSM del exportador de base de datos HANA de Prometheus

- Debe contar con la versión 2.3.1550.0 de SSM Agent instalada o una posterior
- Base de datos SAP HANA
- Sistema operativo Linux (SUSE Linux, RedHat Linux)
- Un secreto con las credenciales de monitoreo de bases de datos SAP HANA, mediante AWS Secrets Manager. Cree un secreto mediante el formato de pares clave-valor, especifique el nombre de usuario clave e ingrese el usuario de la base de datos en el valor. Agregue una segunda contraseña clave y, a continuación, ingrese la contraseña para el valor. Para obtener más información acerca de la creación de secretos, consulte Crear un secreto en la Guía del usuario de AWS Secrets Manager. El secreto debe tener el siguiente formato:

```
{
  "username": "<database_user>",
  "password": "<database_password>"
}
```

Instale y configure el paquete AWSObservabilityExporter-SAP-HANADBExporterInstallAndConfigure

El paquete `AWSObservabilityExporter-SAP-HANADBExporterInstallAndConfigure` es un paquete de SSM Distributor que puede usar para instalar y configurar Prometheus HANA database Exporter (Exportador de base de datos HANA de Prometheus). Cuando el exportador de base de datos HANA de Prometheus envía métricas de base de datos HANA, el agente de CloudWatch se puede configurar para recuperar las métricas del servicio de CloudWatch.

1. Cree un parámetro SSM en Almacén de parámetros SSM para almacenar las configuraciones del exportador. A continuación se muestra un valor de parámetro de ejemplo.

```
{\"exposition_port\":9668,\"multi_tenant\":true,\"timeout\":600,\"hana\":{\"host\":
\"localhost\",\"port\":30013,\"aws_secret_name\":\"HANA_DB_CREDS\",\"scale_out_mode
\":true}}
```

> **Note**
>
> En este ejemplo, la exportación se ejecuta solo en la instancia de Amazon EC2 con la base de datos del SYSTEM activo y permanece inactiva en las demás instancias de EC2 para evitar métricas duplicadas. El exportador puede recuperar toda la información del inquilino de la base de datos desde la base de datos del SYSTEM.

2. Cree un parámetro SSM en SSM Parameter Store para almacenar las consultas de métricas del exportador. El paquete puede aceptar más de un parámetro de métrica. Cada parámetro debe tener un formato de objeto JSON válido. A continuación se muestra un valor parámetro de ejemplo:

```
{\"SELECT MAX(TIMESTAMP) TIMESTAMP, HOST, MEASURED_ELEMENT_NAME CORE,
 SUM(MAP(CAPTION, 'User Time', TO_NUMBER(VALUE), 0)) USER_PCT, SUM(MAP(CAPTION,
 'System Time', TO_NUMBER(VALUE), 0)) SYSTEM_PCT, SUM(MAP(CAPTION, 'Wait
 Time', TO_NUMBER(VALUE), 0)) WAITIO_PCT, SUM(MAP(CAPTION, 'Idle Time', 0,
 TO_NUMBER(VALUE))) BUSY_PCT, SUM(MAP(CAPTION, 'Idle Time', TO_NUMBER(VALUE), 0))
 IDLE_PCT FROM sys.M_HOST_AGENT_METRICS WHERE MEASURED_ELEMENT_TYPE = 'Processor'
 GROUP BY HOST, MEASURED_ELEMENT_NAME;\":{\"enabled\":true,\"metrics\":[{\"name\":
\"hanadb_cpu_user\",\"description\":\"Percentage of CPU time spent by HANA DB in user
 space, over the last minute (in seconds)\",\"labels\":[\"HOST\",\"CORE\"],\"value\":
\"USER_PCT\",\"unit\":\"percent\",\"type\":\"gauge\"},{\"name\":\"hanadb_cpu_system
\",\"description\":\"Percentage of CPU time spent by HANA DB in Kernel space,
 over the last minute (in seconds)\",\"labels\":[\"HOST\",\"CORE\"],\"value\":
\"SYSTEM_PCT\",\"unit\":\"percent\",\"type\":\"gauge\"},{\"name\":\"hanadb_cpu_waitio
\",\"description\":\"Percentage of CPU time spent by HANA DB in IO mode, over the
 last minute (in seconds)\",\"labels\":[\"HOST\",\"CORE\"],\"value\":\"WAITIO_PCT\",
\"unit\":\"percent\",\"type\":\"gauge\"},{\"name\":\"hanadb_cpu_busy\",\"description
\":\"Percentage of CPU time spent by HANA DB, over the last minute (in seconds)\",
\"labels\":[\"HOST\",\"CORE\"],\"value\":\"BUSY_PCT\",\"unit\":\"percent\",\"type\":
\"gauge\"},{\"name\":\"hanadb_cpu_idle\",\"description\":\"Percentage of CPU time not
 spent by HANA DB, over the last minute (in seconds)\",\"labels\":[\"HOST\",\"CORE
\"],\"value\":\"IDLE_PCT\",\"unit\":\"percent\",\"type\":\"gauge\"}]}}
```

Para obtener más información acerca de las consultas de métricas, consulte el repositorio de SUSE / hanadb_exporter en GitHub.

3. Diríjase a la consola SSM Distributor y abra la pestaña Owned by Amazon (Perteneciente a Amazon). Seleccione AWSObservabilityExporter-SAP-HANADBExporterInstallAndConfigure* y elija Install one time (Instalar una vez).
4. Actualice el parámetro de SSM que creó en el primer paso mediante el reemplazo de 'Argumentos adicionales' por lo siguiente:

```
{
  "SSM_EXPORTER_CONFIG": "{{ssm:<*SSM_CONFIGURATIONS_PARAMETER_STORE_NAME>*}}",
  "SSM_SID": "<SAP_DATABASE_SID>",
  "SSM_EXPORTER_METRICS_1": "{{ssm:<SSM_FIRST_METRICS_PARAMETER_STORE_NAME>}}",
  "SSM_EXPORTER_METRICS_2": "{{ssm:<SSM_SECOND_METRICS_PARAMETER_STORE_NAME>}}"
}
```

5. Seleccione las instancias de Amazon EC2 con base de datos SAP HANA y luego elija Run (Ejecutar).

AWSObservabilityExporter-HAClusterExporterInstallAndConfigure

Puede recuperar métricas de clúster de alta disponibilidad (HA) específicas de la carga de trabajo desde el Exportador de clúster HANA de Prometheus para Información de aplicaciones para configurar y monitorear alarmas para una configuración de alta disponibilidad de base de datos SAP HANA. Para obtener más información, consulte la sección Configuración de la base de datos SAP HANA para la supervisión de esta guía.

Para utilizar Distribuidor de AWS Systems Manager para armar, instalar y configurar el paquete exportador de clúster de alta disponibilidad de Prometheus que proporciona AWS independientemente de Información de aplicaciones, siga los pasos que se describen a continuación.

Requisitos previos para usar el paquete SSM del exportador de clúster de alta disponibilidad de Prometheus

- Debe contar con la versión 2.3.1550.0 de SSM Agent instalada o una posterior
- Clúster de alta disponibilidad para Pacemaker, Corosync, SBD y DRBD
- Sistema operativo Linux (SUSE Linux, RedHat Linux)

Instale y configure el paquete **AWSObservabilityExporter-HAClusterExporterInstallAndConfigure**

El paquete `AWSObservabilityExporter-HAClusterExporterInstallAndConfigure` es un paquete de SSM Distributor que puede usar para instalar y configurar el exportador de clúster de alta disponibilidad de Prometheus. Cuando el exportador de base de datos HANA de Prometheus envía métricas de clúster, el agente de CloudWatch se puede configurar para recuperar las métricas del servicio de CloudWatch.

1. Cree un parámetro SSM en SSM Parameter Store para almacenar las configuraciones del exportador en formato JSON. A continuación se muestra un valor de parámetro de ejemplo.

   ```
   {\"port\":\"9664\",\"address\":\"0.0.0.0\",\"log-level\":\"info\",\"crm-mon-path
   \":\"/usr/sbin/crm_mon\",\"cibadmin-path\":\"/usr/sbin/cibadmin\",\"corosync-
   cfgtoolpath-path\":\"/usr/sbin/corosync-cfgtool\",\"corosync-quorumtool-path\":\"/
   usr/sbin/corosync-quorumtool\",\"sbd-path\":\"/usr/sbin/sbd\",\"sbd-config-path\":
   \"/etc/sysconfig/sbd\",\"drbdsetup-path\":\"/sbin/drbdsetup\",\"enable-timestamps
   \":false}
   ```

   Para obtener más información acerca de las configuraciones del exportador, consulte el repositorio `ClusterLabs / ha_cluster_exporter` en GitHub.

2. Diríjase a la consola SSM Distributor y abra la pestaña Owned by Amazon (Perteneciente a Amazon). Seleccione AWSObservabilityExporter-HAClusterExporterInstallAndConfigure* y luego elija Install one time (Instalar una vez).

3. Actualice el parámetro de SSM que creó en el primer paso mediante el reemplazo de ‘Argumentos adicionales’ por lo siguiente:

   ```
   {
     "SSM_EXPORTER_CONFIG": "{{ssm:<*SSM_CONFIGURATIONS_PARAMETER_STORE_NAME>*}}"
   }
   ```

4. Seleccione las instancias de Amazon EC2 con base de datos SAP HANA y luego elija Run (Ejecutar).

AWSObservabilityExporter-SAP-SAPHostExporterInstallAndConfigure

Puede recuperar métricas de SAP NetWeaver específicas de la carga de trabajo desde el exportador de host de Prometheus SAP para que Información de aplicaciones configure y supervise las alarmas

de las implementaciones distribuidas y de alta disponibilidad de SAP NetWeaver. Para obtener más información, consulte Introducción a Información de aplicaciones de Amazon CloudWatch.

Si quiere utilizar el distribuidor de AWS Systems Manager para empaquetar, instalar y configurar el paquete del exportador de host de SAP independientemente de Información de aplicaciones, siga los pasos que se describen a continuación.

Requisitos previos para usar el paquete SSM del exportador de host de Prometheus SAP

- Debe contar con la versión 2.3.1550.0 de SSM Agent instalada o una posterior
- Servidores de aplicaciones de SAP NetWeaver
- Sistema operativo Linux (SUSE Linux, RedHat Linux)

Instale y configure el paquete **AWSObservabilityExporter-SAP-SAPHostExporterInstallAndConfigure**

El paquete `AWSObservabilityExporter-SAP-SAPHostExporterInstallAndConfigure` es un paquete del distribuidor de SSM que puede usar para instalar y configurar el exportador de métricas de Prometheus de SAP NetWeaver. Cuando el exportador de Prometheus envía métricas de SAP NetWeaver, el agente de CloudWatch se puede configurar para recuperar las métricas del servicio CloudWatch.

1. Cree un parámetro SSM en SSM Parameter Store para almacenar las configuraciones del exportador en formato JSON. A continuación se muestra un valor de parámetro de ejemplo.

   ```
   {\"address\":\"0.0.0.0\",\"port\":\"9680\",\"log-level\":\"info\",\"is-HA\":false}
   ```

   - address

     La dirección de destino a la que se enviarán las métricas de Prometheus. El valor predeterminado es `localhost`.

   - puerto

     El puerto de destino al que enviarán las métricas de Prometheus. El valor predeterminado es `9680`.

   - is-HA

`true` para implementaciones de alta disponibilidad de SAP NetWeaver. Para el resto de implementaciones, el valor es `false`.

2. Diríjase a la consola SSM Distributor y abra la pestaña Owned by Amazon (Perteneciente a Amazon). Seleccione AWSObservabilityExporter-SAP-SAPHostExporterInstallAndConfigure y elija Install one time (Instalar una vez).
3. Actualice el parámetro de SSM que creó en el primer paso mediante el reemplazo de 'Argumentos adicionales' por lo siguiente:

```
{
  "SSM_EXPORTER_CONFIG": "{{ssm:<SSM_CONFIGURATIONS_PARAMETER_STORE_NAME>}}",
  "SSM_SID": "<SAP_DATABASE_SID>",
  "SSM_INSTANCES_NUM": "<instances_number seperated by comma>"
}
```

Ejemplo

```
{
  "SSM_EXPORTER_CONFIG": "{{ssm:exporter_config_paramter}}",
  "SSM_INSTANCES_NUM": "11,12,10",
  "SSM_SID": "PR1"
}
```

4. Seleccione las instancias de Amazon EC2 con aplicaciones de SAP NetWeaver y luego elija Run (Ejecutar).

> **Note**
>
> El exportador de Prometheus administra las métricas de SAP NetWeaver en un punto de conexión local. Solo los usuarios del sistema operativo de la instancia Amazon EC2 pueden acceder al punto de conexión local. Por lo tanto, una vez instalado el paquete exportador, las métricas estarán disponibles para todos los usuarios del sistema operativo. El punto de conexión local predeterminado es `localhost:9680/metrics`.

`AWSObservabilityExporter-SQLExporterInstallAndConfigure`

Puede recuperar métricas de SQL Server específicas de la carga de trabajo desde el SQL Exporter de Prometheus para Información de aplicaciones con el fin de supervisar las métricas clave.

Para utilizar Distribuidor de AWS Systems Manager para empaquetar, instalar y configurar el paquete exportador SQL independientemente de Información de aplicaciones, siga los pasos siguientes.

Prerrequisitos para usar el paquete SSM del SQL Exporter de Prometheus

- Debe contar con la versión 2.3.1550.0 de SSM Agent instalada o una posterior
- Instancia de Amazon EC2 que ejecuta SQL Server en Windows con la autenticación de usuarios de SQL Server habilitada.
- Un usuario de SQL Server con los siguientes permisos:

```
GRANT VIEW ANY DEFINITION TO
```

```
GRANT VIEW SERVER STATE TO
```

- Un secreto que contiene la cadena de conexión a la base de datos que utiliza AWS Secrets Manager. Para obtener más información acerca de la creación de secretos, consulte Crear un secreto en la Guía del usuario de AWS Secrets Manager. El secreto debe tener el siguiente formato:

```
{
  "data_source_name":"sqlserver://<username>:<password>@localhost:1433"
}
```

> **Note**
>
> Si la contraseña o el nombre de usuario contienen caracteres especiales, debe codificar los caracteres especiales en porcentaje para garantizar una conexión correcta a la base de datos.

Instale y configure el paquete **`AWSObservabilityExporter-SQLExporterInstallAndConfigure`**

El paquete `AWSObservabilityExporter-SQLExporterInstallAndConfigure` es un paquete del distribuidor de SSM que puede usar para instalar y configurar el exportador de métricas de Prometheus de SQL NetWeaver. Cuando el exportador de Prometheus envía métricas, el agente de CloudWatch se puede configurar para recuperar las métricas del servicio CloudWatch.

1. En función de sus preferencias, prepare la configuración YAML de SQL Exporter. El siguiente ejemplo de configuración tiene configurada una única métrica. Utilice la configuración de ejemplo para actualizar la configuración con métricas adicionales o crear su propia configuración.

```
---
global:
  scrape_timeout_offset: 500ms
  min_interval: 0s
  max_connections: 3
  max_idle_connections: 3
target:
  aws_secret_name: <SECRET_NAME>
  collectors:
    - mssql_standard
collectors:
  - collector_name: mssql_standard
    metrics:
    - metric_name: mssql_batch_requests
      type: counter
      help: 'Number of command batches received.'
      values: [cntr_value]
      query: |
        SELECT cntr_value
        FROM sys.dm_os_performance_counters WITH (NOLOCK)
        WHERE counter_name = 'Batch Requests/sec'
```

2. Copie el archivo de configuración YAML del SQL Exporter de Prometheus codificado como Base64 a un nuevo parámetro SSM en el Almacén de parámetros de SSM.
3. Diríjase a la consola SSM Distributor y abra la pestaña Owned by Amazon (Perteneciente a Amazon). Seleccione AWSObservabilityExporter-SQLExporterInstallandConfigure y elija Instalar una vez.
4. Sustituya los “Argumentos adicionales” por la siguiente información. `SSM_PARAMETER_NAME` es el nombre del parámetro que creó en el paso 2.

```
{
```

```
  "SSM_EXPORTER_CONFIGURATION":
    "{{ssm:<SSM_PARAMETER_STORE_NAME>}}",
    "SSM_PROMETHEUS_PORT": "9399",
    "SSM_WORKLOAD_NAME": "SQL"
}
```

5. Seleccione la instancia de Amazon EC2 con la base de datos de SQL Server y, a continuación, elija ejecutar.

Documentos de Systems Manager (SSM) de AWS utilizados por Información de aplicaciones de CloudWatch

Información de aplicaciones utiliza los documentos de SSM que se enumeran en esta sección para definir las acciones que AWS Systems Manager realiza en las instancias administradas. Estos documentos utilizan la función de `Run Command` de Systems Manager para automatizar las tareas necesarias para llevar a cabo las funciones de monitoreo de Información de aplicaciones. Información de aplicaciones mantiene los programas de ejecución de estos documentos y no se pueden modificar.

Para obtener más información sobre los documentos de SSM, consulte Documentos de AWS Systems Manager en la Guía del usuario de AWS Systems Manager.

Documentos administrados por Información de aplicaciones de CloudWatch

En la siguiente tabla se enumeran los documentos de SSM que administra Información de aplicaciones.

Nombre del documento	Descripción	Programa de ejecución
`AWSEC2-DetectWorkload`	Detecta de forma automática las aplicaciones que se ejecutan en su entorno de aplicaciones y que Información de aplicaciones puede configurar para monitorearlas.	Este documento se ejecuta a cada hora en su entorno de aplicaciones para obtener detalles actualizados de la aplicación.
`AWSEC2-CheckPerformanceCounterSets`	Comprueba si los espacios de nombres del Contador de rendimiento están habilitados	Este documento se ejecuta a cada hora en el entorno de su aplicación y solo monitorea

Nombre del documento	Descripción	Programa de ejecución
	en las instancias de Windows de Amazon EC2.	las métricas del Contador de rendimiento si los espacios de nombres correspondientes están habilitados.
`AWSEC2-ApplicationInsightsCloudwatchAgentInstallAndConfigure`	Instala y configura el Agente de CloudWatch en función de la configuración de monitoreo de los componentes de la aplicación.	Este documento se ejecuta cada 30 minutos para asegurar que la configuración del Agente de CloudWatch sea siempre precisa y esté actualizada. El documento también se ejecuta inmediatamente después de realizar un cambio en la configuración de monitoreo de la aplicación, como añadir o eliminar métricas o actualizar las configuraciones de registro.

Documentos administrados por AWS Systems Manager

Información de aplicaciones de CloudWatch utiliza los siguientes documentos y Systems Manager los administra.

`AWS-ConfigureAWSPackage`

La Información de aplicaciones utiliza este documento para instalar y desinstalar los paquetes del distribuidor exportador Prometheus, para recopilar métricas específicas de las cargas de trabajo y permitir un monitoreo exhaustivo de las cargas de trabajo en las instancias de Amazon EC2 de los clientes. Información de aplicaciones de CloudWatch instala los paquetes del distribuidor exportador Prometheus solo si la carga de trabajo de destino correlacionada se ejecuta en su instancia.

La siguiente tabla enumera los paquetes del distribuidor exportador Prometheus y las cargas de trabajo de destino correlacionadas.

Nombre del paquete del distribuidor exportador Prometheus	Carga de trabajo de destino
`AWSObservabilityExporter-HAClusterExporterInstallAndConfigure`	SAP HANA HA
`AWSObservabilityExporter-JMXExporterInstallAndConfigure`	Java/JMX
`AWSObservabilityExporter-SAP-HANADBExporterInstallAndConfigure`	SAP HANA
`AWSObservabilityExporter-SAP-SAPHostExporterInstallAndConfigure`	NetWeaver
`AWSObservabilityExporter-SQLExporterInstallAndConfigure`	SQL Server (Windows) y SAP ASE (Linux)

`AmazonCloudWatch-ManageAgent`

La Información de aplicaciones utiliza este documento para administrar el estado y la configuración del Agente de CloudWatch en sus instancias y para recopilar métricas y registros internos a nivel del sistema de las instancias de Amazon EC2 en todos los sistemas operativos.

Introducción a Información de aplicaciones de Amazon CloudWatch

Para comenzar a utilizar CloudWatch, verifique que se cumplan los requisitos previos que se describen a continuación y que ha creado una política de IAM. A continuación, puede comenzar a utilizar el enlace de la consola para habilitar Información de aplicaciones de CloudWatch. Para configurar los recursos de la aplicación, siga los pasos que se indican en Instalación, configuración y administración de la aplicación para el monitoreo.

Contenido

- Política de IAM
- Permisos de rol de IAM para la incorporación de aplicaciones basadas en cuentas
- Instalación, configuración y administración de la aplicación para el monitoreo

Acceso a Información de aplicaciones de CloudWatch

Puede acceder y administrar Información de aplicaciones de CloudWatch a través de una de las siguientes interfaces:

- Consola de CloudWatch. Para agregar monitores para su aplicación, elija Información de aplicaciones debajo de Información en el panel de navegación izquierdo de la consola de CloudWatch. Una vez configurada la aplicación, puede utilizar la consola de CloudWatch para ver y analizar los problemas detectados.
- AWS Command Line Interface (AWS CLI). Puede utilizar el AWS CLI para acceder a las operaciones de la API de AWS. Para obtener más información, consulte Instalación de AWS Command Line Interface en la Guía del usuario de AWS Command Line Interface. Para obtener más información sobre la API de Información de aplicaciones, consulte la Referencia de la API de Información de aplicaciones de Amazon CloudWatch.

Requisitos previos

Debe completar los siguientes requisitos previos para configurar una aplicación con Información de aplicaciones de CloudWatch:

- Habilitación de AWS Systems Manager: instale Systems Manager Agent (SSM Agent) en sus instancias de Amazon EC2 y habilite las instancias para SSM. Para obtener información acerca de la instalación de SSM Agent, consulte Configuración de AWS Systems Manager en la Guía del usuario de AWS Systems Manager.
- Rol de instancia de EC2: debe adjuntar los siguientes roles de instancia de Amazon EC2 para habilitar Systems Manager
  - Debe adjuntar el rol de `AmazonSSMManagedInstanceCore` para habilitar Systems Manager. Para obtener más información, consulte Ejemplos de políticas basadas en identidades de AWS Systems Manager.
  - Debe adjuntar la política `CloudWatchAgentServerPolicy` para permitir que las métricas y registros de instancias se emitan a través de CloudWatch. Consulte Creación de roles de IAM y usuarios para utilizarlos con el agente de CloudWatch para obtener más información.

- Grupos de recursos de AWS: para incorporar las aplicaciones a Información de aplicaciones de CloudWatch, cree un grupo de recursos que incluya todos los recursos de AWS asociados que se utilizan en la pila de aplicaciones. Esto incluye los equilibradores de carga de aplicación, las instancias de Amazon EC2 con el front-end de IIS y web, las capas de nodos de trabajo de .NET y la base de datos SQL Server. Para obtener más información sobre los componentes de las aplicaciones y los conjuntos de tecnologías compatibles con Información de aplicaciones, consulte Componentes de aplicaciones admitidos. Información de aplicaciones de CloudWatch incluye automáticamente grupos de escalado automático con las mismas etiquetas o pilas de CloudFormation como su grupo de recursos, ya que los grupos de recursos no son compatibles con los grupos de escalado automático. Para obtener más información, consulte Introducción a AWSResource Groups.
- Permisos de IAM: en el caso de los usuarios que no sean administradores, debe crear una política de AWS Identity and Access Management (IAM) a fin de que Información de aplicaciones cree un rol vinculado al servicio y este se asocie a la identidad del usuario. Para obtener información acerca de la creación de una política de IAM, consulte Política de IAM.
- Rol vinculado al servicio: Información de aplicaciones utiliza roles vinculados al servicio de AWS Identity and Access Management (IAM). Al crear su primera aplicación de Información de aplicaciones en la consola, Información de aplicaciones le crea el rol vinculado a un servicio. Para obtener más información, consulte Uso de roles vinculados a un servicio para CloudWatch Application Insights.
- Compatibilidad con las métricas del contador de rendimiento para las instancias EC2 de Windows: para monitorear las métricas del contador de rendimiento en las instancias de Amazon EC2 de Windows, los contadores de rendimiento deben estar instalados en las instancias. Para las métricas del contador de desempeño y los nombres del conjunto de contadores de desempeño correspondientes, consulte Métricas del contador de desempeño. Para obtener más información acerca de los contadores de desempeño, consulte Contadores de desempeño.
- Agente de Amazon CloudWatch: Información de aplicaciones instala y configura el agente de CloudWatch. Si tiene instalado el agente de CloudWatch, Información de aplicaciones conserva su configuración. Para evitar un conflicto de fusión, elimine la configuración de los recursos que desee utilizar en Información de aplicaciones del archivo de configuración del agente de CloudWatch existente. Para obtener más información, consulte Cree o edite de forma manual el archivo de configuración del agente de CloudWatch.

Política de IAM

Para utilizar Información de aplicaciones de CloudWatch, cree una política de AWS Identity and Access Management (IAM) y adjúntela al usuario, grupo o rol. Para obtener más información acerca de los usuarios, los grupos, los roles y los roles, consulte Identidades IAM (usuarios, grupos de usuarios y roles). La política de IAM define los permisos del usuario.

Para crear una política de IAM mediante la consola, realice el siguiente procedimiento.

Para crear una política de IAM mediante la consola de IAM, siga los pasos a continuación.

1. Vaya a la consola de IAM. En el panel de navegación izquierdo, seleccione Policies (Políticas).
2. En la parte superior de la página, seleccione Create policy (Crear política).
3. Seleccione la pestaña JSON.
4. Copie y pegue el siguiente documento JSON en la pestaña JSON.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "applicationinsights:*",
                "iam:CreateServiceLinkedRole",
                "iam:ListRoles",
                "resource-groups:ListGroups"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

5. Elija Review Policy (Revisar política).
6. En Nombre escriba un nombre para la política; por ejemplo, “AppInsightsPolicy”. Si lo desea, en Description (Descripción) escriba una descripción.
7. Seleccione Create Policy (Crear política).
8. En el panel de navegación izquierdo, seleccione Grupos de usuarios, Usuarios o Roles.
9. Elija el nombre del grupo de usuarios, usuario o rol al que desea asociar la política.
10. Seleccione Add permissions (Añadir permisos).

11. Seleccione Asociar directamente las políticas existentes.
12. Busque la política que acaba de crear y seleccione la casilla de verificación situada a la izquierda del nombre de la política.
13. A continuación, seleccione Next: Review (Siguiente: Revisar).
14. Asegúrese de que se muestra la política correcta y seleccione Add permissions (Añadir permisos).
15. Asegúrese de iniciar sesión con el usuario asociado a la política que acaba de crear cuando utilice Información de aplicaciones de CloudWatch.

Para crear una política de IAM mediante AWS CLI

Para crear una política de IAM mediante AWS CLI, ejecute la operación create-policy desde la línea de comandos, con el documento en formato JSON mencionado como archivo en su carpeta actual.

Para crear una política de IAM mediante AWS Tools for Windows PowerShell

Para crear una política de IAM mediante AWS Tools for Windows PowerShell, ejecute el cmdlt New-IAMPolicy con el documento en formato JSON mencionado como un archivo en la carpeta actual.

Permisos de rol de IAM para la incorporación de aplicaciones basadas en cuentas

Si quiere incorporar todos los recursos de su cuenta y decide no utilizar la política administrada de Información de aplicaciones para acceder completamente a la funcionalidad de Información de aplicaciones, debe adjuntar los siguientes permisos a su rol de IAM para que Información de aplicaciones pueda descubrir todos los recursos de su cuenta:

```
"ec2:DescribeInstances"
"ec2:DescribeNatGateways"
"ec2:DescribeVolumes"
"ec2:DescribeVPCs"
"rds:DescribeDBInstances"
"rds:DescribeDBClusters"
"sqs:ListQueues"
"elasticloadbalancing:DescribeLoadBalancers"
"autoscaling:DescribeAutoScalingGroups"
"lambda:ListFunctions"
"dynamodb:ListTables"
"s3:ListAllMyBuckets"
"sns:ListTopics"
```

```
"states:ListStateMachines"
"apigateway:GET"
"ecs:ListClusters"
"ecs:DescribeTaskDefinition"
"ecs:ListServices"
"ecs:ListTasks"
"eks:ListClusters"
"eks:ListNodegroups"
"fsx:DescribeFileSystems"
"route53:ListHealthChecks"
"route53:ListHostedZones"
"route53:ListQueryLoggingConfigs"
"route53resolver:ListFirewallRuleGroups"
"route53resolver:ListFirewallRuleGroupAssociations"
"route53resolver:ListResolverEndpoints"
"route53resolver:ListResolverQueryLogConfigs"
"route53resolver:ListResolverQueryLogConfigAssociations"
"logs:DescribeLogGroups"
"resource-explorer:ListResources"
```

Instalación, configuración y administración de la aplicación para el monitoreo

En esta sección se detallan los pasos que tiene que seguir para instalar, configurar y administrar la aplicación de Información de aplicaciones de CloudWatch mediante la consola, AWS CLI y AWS Tools for Windows PowerShell.

Temas

- Instale, configure y administre la aplicación para el monitoreo desde la consola de CloudWatch
- Instale, configure y administre la aplicación para el monitoreo mediante la línea de comandos.
- Información de aplicaciones CloudWatch Events y notificaciones para problemas detectados

Instale, configure y administre la aplicación para el monitoreo desde la consola de CloudWatch

En esta sección se detallan los pasos a seguir para instalar, configurar y administrar la aplicación para el monitoreo desde la consola de CloudWatch.

Procedimientos de la consola

- Agregue y configure una aplicación
- Habilitación del monitoreo de recursos de Información de aplicaciones para Amazon ECS y Amazon EKS

- Desactive el monitoreo para un componente de aplicación
- Eliminar una aplicación

Agregue y configure una aplicación

Adición y configuración de una aplicación desde la consola de CloudWatch

Siga estos pasos para comenzar a utilizar Información de aplicaciones de CloudWatch desde la consola de CloudWatch.

1. Inicio. Abra la página de inicio de la consola de CloudWatch. Desde el panel de navegación izquierdo, en Insights, elija Application Insights. En la página que se abre, podrá visualizar la lista de aplicaciones que se supervisan con Información de aplicaciones de CloudWatch, además del estado de supervisión.
2. Agregar una aplicación. Para configurar el monitoreo de la aplicación, elija Add an application (Agregar una aplicación). Cuando elija Add an application (Agregar una aplicación), deberá Choose Application Type (Elegir el tipo de aplicación).

   - Aplicación basada en Resource Groups. Al seleccionar esta opción, puede elegir los Resource Groups de esta cuenta que se van a monitorear. Para usar varias aplicaciones en un componente, debe usar la supervisión basada en grupos de recursos.
   - Aplicación basada en cuentas. Al seleccionar esta opción, podrá monitorear todos los recursos de esta cuenta. Si desea monitorear todos los recursos de una cuenta, le recomendamos esta opción en lugar de la opción basada en grupos de recursos porque el proceso de incorporación de aplicaciones es más rápido.

   > **Note**
   >
   > No es posible combinar el monitoreo basado en grupos de recursos con el monitoreo basado en cuentas mediante Información de aplicaciones. Para cambiar el tipo de aplicación, debe eliminar todas las aplicaciones que se están supervisando y Choose Application Type (Elegir el tipo de aplicación).

   Cuando agrega su primera aplicación para realizar la supervisión, Información de aplicaciones de CloudWatch crea un rol vinculado a un servicio en su cuenta, lo que le otorga permisos a Información de aplicaciones para llamar a otros servicios de AWS en su nombre. Para

obtener más información sobre el rol vinculado a un servicio creado que crea Información de aplicaciones en su cuenta, consulte Uso de roles vinculados a un servicio para CloudWatch Application Insights.

3. Resource-based application monitoring

    1. Seleccionar el grupo de recursos.En la página Specify application details (Especificar detalles de la aplicación), seleccione el grupo de recursos de AWS que contiene los recursos de aplicaciones de la lista desplegable. Estos recursos incluyen los servidores front-end, balanceadores de carga, grupos de Auto Scaling y servidores de bases de datos.

        Si no ha creado un grupo de recursos para su aplicación, puede crear uno si elige Create new resource group (Crear nuevo grupo de recursos). Para obtener más información acerca de Resource Groups, consulte la Guía del usuario de AWS Resource Groups.

    2. Monitoreo de CloudWatch Events. Marque el casillero para integrar el monitoreo de Información de aplicaciones con Eventos de CloudWatch y obtener información de Amazon EBS, Amazon EC2, AWS CodeDeploy, Amazon ECS, API y notificaciones de AWS Health, Amazon RDS, Amazon S3 y AWS Step Functions.
    3. Integrar con Systems Manager OpsCenter de AWS. Para ver y recibir notificaciones cuando se detecten problemas en las aplicaciones seleccionadas, seleccione el casillero Generate Systems Manager OpsCenter OpsItems for remedial actions (Generar OpsItems de OpsCenter de Systems Manager para crear acciones correctivas). Para realizar un seguimiento de las operaciones que se realizan para resolver elementos de trabajo operativos (OpsItems) relacionados con sus recursos de AWS, proporcione el ARN del tema de SNS.
    4. Etiquetas: opcional. Información de aplicaciones de eventos de CloudWatch es compatible con Resource Groups basados en etiquetas y en CloudFormation (con la excepción de los grupos de escalado automático). Para obtener más información, consulte Uso de Tag Editor.
    5. Elija Siguiente.

        Se generará un ARN para la aplicación en el siguiente formato.

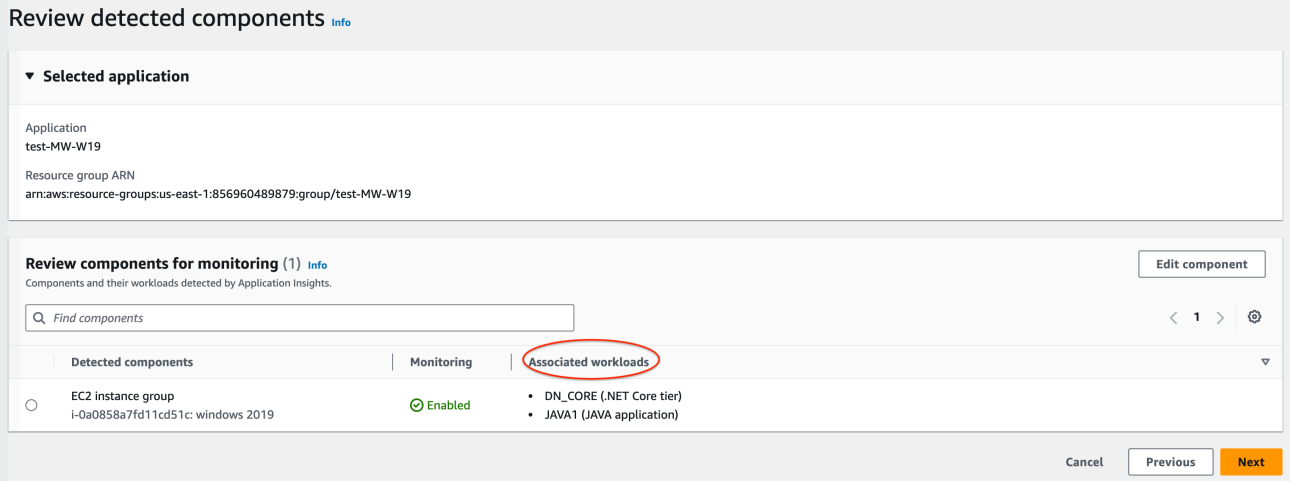
        ```
        arn:partition:applicationinsights:region:account-id:application/resource-group/resource-group-name
        ```

        Ejemplo

```
arn:aws:applicationinsights:us-east-1:123456789012:application/resource-group/
my-resource-group
```

6. En la página Revisar los componentes detectados, en Revisar los componentes para su supervisión, la tabla muestra los componentes detectados y sus cargas de trabajo detectadas asociadas.

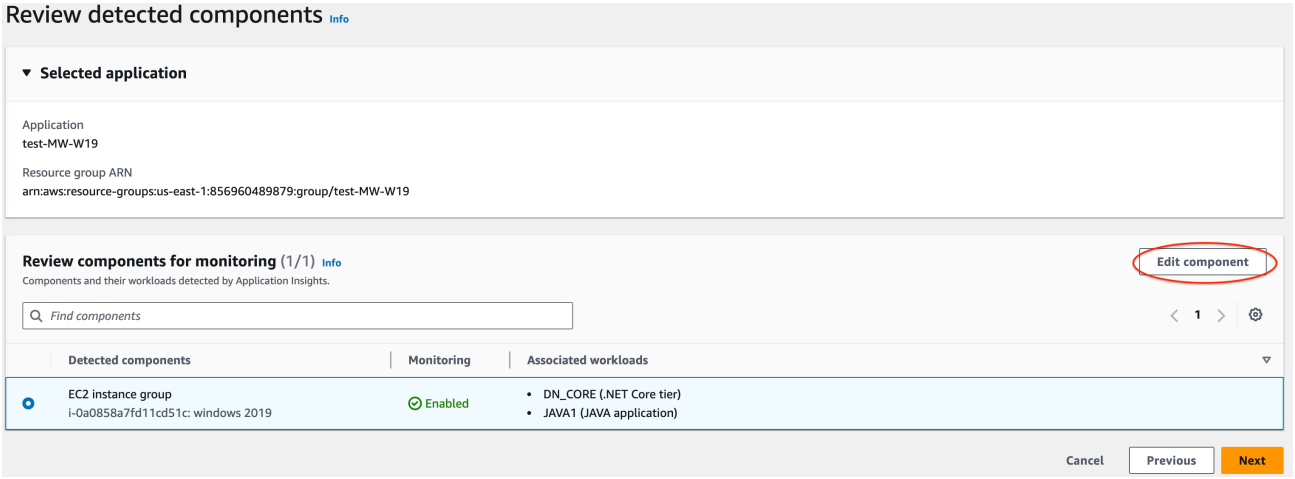
> **Note**
>
> En el caso de los componentes que admiten varias cargas de trabajo personalizadas, puede supervisar hasta cinco cargas de trabajo para cada componente. Estas cargas de trabajo se supervisarán por separado del componente.



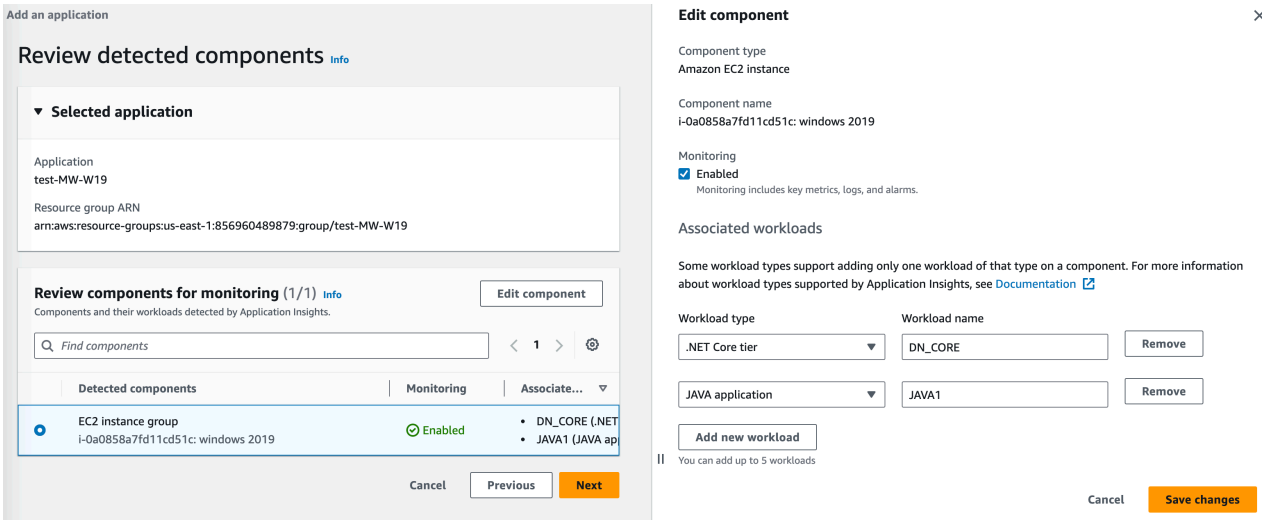
En Cargas de trabajo asociadas, hay varios mensajes posibles que aparecen si una carga de trabajo no aparece en la lista.

- No se pudieron detectar las cargas de trabajo: se produjo un problema al intentar detectar las cargas de trabajo. Asegúrese de realizar los pasos que se indican en Requisitos previos. Si necesita añadir cargas de trabajo, elija Editar componente.
- No se detectaron cargas de trabajo: no detectamos ninguna carga de trabajo. Es posible que necesite añadir cargas de trabajo. Para ello, elija Editar componente.
- No aplicable: el componente no admite cargas de trabajo personalizadas y se supervisará con métricas, alarmas y registros predeterminados. No puede añadir cargas de trabajo a estos componentes.

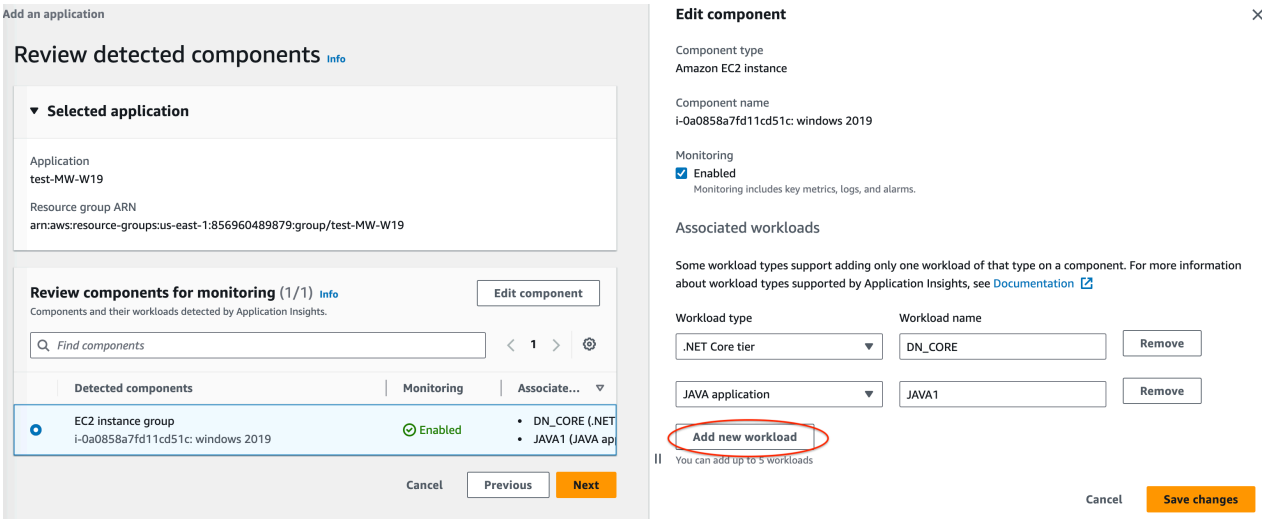
7. Para editar un componente, selecciónelo y, a continuación, elija Editar componente. Se abre un panel lateral con las cargas de trabajo detectadas en el componente. En este panel, puede editar los detalles del componente y añadir nuevas cargas de trabajo.



- Para editar el tipo o el nombre de la carga de trabajo, utilice la lista desplegable.

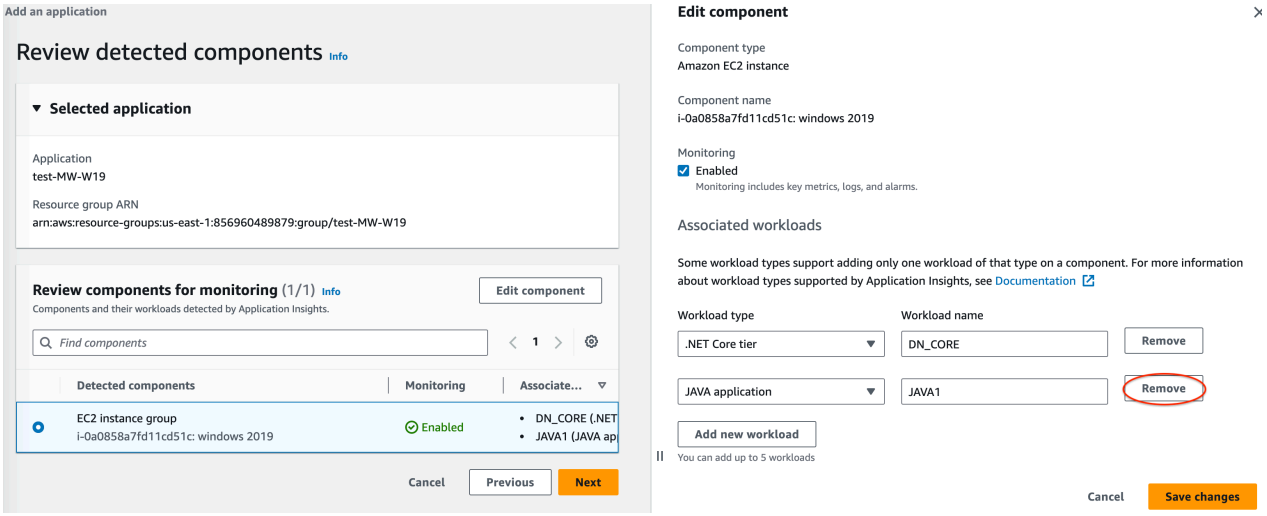


- Para añadir una nueva carga de trabajo al componente, seleccione Añadir nueva carga de trabajo.

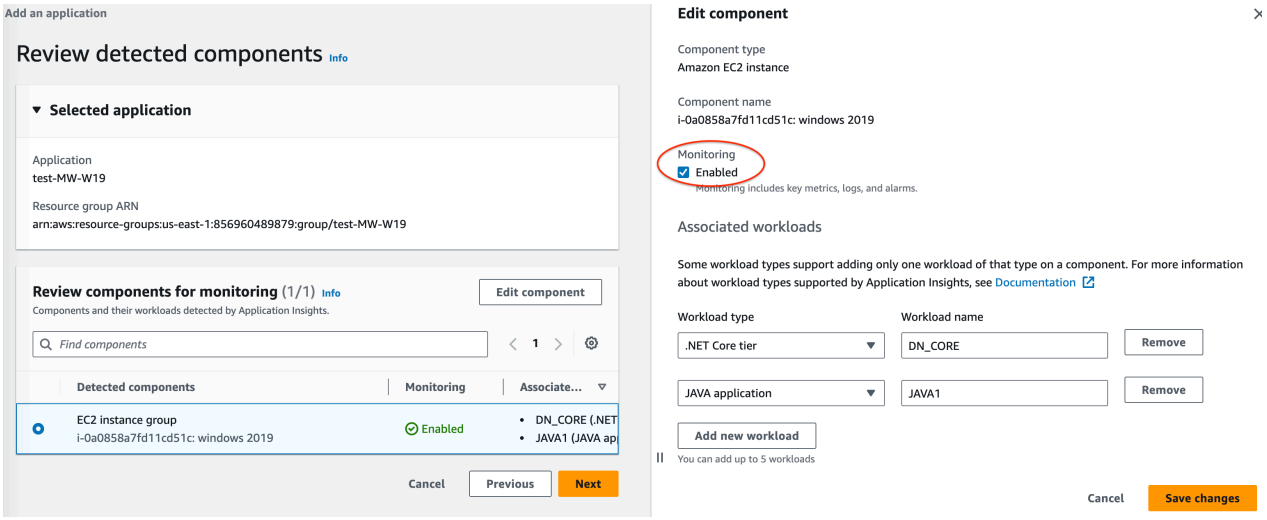


- Si no aparece Añadir nueva carga de trabajo, este componente no admite varias cargas de trabajo.
- Si el encabezado Cargas de trabajo asociadas no aparece, este componente no admite cargas de trabajo personalizadas.

- Para eliminar una carga de trabajo, seleccione Eliminar junto a la carga de trabajo que desea eliminar de la supervisión.



- Para deshabilitar la monitorización de todo el componente, desactive la casilla Monitorización.



- Cuando haya terminado de editar el componente, seleccione Guardar cambios en la esquina inferior derecha. Cualquier cambio en las cargas de trabajo de un componente está visible en la tabla Revisar los componentes para su supervisión, en Cargas de trabajo asociadas.

8. En la página Revisar componentes detectados, seleccione Siguiente.
9. La página Especificar detalles del componente incluye todos los componentes con cargas de trabajo asociadas personalizables del paso anterior.

> **Note**
>
> Si el encabezado de un componente tiene una etiqueta opcional, los detalles adicionales de las cargas de trabajo de ese componente son opcionales.

   Si un componente no aparece en esta página, no tiene ningún detalle adicional que pueda especificarse en este paso.

10. Elija Siguiente.
11. En la página Revisar y enviar, revise todos los detalles de los componentes y la carga de trabajo supervisados.
12. Seleccione Submit (Enviar).

Account-based application monitoring

1. Nombre de la aplicación. Ingrese un nombre para la aplicación basada en cuenta.

2. Monitoreo automatizado de nuevos recursos. Información de aplicaciones utiliza, de forma predeterminada, la configuración recomendada para configurar el monitoreo de los componentes de recursos que se agregan a su cuenta después de incorporar la aplicación. Puede excluir el monitoreo de los recursos agregados después de incorporar la aplicación si desactiva el casillero.
3. Monitoreo de CloudWatch Events. Marque el casillero para integrar el monitoreo de Información de aplicaciones con Eventos de CloudWatch y obtener información de Amazon EBS, Amazon EC2, AWS CodeDeploy, Amazon ECS, API y notificaciones de AWS Health, Amazon RDS, Amazon S3 y AWS Step Functions.
4. Integrar con Systems Manager OpsCenter de AWS. Para ver y recibir notificaciones cuando se detecten problemas en las aplicaciones seleccionadas, seleccione el casillero Generate Systems Manager OpsCenter OpsItems for remedial actions (Generar OpsItems de OpsCenter de Systems Manager para crear acciones correctivas). Para realizar un seguimiento de las operaciones que se realizan para resolver elementos de trabajo operativos (OpsItems) relacionados con sus recursos de AWS, proporcione el ARN del tema de SNS.
5. Etiquetas: opcional. Información de aplicaciones de eventos de CloudWatch es compatible con Resource Groups basados en etiquetas y en CloudFormation (con la excepción de los grupos de escalado automático). Para obtener más información, consulte Uso de Tag Editor.
6. Recursos detectados. Todos los recursos detectados en su cuenta se agregan a esta lista. Si Información de aplicaciones no puede detectar todos los recursos de su cuenta, aparece un mensaje de error en la parte superior de la página. Este mensaje incluye un enlace a la documentación sobre cómo agregar los permisos necesarios.
7. Elija Siguiente.

   Se generará un ARN para la aplicación en el siguiente formato.

   ```
   arn:partition:applicationinsights:region:account-id:application/
   TBD/application-name
   ```

   Ejemplo

   ```
   arn:aws:applicationinsights:us-east-1:123456789012:application/TBD/my-
   application
   ```

4. Después de enviar la configuración de monitoreo de aplicaciones, se le dirigirá a la página de detalles de la aplicación, donde podrá ver el Application summary (Resumen de aplicaciones), la lista de Monitored components (Componentes monitoreados) y Unmonitored components (Componentes no monitoreados) y, si selecciona las pestañas situadas junto a Components (Componentes), el Configuration history (Historial de configuración), los Log patterns (Patrones de registro) y cualquiera de las Tags (Etiquetas) que haya aplicado.

   Para ver las observaciones de la aplicación, elija View Insights (Visualizar observaciones).

   Puede actualizar sus selecciones para el monitoreo y la integración de CloudWatch Events con Systems Manager OpsCenter de AWS si eligeEdit (Editar).

   Debajo de Components (Componentes), puede seleccionar el menú de Actions (Acciones) para crear, modificar o desagrupar un grupo de instancias.

   Puede administrar el monitoreo de componentes, incluidos el nivel de aplicación, los grupos de registros, los registros de eventos, las métricas y las alarmas personalizadas, si selecciona la viñeta junto a un componente y Manage monitoring (Administrar el monitoreo).

Habilitación del monitoreo de recursos de Información de aplicaciones para Amazon ECS y Amazon EKS

Puede habilitar Información de aplicaciones para que monitoree las aplicaciones y microservicios en contenedores desde la consola de Información de contenedores. Información de aplicaciones puede monitorear los siguientes recursos:

- Clústeres de Amazon ECS
- Servicios de Amazon ECS
- Tareas de Amazon ECS
- Clústeres de Amazon EKS

Cuando Información de aplicaciones está habilitada, proporciona métricas y registros recomendados, detecta posibles problemas, genera Eventos de CloudWatch y crea paneles automáticos para sus aplicaciones y microservicios en contenedores.

Puede habilitar Información de aplicaciones para recursos en contenedores desde las consolas Información de contenedores o Información de aplicaciones.

Habilitación de Información de aplicaciones desde la consola de Información de contenedores

Desde la consola de Información de contenedores, en el panel de Monitoreo del rendimiento de Información de contenedores, elija Configuración automática de Información de aplicaciones. Cuando Información de aplicaciones está habilitada, muestra detalles de los problemas detectados.

Habilitación de Información de aplicaciones desde la consola de Información de aplicaciones

Cuando aparecen clústeres de ECS en la lista de componentes, Información de aplicaciones habilita el monitoreo adicional de contenedores con Información de contenedores de forma automática.

En el caso de los clústeres de EKS, usted puede habilitar el monitoreo adicional con Información de contenedores para proporcionar información de diagnóstico, como errores de reinicio de contenedores, para ayudarlo a aislar y resolver problemas. Se requieren pasos adicionales para configurar Información de contenedores para EKS. Consulte Configuración de Información de contenedores en Amazon EKS y Kubernetes para obtener información sobre los pasos para configurar Información de contenedores en EKS.

El monitoreo adicional de EKS con Información de contenedores es compatible con instancias Linux con EKS.

Para obtener más información sobre la compatibilidad de Información de contenedores para clústeres ECS y EKS, consulte Información de contenedores.

Desactive el monitoreo para un componente de aplicación

Para desactivar el monitoreo de un componente de la aplicación, seleccione el componente para el que desea desactivar el monitoreo en la página Application Details (Detalles de la aplicación). Elija Actions (Acciones) y luego, Remove from monitoring (Detener el monitoreo).

Eliminar una aplicación

Para eliminar una aplicación, desde el panel de CloudWatch, en el panel de navegación izquierdo de CloudWatch, elija Application Insights en Insights. Seleccione la aplicación que desea eliminar. Debajo de Actions (Acciones), elija Delete application (Eliminar aplicación). Esto elimina la monitorización y elimina todos los monitores guardados para los componentes de la aplicación. Los recursos de la aplicación no se eliminan.

Instale, configure y administre la aplicación para el monitoreo mediante la línea de comandos.

En esta sección se detallan los pasos a seguir para instalar, configurar y administrar la aplicación para el monitoreo mediante la AWS CLI y AWS Tools for Windows PowerShell.

Procedimientos de línea de comandos

- Adición y administración de una aplicación
- Administración y actualización del monitoreo
- Configurar la supervisión de los grupos de disponibilidad siempre activos de SQL
- Configurar la monitoreo para RDS de MySQL
- Configurar la supervisión para EC2 de MySQL
- Configuración del monitoreo para RDS de PostgreSQL
- Configurar el monitoreo para EC2 de PostgreSQL
- Configuración del monitoreo para RDS de Oracle
- Configuración del monitoreo para EC2 de Oracle.

Adición y administración de una aplicación

Puede añadir, obtener información, administrar y configurar su aplicación Información de aplicaciones mediante la línea de comandos.

Temas

- Adición de una aplicación
- Descripción de una aplicación
- Enumeración de componentes en una aplicación
- Descripción de un componente
- Agrupación de recursos similares en un componente personalizado
- Cancelación de agrupación de un componente personalizado
- Actualización de una aplicación
- Actualización de un componente personalizado

Adición de una aplicación

Agregue una aplicación mediante AWS CLI

Utilice el siguiente comando para usar el AWS CLI a fin de agregar una aplicación para su grupo de recursos denominado `my-resource-group` y mantenga OpsCenter habilitado para entregar el opsItem creado a `arn:aws:sns:us-east-1:123456789012:MyTopic` un tema ARN de SNS.

```
aws application-insights create-application --resource-group-name my-resource-group --
ops-center-enabled --ops-item-sns-topic-arn arn:aws:sns:us-east-1:123456789012:MyTopic
```

Agregue una aplicación mediante AWS Tools for Windows PowerShell

Utilice el siguiente comando para utilizar la AWS Tools for Windows PowerShell para agregar una aplicación para su grupo de recursos denominado `my-resource-group`, con OpsCenter habilitado para entregar el opsItem creado a `arn:aws:sns:us-east-1:123456789012:MyTopic` un tema ARN de SNS.

```
New-CWAIApplication -ResourceGroupName my-resource-group -OpsCenterEnabled true -
OpsItemSNSTopicArn arn:aws:sns:us-east-1:123456789012:MyTopic
```

Descripción de una aplicación

Descripción de una aplicación mediante la AWS CLI

Utilice el siguiente comando para usar la AWS CLI a fin de describir una aplicación creada en un grupo de recursos denominado `my-resource-group`.

```
aws application-insights describe-application --resource-group-name my-resource-group
```

Describa una aplicación mediante AWS Tools for Windows PowerShell

Utilice el siguiente comando para usar la AWS Tools for Windows PowerShell a fin de describir una aplicación creada en un grupo de recursos denominado `my-resource-group`.

```
Get-CWAIApplication -ResourceGroupName my-resource-group
```

Enumeración de componentes en una aplicación

Enumeración de componentes en una aplicación mediante AWS CLI

Utilice el siguiente comando para usar AWS CLI a fin de enumerar los componentes creados en un grupo de recursos denominado `my-resource-group`.

```
aws application-insights list-components --resource-group-name my-resource-group
```

Enumere componentes en una aplicación mediante AWS Tools for Windows PowerShell

Utilice el siguiente comando para usar AWS Tools for Windows PowerShell a fin de enumerar los componentes creados en un grupo de recursos denominado `my-resource-group`.

```
Get-CWAIComponentList -ResourceGroupName my-resource-group
```

Descripción de un componente

Describa un componente mediante AWS CLI

Puede utilizar el siguiente comando de AWS CLI para describir un componente denominado `my-component` que pertenece a una aplicación creada en un grupo de recursos denominado `my-resource-group`.

```
aws application-insights describe-component --resource-group-name my-resource-group --
component-name my-component
```

Describa un componente mediante AWS Tools for Windows PowerShell

Puede utilizar el siguiente comando de AWS Tools for Windows PowerShell para describir un componente denominado `my-component` que pertenece a una aplicación creada en un grupo de recursos denominado `my-resource-group`.

```
Get-CWAIComponent -ComponentName my-component -ResourceGroupName my-resource-group
```

Agrupación de recursos similares en un componente personalizado

Le recomendamos que agrupe recursos similares, como instancias de servidor web .NET, en componentes personalizados para facilitar la incorporación y para una mejor monitorización e información. En la actualidad, Información de aplicaciones de CloudWatch admite grupos personalizados para instancias EC2.

Para agrupar recursos en un componente personalizado mediante AWS CLI

Utilice el siguiente comando para usar AWS CLI para agrupar tres instancias (`arn:aws:ec2:us-east-1:123456789012:instance/i-11111`, `arn:aws:ec2:us-east-1:123456789012:instance/i-22222` y `arn:aws:ec2:us-east-1:123456789012:instance/i-33333`) en un componente personalizado denominado

`my-component` para una aplicación creada para el grupo de recursos denominado `my-resource-group`.

```
aws application-insights create-component --resource-group-name my-
resource-group --component-name my-component --resource-list arn:aws:ec2:us-
east-1:123456789012:instance/i-11111 arn:aws:ec2:us-east-1:123456789012:instance/
i-22222 arn:aws:ec2:us-east-1:123456789012:instance/i-33333
```

Para agrupar recursos en un componente personalizado mediante AWS Tools for Windows PowerShell

Utilice el siguiente comando para usar AWS Tools for Windows PowerShell a fin de agrupar tres instancias (`arn:aws:ec2:us-east-1:123456789012:instance/i-11111`, `arn:aws:ec2:us-east-1:123456789012:instance/i-22222` y `arn:aws:ec2:us-east-1:123456789012:instance/i-33333`) en un componente personalizado denominado `my-component` para una aplicación creada para el grupo de recursos denominado `my-resource-group`.

```
New-CWAIComponent -ResourceGroupName my-resource-group -ComponentName my-component
 -ResourceList arn:aws:ec2:us-east-1:123456789012:instance/i-11111,arn:aws:ec2:us-
east-1:123456789012:instance/i-22222,arn:aws:ec2:us-east-1:123456789012:instance/
i-33333
```

Cancelación de agrupación de un componente personalizado

Para desagrupar un componente personalizado mediante AWS CLI

Utilice el siguiente comando para usar AWS CLI para desagrupar un componente personalizado denominado `my-component` en una aplicación creada en el grupo de recursos denominado `my-resource-group`.

```
aws application-insights delete-component --resource-group-name my-resource-group --
component-name my-new-component
```

Para desagrupar un componente personalizado mediante AWS Tools for Windows PowerShell

Utilice el siguiente comando para usar AWS Tools for Windows PowerShell para desagrupar un componente personalizado denominado `my-component` en una aplicación creada en el grupo de recursos denominado `my-resource-group`.

```
Remove-CWAIComponent -ComponentName my-component -ResourceGroupName my-resource-group
```

Actualización de una aplicación

Actualice una aplicación mediante AWS CLI

Mediante el siguiente comando, puede utilizar AWS CLI para actualizar una aplicación a fin de generar OpsItems de OpsCenter Systems Manager de AWS para los problemas detectados con la aplicación y para asociar los OpsItems creados al tema de SNS `arn:aws:sns:us-east-1:123456789012:MyTopic`.

```
aws application-insights update-application --resource-group-name my-resource-group --
ops-center-enabled --ops-item-sns-topic-arn arn:aws:sns:us-east-1:123456789012:MyTopic
```

Actualización de una aplicación mediante Tools for Windows PowerShell de AWS

Mediante el siguiente comando, puede utilizar AWS Tools for Windows PowerShell para actualizar una aplicación a fin de generar OpsItems de OpsCenter de Systems Manager de AWS para los problemas detectados con la aplicación y para asociar los OpsItems creados al tema de SNS `arn:aws:sns:us-east-1:123456789012:MyTopic`.

```
Update-CWAIApplication -ResourceGroupName my-resource-group -OpsCenterEnabled true -
OpsItemSNSTopicArn arn:aws:sns:us-east-1:123456789012:MyTopic
```

Actualización de un componente personalizado

Actualice un componente personalizado mediante AWS CLI

Mediante el siguiente comando, puede utilizar AWS CLI para actualizar un componente personalizado denominado `my-component` con un nuevo nombre de componente, `my-new-component`, y un grupo de instancias actualizado.

```
aws application-insights update-component --resource-group-name my-resource-
group --component-name my-component --new-component-name my-new-component --
resource-list arn:aws:ec2:us-east-1:123456789012:instance/i-44444 arn:aws:ec2:us-
east-1:123456789012:instance/i-55555
```

Actualización de un componente personalizado con Tools for Windows PowerShell de AWS

Mediante el siguiente comando, puede utilizar AWS Tools for Windows PowerShell para actualizar un componente personalizado denominado `my-component` con un nuevo nombre de componente, `my-new-component`, y un grupo de instancias actualizado.

```
Update-CWAIComponent -ComponentName my-component -NewComponentName my-new-component -ResourceGroupName my-resource-group -ResourceList arn:aws:ec2:us-east-1:123456789012:instance/i-44444,arn:aws:ec2:us-east-1:123456789012:instance/i-55555
```

Administración y actualización del monitoreo

Puede administrar y actualizar la monitorización de su aplicación Información de aplicaciones mediante la línea de comandos.

Temas

- Enumeración de problemas con la aplicación
- Descripción de un problema de la aplicación
- Descripción de las anomalías o los errores asociados a un problema
- Descripción de una anomalía o un error con la aplicación
- Descripción de las configuraciones de monitorización de un componente
- Descripción de la configuración de monitorización recomendada de un componente
- Actualización de las configuraciones de monitorización de un componente
- Eliminación de un grupo de recursos especificado del monitoreo de Información de aplicaciones

Enumeración de problemas con la aplicación

Enumere problemas con la aplicación mediante AWS CLI

Utilice el siguiente comando para usar AWS CLI para enumerar los problemas de la aplicación detectados entre 1000 y 10 000 milisegundos desde la fecha de inicio de Unix para una aplicación creada en un grupo de recursos denominado `my-resource-group`.

```
aws application-insights list-problems --resource-group-name my-resource-group --start-time 1000 --end-time 10000
```

Enumeración de problemas con la aplicación mediante Tools for Windows PowerShell de AWS

Utilice el siguiente comando para usar AWS Tools for Windows PowerShell para enumerar los problemas de la aplicación detectados entre 1000 y 10 000 milisegundos desde la fecha de inicio de Unix para una aplicación creada en un grupo de recursos denominado `my-resource-group`.

```
$startDate = "8/6/2019 3:33:00"
$endDate = "8/6/2019 3:34:00"
Get-CWAIProblemList -ResourceGroupName my-resource-group -StartTime $startDate -
EndTime $endDate
```

Descripción de un problema de la aplicación

Describa un problema de la aplicación mediante AWS CLI

Utilice el siguiente comando para usar AWS CLI a fin de describir un problema con el ID de problema `p-1234567890`.

```
aws application-insights describe-problem –problem-id p-1234567890
```

Descripción de un problema de la aplicación mediante Tools for Windows PowerShell de AWS

Utilice el siguiente comando para usar AWS Tools for Windows PowerShell a fin de describir un problema con el ID de problema `p-1234567890`.

```
Get-CWAIProblem -ProblemId p-1234567890
```

Descripción de las anomalías o los errores asociados a un problema

Describa las anomalías o los errores asociados a un problema mediante AWS CLI

Utilice el siguiente comando para usar AWS CLI para describir las anomalías o los errores asociados a un problema con el ID de problema `p-1234567890`.

```
aws application-insights describe-problem-observations --problem-id -1234567890
```

Describa las anomalías o los errores asociados a un problema mediante AWS Tools for Windows PowerShell

Utilice el siguiente comando para usar AWS Tools for Windows PowerShell para describir las anomalías o los errores asociados a un problema con el ID de problema `p-1234567890`.

```
Get-CWAIProblemObservation -ProblemId p-1234567890
```

Descripción de una anomalía o un error con la aplicación

Descripción de una anomalía o un error con la aplicación mediante la CLI de AWS

Utilice el siguiente comando para usar AWS CLI a fin de describir una anomalía o un error con la aplicación con el ID de observación `o-1234567890`.

```
aws application-insights describe-observation –observation-id o-1234567890
```

Descripción de una anomalía o un error con la aplicación mediante Tools for Windows PowerShell de AWS

Utilice el siguiente comando para usar AWS Tools for Windows PowerShell a fin de describir una anomalía o un error con la aplicación con el ID de observación `o-1234567890`.

```
Get-CWAIObservation -ObservationId o-1234567890
```

Descripción de las configuraciones de monitorización de un componente

Describa las configuraciones de monitoreo de un componente mediante AWS CLI

Utilice el siguiente comando para usar AWS CLI a fin de describir la configuración de monitoreo de un componente denominado `my-component` en una aplicación creada en el grupo de recursos `my-resource-group`.

```
aws application-insights describe-component-configuration –resource-group-name my-resource-group –component-name my-component
```

Descripción de las configuraciones de monitoreo de un componente mediante Tools for Windows PowerShell de AWS

Utilice el siguiente comando para usar AWS Tools for Windows PowerShell a fin de describir la configuración de monitoreo de un componente denominado `my-component`, en una aplicación creada en el grupo de recursos `my-resource-group`.

```
Get-CWAIComponentConfiguration -ComponentName my-component -ResourceGroupName my-resource-group
```

Para obtener más información sobre la configuración de componentes y, por ejemplo, archivos JSON, consulte Uso de configuraciones de componentes.

Descripción de la configuración de monitorización recomendada de un componente

Describa la configuración de monitoreo recomendada de un componente mediante AWS CLI

Cuando el componente forma parte de una aplicación de .NET Worker, puede utilizar AWS CLI para describir la configuración de monitoreo recomendada de un componente denominado `my-component` en una aplicación creada en el grupo de recursos `my-resource-group` mediante el siguiente comando.

```
aws application-insights describe-component-configuration-recommendation --resource-group-name my-resource-group --component-name my-component --tier DOT_NET_WORKER
```

Describa la configuración de monitoreo recomendada de un componente mediante AWS Tools for Windows PowerShell

Cuando el componente forma parte de una aplicación de .NET Worker, puede utilizar AWS Tools for Windows PowerShell para describir la configuración de monitoreo recomendada de un componente denominado `my-component` en una aplicación creada en el grupo de recursos `my-resource-group` mediante el siguiente comando.

```
Get-CWAIComponentConfigurationRecommendation -ComponentName my-component -ResourceGroupName my-resource-group -Tier DOT_NET_WORKER
```

Para obtener más información sobre la configuración de componentes y, por ejemplo, archivos JSON, consulte Uso de configuraciones de componentes.

Actualización de las configuraciones de monitorización de un componente

Actualice las configuraciones de monitoreo de un componente mediante AWS CLI

Utilice el siguiente comando para usar AWS CLI a fin de actualizar el componente denominado `my-component` en una aplicación creada en el grupo de recursos llamado `my-resource-group`. El comando incluye estas acciones:

1. Habilite la monitorización del componente.
2. Establezca la capa del componente en `.NET Worker`.
3. Actualice la configuración JSON del componente para la lectura desde el archivo local `configuration.txt`.

```
aws application-insights update-component-configuration --resource-group-name my-
resource-group --component-name my-component --tier DOT_NET_WORKER --monitor --
component-configuration "file://configuration.txt"
```

Actualice las configuraciones de monitoreo de un componente mediante AWS Tools for Windows PowerShell

Utilice el siguiente comando para usar AWS Tools for Windows PowerShell a fin de actualizar el componente denominado `my-component` en una aplicación creada en el grupo de recursos llamado `my-resource-group`. El comando incluye estas acciones:

1. Habilite la monitorización del componente.
2. Establezca la capa del componente en `.NET Worker`.
3. Actualice la configuración JSON del componente para la lectura desde el archivo local `configuration.txt`.

```
[string]$config = Get-Content -Path configuration.txt
Update-CWAIComponentConfiguration -ComponentName my-component -ResourceGroupName my-
resource-group -Tier DOT_NET_WORKER -Monitor 1 -ComponentConfiguration $config
```

Para obtener más información sobre la configuración de componentes y, por ejemplo, archivos JSON, consulte Uso de configuraciones de componentes.

Eliminación de un grupo de recursos especificado del monitoreo de Información de aplicaciones

Eliminación de un grupo de recursos específico del monitoreo de Información de aplicaciones mediante la AWS CLI

Utilice el siguiente comando para usar AWS CLI a fin de eliminar del monitoreo una aplicación creada en el grupo de recursos denominado `my-resource-group`.

```
aws application-insights delete-application --resource-group-name my-resource-group
```

Eliminación de un grupo de recursos específico del monitoreo de Información de aplicaciones mediante la AWS Tools for Windows PowerShell

Utilice el siguiente comando para usar AWS Tools for Windows PowerShell a fin de eliminar del monitoreo una aplicación creada en el grupo de recursos denominado `my-resource-group`.

```
Remove-CWAIApplication -ResourceGroupName my-resource-group
```

Configurar la supervisión de los grupos de disponibilidad siempre activos de SQL

1. Crear una aplicación para el grupo de recursos con las instancias EC2 de HA de SQL.

```
aws application-insights create-application --region <REGION> --resource-group-name
  <RESOURCE_GROUP_NAME>
```

2. Defina las instancias EC2 que representan el clúster de alta disponibilidad de SQL creando un nuevo componente de aplicación.

```
aws application-insights create-component --resource-group-name
 "<RESOURCE_GROUP_NAME>" --component-name SQL_HA_CLUSTER --resource-list
  "arn:aws:ec2:<REGION>:<ACCOUNT_ID>:instance/<CLUSTER_INSTANCE_1_ID>"
 "arn:aws:ec2:<REGION>:<ACCOUNT_ID>:instance/<CLUSTER_INSTANCE_2_ID>
```

3. Configure el componente de alta disponibilidad de SQL.

```
aws application-insights  update-component-configuration --resource-group-name
 "<RESOURCE_GROUP_NAME>" --region <REGION> --component-name "SQL_HA_CLUSTER" --
monitor --tier SQL_SERVER_ALWAYSON_AVAILABILITY_GROUP --monitor  --component-
configuration '{
  "subComponents" : [ {
    "subComponentType" : "AWS::EC2::Instance",
    "alarmMetrics" : [ {
      "alarmMetricName" : "CPUUtilization",
      "monitor" : true
    }, {
      "alarmMetricName" : "StatusCheckFailed",
      "monitor" : true
    }, {
      "alarmMetricName" : "Processor % Processor Time",
      "monitor" : true
    }, {
      "alarmMetricName" : "Memory % Committed Bytes In Use",
      "monitor" : true
    }, {
      "alarmMetricName" : "Memory Available Mbytes",
      "monitor" : true
    }, {
      "alarmMetricName" : "Paging File % Usage",
```

```
      "monitor" : true
    }, {
      "alarmMetricName" : "System Processor Queue Length",
      "monitor" : true
    }, {
      "alarmMetricName" : "Network Interface Bytes Total/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "PhysicalDisk % Disk Time",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Buffer Manager Buffer cache hit ratio",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Buffer Manager Page life expectancy",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:General Statistics Processes blocked",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:General Statistics User Connections",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Locks Number of Deadlocks/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:SQL Statistics Batch Requests/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Database Replica File Bytes Received/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Database Replica Log Bytes Received/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Database Replica Log remaining for undo",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Database Replica Log Send Queue",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Database Replica Mirrored Write Transaction/
sec",
      "monitor" : true
```

```
    }, {
      "alarmMetricName" : "SQLServer:Database Replica Recovery Queue",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Database Replica Redo Bytes Remaining",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Database Replica Redone Bytes/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Database Replica Total Log requiring undo",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Database Replica Transaction Delay",
      "monitor" : true
    } ],
    "windowsEvents" : [ {
      "logGroupName" : "WINDOWS_EVENTS-Application-<RESOURCE_GROUP_NAME>",
      "eventName" : "Application",
      "eventLevels" : [ "WARNING", "ERROR", "CRITICAL", "INFORMATION" ],
      "monitor" : true
    }, {
      "logGroupName" : "WINDOWS_EVENTS-System-<RESOURCE_GROUP_NAME>",
      "eventName" : "System",
      "eventLevels" : [ "WARNING", "ERROR", "CRITICAL" ],
      "monitor" : true
    }, {
      "logGroupName" : "WINDOWS_EVENTS-Security-<RESOURCE_GROUP_NAME>",
      "eventName" : "Security",
      "eventLevels" : [ "WARNING", "ERROR", "CRITICAL" ],
      "monitor" : true
    } ],
    "logs" : [ {
      "logGroupName" : "SQL_SERVER_ALWAYSON_AVAILABILITY_GROUP-
<RESOURCE_GROUP_NAME>",
      "logPath" : "C:\\Program Files\\Microsoft SQL Server\\MSSQL**.MSSQLSERVER\
\MSSQL\\Log\\ERRORLOG",
      "logType" : "SQL_SERVER",
      "monitor" : true,
      "encoding" : "utf-8"
    } ]
  }, {
    "subComponentType" : "AWS::EC2::Volume",
    "alarmMetrics" : [ {
```

```
      "alarmMetricName" : "VolumeReadBytes",
      "monitor" : true
    }, {
    "alarmMetricName" : "VolumeWriteBytes",
      "monitor" : true
    }, {
    "alarmMetricName" : "VolumeReadOps",
      "monitor" : true
    }, {
    "alarmMetricName" : "VolumeWriteOps",
      "monitor" : true
    }, {
    "alarmMetricName" : "VolumeQueueLength",
      "monitor" : true
    }, {
    "alarmMetricName" : "VolumeThroughputPercentage",
      "monitor" : true
    }, {
    "alarmMetricName" : "BurstBalance",
      "monitor" : true
    } ]
  } ]
}'
```

> **Note**
>
> Información de aplicaciones debe adquirir registros de eventos de aplicación (nivel de información) para detectar actividades de clúster como la conmutación por error.

Configurar la monitoreo para RDS de MySQL

1. Crear una aplicación para el grupo de recursos con la instancia de base de datos MySQL en RDS.

```
aws application-insights create-application --region <REGION> --resource-group-name
  <RESOURCE_GROUP_NAME>
```

2. El registro de errores están habilitados de forma predeterminada. El registro de consultas lentas se puede habilitar mediante grupos de parámetros de datos. Para obtener más información, consulte Acceso al registro de consultas lentas de MySQL y al registro general.

- `set slow_query_log = 1`
- `set log_output = FILE`

3. Exportar los registros que se van a supervisar a los CloudWatch Logs. Para obtener más información, consulte Publicación de registros de MySQL en Amazon CloudWatch Logs.
4. Configurar el componente RDS de MySQL.

```
aws application-insights  update-component-configuration --resource-group-name
 "<RESOURCE_GROUP_NAME>" --region <REGION> --component-name "<DB_COMPONENT_NAME>"
 --monitor --tier DEFAULT --monitor  --component-configuration "{\"alarmMetrics\":
[{\"alarmMetricName\":\"CPUUtilization\",\"monitor\":true}],\"logs\":[{\"logType\":
\"MYSQL\",\"monitor\":true},{\"logType\": \"MYSQL_SLOW_QUERY\",\"monitor\":false}]}"
```

Configurar la supervisión para EC2 de MySQL

1. Crear una aplicación para el grupo de recursos con las instancias EC2 de HA de SQL.

```
aws application-insights create-application --region <REGION> --resource-group-name
  <RESOURCE_GROUP_NAME>
```

2. El registro de errores están habilitados de forma predeterminada. El registro de consultas lentas se puede habilitar mediante grupos de parámetros de datos. Para obtener más información, consulte Acceso al registro de consultas lentas de MySQL y al registro general.
   - `set slow_query_log = 1`
   - `set log_output = FILE`
3. Configurar el componente MySQL en EC2.

```
aws application-insights  update-component-configuration --resource-group-name
 "<RESOURCE_GROUP_NAME>" --region <REGION> --component-name "<DB_COMPONENT_NAME>"
 --monitor --tier MYSQL --monitor  --component-configuration "{\"alarmMetrics\":
[{\"alarmMetricName\":\"CPUUtilization\",\"monitor\":true}],\"logs\":[{\"logGroupName
\":\"<UNIQUE_LOG_GROUP_NAME>\",\"logPath\":\"C:\\\\ProgramData\\\\MySQL\\\\MySQL
 Server **\\\\Data\\\\<FILE_NAME>.err\",\"logType\":\"MYSQL\",\"monitor\":true,
\"encoding\":\"utf-8\"}]}"
```

Configuración del monitoreo para RDS de PostgreSQL

1. Cree una aplicación para el grupo de recursos con la instancia de base de datos RDS de PostgreSQL.

```
aws application-insights create-application --region <REGION> --resource-group-name
  <RESOURCE_GROUP_NAME>
```

2. La publicación de registros de PostgreSQL en CloudWatch no está habilitada de forma predeterminada. Para habilitar la supervisión, abra la consola de RDS y seleccione la base de datos que desea monitorear. Elija Modify (Modificar) en la esquina superior derecha y seleccione el casillero de PostgreSQL. Seleccione Continue (Continuar) para guardar esta configuración.
3. Los registros de PostgreSQL se exportan a CloudWatch.
4. Configurar el componente RDS de PostgreSQL.

```
aws application-insights update-component-configuration --region <REGION> --resource-
group-name <RESOURCE_GROUP_NAME> --component-name <DB_COMPONENT_NAME> --monitor --
tier DEFAULT --component-configuration
"{
   \"alarmMetrics\":[
      {
         \"alarmMetricName\": \"CPUUtilization\",
         \"monitor\": true
      }
   ],
   \"logs\":[
      {
         \"logType\": \"POSTGRESQL\",
         \"monitor\": true
      }
   ]
}"
```

Configurar el monitoreo para EC2 de PostgreSQL

1. Crear una aplicación para el grupo de recursos con las instancias EC2 de PostgreSQL.

```
aws application-insights create-application --region <REGION> --resource-group-name
  <RESOURCE_GROUP_NAME>
```

2. Configurar el componente PostgreSQL de EC2.

```
aws application-insights update-component-configuration --region <REGION> --resource-
group-name <RESOURCE_GROUP_NAME> --component-name <DB_COMPONENT_NAME> --monitor --
tier POSTGRESQL --component-configuration
"{
   \"alarmMetrics\":[
      {
         \"alarmMetricName\":\"CPUUtilization\",
         \"monitor\":true
      }
   ],
   \"logs\":[
      {
         \"logGroupName\":\"<UNIQUE_LOG_GROUP_NAME>\",
         \"logPath\":\"/var/lib/pgsql/data/log/\",
         \"logType\":\"POSTGRESQL\",
         \"monitor\":true,
         \"encoding\":\"utf-8\"
      }
   ]
}"
```

Configuración del monitoreo para RDS de Oracle

1. Crear una aplicación para el grupo de recursos con la instancia de base de datos RDS de Oracle.

```
aws application-insights create-application --region <REGION> --resource-group-name
  <RESOURCE_GROUP_NAME>
```

2. La publicación de registros de Oracle en CloudWatch no está habilitada de forma predeterminada. Para habilitar el monitoreo, abra la consola de RDS y seleccione la base de datos que desea monitorear. Elija Modify (Modificar) en la esquina superior derecha y seleccione los casilleros de registro de Alert (Alerta) y registro del Listener (Agente de escucha). Elija Continue (Continuar) para guardar esta configuración.
3. Los registros de Oracle se exportan a CloudWatch.
4. Configuración del componente RDS de Oracle.

```
aws application-insights update-component-configuration --region <REGION> --resource-
group-name <RESOURCE_GROUP_NAME> --component-name <DB_COMPONENT_NAME> --monitor --
tier DEFAULT --component-configuration
"{
   \"alarmMetrics\":[
      {
         \"alarmMetricName\": \"CPUUtilization\",
         \"monitor\": true
      }
   ],
   \"logs\":[
      {
         \"logType\": \"ORACLE_ALERT\",
         \"monitor\": true
      },
      {
         \"logType\": \"ORACLE_LISTENER\",
         \"monitor\": true
      }
   ]
}"
```

Configuración del monitoreo para EC2 de Oracle.

1. Creación de una aplicación para el grupo de recursos con las instancias EC2 de Oracle.

```
aws application-insights create-application --region <REGION> --resource-group-name
  <RESOURCE_GROUP_NAME>
```

2. Configuración del componente EC2 de Oracle.

```
aws application-insights update-component-configuration --region <REGION> --resource-
group-name <RESOURCE_GROUP_NAME> --component-name <DB_COMPONENT_NAME> --monitor --
tier ORACLE --component-configuration
"{
   \"alarmMetrics\":[
      {
         \"alarmMetricName\":\"CPUUtilization\",
         \"monitor\":true
      }
   ],
```

```
   \"logs\":[
      {
         \"logGroupName\":\"<UNIQUE_LOG_GROUP_NAME>\",
         \"logPath\":\"/opt/oracle/diag/rdbms/*/*/trace\",
         \"logType\":\"ORACLE_ALERT\",
         \"monitor\":true,
      },
      {
         \"logGroupName\":\"<UNIQUE_LOG_GROUP_NAME>\",
         \"logPath\":\"/opt/oracle/diag/tnslsnr/$HOSTNAME/listener/trace/\",
         \"logType\":\"ORACLE_ALERT\",
         \"monitor\":true,
      }
   ]
}"
```

Información de aplicaciones CloudWatch Events y notificaciones para problemas detectados

Para cada aplicación que se agregue a Información de aplicaciones de CloudWatch, se publica un evento de CloudWatch para los siguientes eventos en base al mejor esfuerzo:

- Creación del problema. Se emite cuando Información de aplicaciones de CloudWatch detecta un problema nuevo.
  - Tipo de detalle: “Problema detectado por Información de aplicaciones”
  - Detalles:
    - `problemId`: el ID del problema detectado.
    - `region`: la Región de AWS en la que se creó el problema.
    - `resourceGroupName`: el grupo de recursos de la aplicación registrada para la que se detectó el problema.
    - `status`: el estado del problema. Los estados y las definiciones posibles son los siguientes:
      - `In progress`: se ha identificado un nuevo problema. El problema sigue siendo la recepción de observaciones.
      - `Recovering`: el problema se está estabilizando. Puede resolver el problema de forma manual cuando se encuentre en este estado.
      - `Resolved`: el problema está resuelto. No hay observaciones nuevas sobre este problema.
      - `Recurring`: el problema se resolvió en las últimas 24 horas. Se ha vuelto a abrir como resultado de observaciones adicionales.

    - `severity`: la gravedad del problema.
    - `problemUrl`: la URL de la consola del problema.

- Actualización del problema. Se emite cuando el problema se actualiza con una nueva observación o cuando una observación existente se actualiza y el problema se actualiza en consecuencia; las actualizaciones incluyen la resolución o cierre del problema.
  - Tipo de detalle: “Problema detectado por Información de aplicaciones”
  - Detalles:
    - `problemId`: el ID de problema creado.
    - `region`: la Región de AWS en la que se creó el problema.
    - `resourceGroupName`: el grupo de recursos de la aplicación registrada para la que se detectó el problema.
    - `status`: el estado del problema.
    - `severity`: la gravedad del problema.
    - `problemUrl`: la URL de la consola del problema.

Cómo recibir notificaciones para los eventos de problemas generados por una aplicación

En la consola de CloudWatch, seleccione Rules (Reglas) en Events (Eventos), en el panel de navegación izquierdo. En la página Rules (Reglas), seleccione Create rule (Crear regla). Elija Información de aplicaciones de Amazon CloudWatch de la lista desplegable Nombre del servicio y seleccione Tipo de evento. A continuación, elija Add target (Agregar destino) y seleccione el destino y los parámetros; por ejemplo, un tema de SNS o una función de Lambda.

Acciones a través de AWS Systems Manager. Información de aplicaciones de CloudWatch proporciona integración incorporada con Systems Manager OpsCenter. Si decide utilizar esta integración para su aplicación, se crea un OpsItem en la consola de OpsCenter para cada problema detectado en la aplicación. Desde la consola de OpsCenter, puede ver información resumida sobre el problema detectado por Información de aplicaciones de CloudWatch y elegir un manual de procedimientos de Systems Manager Automation para tomar medidas correctivas o identificar mejor los procesos de Windows que estén causando problemas de recursos en su aplicación.

Información de aplicaciones: observabilidad entre cuentas

Gracias a la observabilidad entre cuentas de Información de aplicaciones de CloudWatch, puede supervisar y solucionar problemas en las aplicaciones que abarcan varias cuentas de AWS de una región.

Puede usar el Administrador de acceso a la observabilidad de Amazon CloudWatch para configurar una o más de sus cuentas AWS como una cuenta de monitorización. Para proporcionar a la cuenta de monitoreo la capacidad de ver los datos de su cuenta de origen, cree un depósito en su cuenta de monitoreo. Use el receptor para crear un enlace desde su cuenta de origen a su cuenta de monitoreo. Para obtener más información, consulte Observabilidad entre cuentas de CloudWatch.

Recursos necesarios de

Para que la observabilidad multicuenta de Información de aplicaciones de CloudWatch funcione correctamente, asegúrese de que los siguientes tipos de telemetría se compartan a través del administrador de acceso a la observabilidad de CloudWatch.

- Aplicaciones en Información de aplicaciones de CloudWatch
- Métricas en Amazon CloudWatch
- Grupos de registro de Amazon CloudWatch Logs
- Trazas en AWS X-Ray

Uso de configuraciones de componentes

La configuración de un componente es un archivo de texto en formato JSON que describe las opciones de configuración del componente. En esta sección se proporciona un fragmento de plantilla de ejemplo, descripciones de secciones de configuración de componentes y configuraciones de componentes de ejemplo.

Temas

- Fragmento de plantilla de configuración del componente
- Secciones de configuración del componente
- Ejemplos de configuración del componente

Fragmento de plantilla de configuración del componente

El siguiente ejemplo muestra un fragmento de código de plantilla en formato JSON.

```
{
  "alarmMetrics" : [
    list of alarm metrics
  ],
  "logs" : [
    list of logs
  ],
  "processes" : [
   list of processes
  ],
  "windowsEvents" : [
    list of windows events channels configurations
  ],
  "alarms" : [
    list of CloudWatch alarms
  ],
  "jmxPrometheusExporter": {
    JMX Prometheus Exporter configuration
  },
  "hanaPrometheusExporter": {
      SAP HANA Prometheus Exporter configuration
  },
  "haClusterPrometheusExporter": {
      HA Cluster Prometheus Exporter configuration
  },
  "netWeaverPrometheusExporter": {
      SAP NetWeaver Prometheus Exporter configuration
  },
  "subComponents" : [
    {
      "subComponentType" : "AWS::EC2::Instance" ...
      component nested instances configuration
    },
    {
      "subComponentType" : "AWS::EC2::Volume" ...
      component nested volumes configuration
    }
  ]
}
```

Secciones de configuración del componente

La configuración de un componente incluye varias secciones principales. Las secciones de la configuración de un componente pueden estar en cualquier orden.

- alarmMetrics (opcional)

  Una lista de métricas que se van a monitorizar para el componente. Todos los tipos de componente pueden tener una sección alarmMetrics.
- logs (opcional)

  Una lista de registros que se van a monitorizar para el componente. Solo las instancias EC2 pueden tener una sección de registros.
- procesos (opcional)

  Una lista de procesos a monitorizar para el componente. Solo las instancias EC2 pueden tener una sección de procesos.
- subComponents (opcional)

  Configuración de la instancia anidada y subcomponente de volumen para el componente. Los siguientes tipos de componente pueden tener instancias anidadas y una sección de subcomponentes: ELB, ASG, instancias EC2 agrupadas de manera personalizada e instancias EC2.
- alarms (opcional)

  Una lista de alarmas que se van a monitorizar para el componente. Todos los tipos de componente pueden tener una sección alarm.
- windowsEvents (opcional)

  Una lista de windows events (eventos de Windows) para monitorear el componente. Solo Windows en instancias EC2 tienen una sección `windowsEvents`.
- JMXPrometheusExporter (opcional)

  Configuración de JMXPrometheus Exporter
- HanaprometheusExporter (opcional)

  Configuración de SAP HANA Prometheus Exporter (Exportador SAP Hana para Prometheus).
- haClusterPrometheusExporter (opcional)

Configuración de HA Cluster Prometheus Exporter (Exportador de clúster de alta disponibilidad para Prometheus).

- netWeaverPrometheusExporter (opcional)

  Configuración del exportador de Prometheus de SAP NetWeaver

- sapAsePrometheusExporter (opcional)

  Configuración de SAP ASE Prometheus Exporter.

En el siguiente ejemplo se muestra la sintaxis para el fragmento de la sección de subcomponentes en formato JSON.

```
[
  {
    "subComponentType" : "AWS::EC2::Instance",
    "alarmMetrics" : [
      list of alarm metrics
    ],
    "logs" : [
      list of logs
    ],
    "processes": [
      list of processes
    ],
    "windowsEvents" : [
      list of windows events channels configurations
    ]
  },
  {
    "subComponentType" : "AWS::EC2::Volume",
    "alarmMetrics" : [
      list of alarm metrics
    ]
  }
]
```

Propiedades de la sección de configuración del componente

En esta sección se describen las propiedades de cada sección de configuración de componentes.

Secciones

- Métrica
- Registro
- Proceso
- JMX Prometheus Exporter
- HANA Prometheus Exporter (Exportador HANA para Prometheus)
- HA Cluster Prometheus Exporter (Exportador de clúster de alta disponibilidad para Prometheus)
- Exportador de Prometheus de NetWeaver
- SAP ASE Prometheus Exporter
- Eventos de Windows
- Alarma

Métrica

Define una métrica que se va a monitorizar para el componente.

JSON

```
{
  "alarmMetricName" : "monitoredMetricName",
  "monitor" : true/false
}
```

Propiedades

- alarmMetricName (obligatorio)

  El nombre de la métrica que se va a monitorizar para el componente. Para conocer las métricas admitidas por Información de aplicaciones, consulte Registros y métricas que Información de aplicaciones de Amazon CloudWatch admite.
- monitor (opcional)

  Valor booleano para indicar si se debe monitorizar la métrica. El valor predeterminado es `true`.

Registro

Define un registro que se va a monitorizar para el componente.

JSON

```
{
  "logGroupName" : "logGroupName",
  "logPath" : "logPath",
  "logType" : "logType",
  "encoding" : "encodingType",
  "monitor" : true/false
}
```

Propiedades

- logGroupName (obligatorio)

  El nombre del grupo de registros de CloudWatch que se asociará al registro monitorizado. Para conocer las restricciones de nombre del grupo de registros, consulte CreateLogGroup.

- logPath (obligatorio para los componentes de la instancia EC2 y no para los componentes que no usan un agente de CloudWatch, como AWS Lambda)

  La ruta de los registros que se van a monitorizar. La ruta de los registros debe ser una ruta absoluta de archivos del sistema de Windows. Para obtener más información, consulte Archivo de configuración del agente de CloudWatch: sección de registros.

- logType (obligatorio)

  El tipo de registro decide los patrones de registro con respecto a los cuales Información de aplicaciones analiza el registro. El tipo de registro se selecciona entre las siguientes opciones:

  - SQL_SERVER
  - MYSQL
  - MYSQL_SLOW_QUERY
  - POSTGRESQL
  - ORACLE_ALERT
  - ORACLE_LISTENER
  - IIS
  - APPLICATION
  - WINDOWS_EVENTS
  - WINDOWS_EVENTS_ACTIVE_DIRECTORY

  - `WINDOWS_EVENTS_DNS`
  - `WINDOWS_EVENTS_IIS`
  - `WINDOWS_EVENTS_SHAREPOINT`
  - `SQL_SERVER_ALWAYSON_AVAILABILITY_GROUP`
  - `SQL_SERVER_FAILOVER_CLUSTER_INSTANCE`
  - `DEFAULT`
  - `CUSTOM`
  - `STEP_FUNCTION`
  - `API_GATEWAY_ACCESS`
  - `API_GATEWAY_EXECUTION`
  - `SAP_HANA_LOGS`
  - `SAP_HANA_TRACE`
  - `SAP_HANA_HIGH_AVAILABILITY`
  - `SAP_NETWEAVER_DEV_TRACE_LOGS`
  - `PACEMAKER_HIGH_AVAILABILITY`
- encoding (opcional)

  El tipo de codificación de los registros que se van a monitorizar. La codificación especificada debe incluirse en la lista de codificaciones compatibles con el agente de CloudWatch. Si no se proporciona, Información de aplicaciones de CloudWatch utiliza el tipo de codificación predeterminado utf-8, excepto para:

  - `SQL_SERVER`: codificación utf-16
  - `IIS`: codificación ascii
- monitor (opcional)

  Valor booleano que indica si se deben monitorizar los registros. El valor predeterminado es `true`.

Proceso

Define un proceso que se va a monitorizar para el componente.

JSON

```
{
```

```
  "processName" : "monitoredProcessName",
  "alarmMetrics" : [
      list of alarm metrics
  ]
}
```

Propiedades

- processName (obligatorio)

  El nombre del proceso a monitorizar para el componente. El nombre del proceso no debe contener una raíz de proceso, como `sqlservr` o `sqlservr.exe`.

- alarmMetrics (obligatorio)

  Una lista de métricas a supervisar para este proceso. Para ver las métricas de procesos admitidas por Información de aplicaciones de CloudWatch, consulte Amazon Elastic Compute Cloud (EC2) .

JMX Prometheus Exporter

Define la configuración de JMX Prometheus Exporter.

JSON

```
"JMXPrometheusExporter": {
  "jmxURL" : "JMX URL",
  "hostPort" : "The host and port",
  "prometheusPort" : "Target port to emit Prometheus metrics"
}
```

Propiedades

- jmxURL (opcional)

  Una URL completa de JMX a la que conectarse.

- hostPort (opcional)

  El host y el puerto al que se conectarán a través de JMX remoto. Solo puede especificarse uno de `jmxURL` y `hostPort`.

- prometheusPort (opcional)

El puerto de destino al que enviarán las métricas de Prometheus. Si no se especifica, se utiliza el puerto predeterminado 9404.

HANA Prometheus Exporter (Exportador HANA para Prometheus)

Define la configuración de HANA Prometheus Exporter.

JSON

```
"hanaPrometheusExporter": {
    "hanaSid": "SAP HANA  SID",
    "hanaPort": "HANA database port",
    "hanaSecretName": "HANA secret name",
    "prometheusPort": "Target port to emit Prometheus metrics"
}
```

Propiedades

- hanaSid

  ID de sistema SAP (SID) de tres caracteres del sistema SAP HANA.
- hanaPort

  El puerto de base de datos HANA mediante el cual el exportador consultará las métricas de HANA.
- hanaSecretName

  El AWS Secrets Manager secreto que almacena las credenciales de usuario de monitoreo de HANA. En el exportador HANA de Prometheus se utilizan estas credenciales para conectarse a la base de datos y consultar métricas de HANA.
- prometheusPort (opcional)

  El puerto de destino al que se enviarán las métricas de Prometheus. Si no se especifica, se utiliza el puerto predeterminado 9668.

HA Cluster Prometheus Exporter (Exportador de clúster de alta disponibilidad para Prometheus)

Define la configuración de HA Cluster Prometheus Exporter (Exportador de clúster de alta disponibilidad para Prometheus).

JSON

```
"haClusterPrometheusExporter": {
    "prometheusPort": "Target port to emit Prometheus metrics"
}
```

Propiedades

- prometheusPort (opcional)

  El puerto de destino al que se enviarán las métricas de Prometheus. Si no se especifica, se utiliza el puerto predeterminado 9664.

Exportador de Prometheus de NetWeaver

Permite definir la configuración del exportador de Prometheus de NetWeaver.

JSON

```
"netWeaverPrometheusExporter": {
    "sapSid": "SAP NetWeaver  SID",
    "instanceNumbers": [ "Array of instance Numbers of SAP NetWeaver system "],
"prometheusPort": "Target port to emit Prometheus metrics"
}
```

Propiedades

- sapSid

  ID de sistema SAP (SID) de tres caracteres del sistema SAP NetWeaver.

- instanceNumbers

  Matriz de números de instancia del sistema SAP NetWeaver.

  Ejemplo: `"instanceNumbers": [ "00", "01"]`

- prometheusPort (opcional)

  El puerto de destino al que se enviarán las métricas de Prometheus. Si no se especifica, se utiliza el puerto predeterminado 9680.

SAP ASE Prometheus Exporter

Define la configuración de SAP ASE Prometheus Exporter.

JSON

```
"sapASEPrometheusExporter": {
    "sapAseSid": "SAP ASE SID",
    "sapAsePort": "SAP ASE database port",
    "sapAseSecretName": "SAP ASE secret name",
    "prometheusPort": "Target port to emit Prometheus metrics",
    "agreeToEnableASEMonitoring": true
}
```

Propiedades

- sapAseSid

  Identificador del sistema SAP (SID) de tres caracteres del sistema SAP ASE.
- sapAsePort

  El puerto de base de datos de SAP ASE mediante el cual el exportador consultará las métricas de ASE.
- sapAseSecretName

  El AWS Secrets Manager secreto que almacena las credenciales de usuario de supervisión de ASE. En SAP ASE Prometheus Exporter se utilizan estas credenciales para conectarse a la base de datos y consultar métricas de ASE.
- prometheusPort (opcional)

  El puerto de destino al que se enviarán las métricas de Prometheus. Si no se especifica, se utiliza el puerto predeterminado 9399. Si hay otra base de datos ASE que utilice el puerto predeterminado, entonces usaremos el 9499.

Eventos de Windows

Define los eventos de Windows que se van a registrar.

JSON

```
{
```

```
  "logGroupName" : "logGroupName",
  "eventName" : "eventName",
  "eventLevels" : ["ERROR","WARNING","CRITICAL","INFORMATION","VERBOSE"],
  "monitor" : true/false
}
```

Propiedades

- logGroupName (obligatorio)

  El nombre del grupo de registros de CloudWatch que se asociará al registro monitorizado. Para conocer las restricciones de nombre del grupo de registros, consulte CreateLogGroup.
- eventName (obligatorio)

  El tipo de eventos de Windows que se van a registrar. Equivale al nombre del canal del registro de eventos de Windows. Por ejemplo, System, Security, CustomEventName, etc. Este campo es obligatorio para cada tipo de evento de Windows que se registrará.
- eventLevels (obligatorio)

  Los niveles de evento que registrar. Debe especificar cada nivel que se registrará. Entre los valores posibles se incluyen `INFORMATION`, `WARNING`, `ERROR`, `CRITICAL` y `VERBOSE`. Este campo es obligatorio para cada tipo de evento de Windows que se registrará.
- monitor (opcional)

  Valor booleano que indica si se deben monitorizar los registros. El valor predeterminado es `true`.

Alarma

Define una alarma de CloudWatch que se va a monitorizar para el componente.

JSON

```
{
  "alarmName" : "monitoredAlarmName",
  "severity" : HIGH/MEDIUM/LOW
}
```

Propiedades

- alarmName (obligatorio)

El nombre de la alarma de CloudWatch que se va a monitorizar para el componente.

- severity (opcional)

  Indica el grado de interrupción al sonar la alarma.

Ejemplos de configuración del componente

Los siguientes ejemplos muestran configuraciones de componentes en formato JSON para los servicios relevantes.

Configuraciones de componentes de ejemplo

- Tabla de Amazon DynamoDB
- Amazon EC2 Auto Scaling (ASG) Amazon EC2 Auto Scaling
- Clúster de Amazon EKS
- Instancia de Amazon Elastic Compute Cloud (EC2)
- Amazon Elastic Container Service (Amazon ECS)
- Servicios de Amazon ECS
- Tareas de Amazon ECS
- Amazon Elastic File System (Amazon EFS)
- Amazon FSx
- Aurora MySQL de Amazon Relational Database Service (RDS)
- Instancia de Amazon Relational Database Service (RDS)
- Comprobación de estado de Amazon Route 53
- Zona alojada de Amazon Route 53
- Punto de conexión de Amazon Route 53 Resolver
- Configuración de registro de consultas Amazon Route 53 Resolver
- Bucket de Amazon S3
- Amazon Simple Queue Service (SQS)
- Tema de Amazon SNS
- Amazon Virtual Private Cloud (Amazon VPC)
- Pasarelas de traducción de direcciones de red (NAT) de Amazon VPC
- Etapas de la API REST de API Gateway

- Elastic Load Balancing de aplicaciones
- AWS Lambda Función
- Grupo de reglas de AWS Network Firewall
- Asociación de grupos de reglas AWS Network Firewall
- AWS Step Functions
- Instancias de Amazon EC2 agrupadas por cliente
- Elastic Load Balancing
- Java
- Kubernetes en Amazon EC2
- RDS MariaDB y RDS MySQL
- Oracle de RDS
- PostgreSQL de RDS
- SAP ASE en Amazon EC2
- Alta disponibilidad de SAP ASE en Amazon EC2
- SAP HANA en Amazon EC2
- Alta disponibilidad de SAP HANA en Amazon EC2
- SAP NetWeaver en Amazon EC2
- Alta disponibilidad de SAP NetWeaver en Amazon EC2
- Grupos de disponibilidad siempre activos de SQL
- Instancia de clúster de conmutación por error de SQL

Tabla de Amazon DynamoDB

En el siguiente ejemplo se muestra una configuración de componentes en formato JSON para la tabla de Amazon DynamoDB.

```
{
  "alarmMetrics": [
    {
      "alarmMetricName": "SystemErrors",
      "monitor": false
    },
    {
      "alarmMetricName": "UserErrors",
```

```
      "monitor": false
    },
    {
      "alarmMetricName": "ConsumedReadCapacityUnits",
      "monitor": false
    },
    {
      "alarmMetricName": "ConsumedWriteCapacityUnits",
      "monitor": false
    },
    {
      "alarmMetricName": "ReadThrottleEvents",
      "monitor": false
    },
    {
      "alarmMetricName": "WriteThrottleEvents",
      "monitor": false
    },
    {
      "alarmMetricName": "ConditionalCheckFailedRequests",
      "monitor": false
    },
    {
      "alarmMetricName": "TransactionConflict",
      "monitor": false
    }
  ],
  "logs": []
}
```

Amazon EC2 Auto Scaling (ASG) Amazon EC2 Auto Scaling

En el siguiente ejemplo se muestra una configuración de componentes en formato JSON para Amazon EC2 Auto Scaling (ASG).

```
{
    "alarmMetrics" : [
      {
        "alarmMetricName" : "CPUCreditBalance"
      }, {
        "alarmMetricName" : "EBSIOBalance%"
      }
    ],
```

```
    "subComponents" : [
      {
        "subComponentType" : "AWS::EC2::Instance",
        "alarmMetrics" : [
          {
            "alarmMetricName" : "CPUUtilization"
          }, {
            "alarmMetricName" : "StatusCheckFailed"
          }
        ],
        "logs" : [
          {
            "logGroupName" : "my_log_group",
            "logPath" : "C:\\LogFolder\\*",
            "logType" : "APPLICATION"
          }
        ],
        "processes" : [
          {
            "processName" : "my_process",
            "alarmMetrics" : [
              {
                  "alarmMetricName" : "procstat cpu_usage",
                  "monitor" : true
              }, {
                  "alarmMetricName" : "procstat memory_rss",
                  "monitor" : true
              }
          ]
      }
  ],
        "windowsEvents" : [
          {
            "logGroupName" : "my_log_group_2",
            "eventName" : "Application",
            "eventLevels" : [ "ERROR", "WARNING", "CRITICAL" ]
          }
        ]
      }, {
        "subComponentType" : "AWS::EC2::Volume",
        "alarmMetrics" : [
          {
            "alarmMetricName" : "VolumeQueueLength"
          }, {
```

```
            "alarmMetricName" : "BurstBalance"
          }
        ]
      }
    ],
    "alarms" : [
      {
        "alarmName" : "my_asg_alarm",
        "severity" : "LOW"
      }
    ]
  }
```

Clúster de Amazon EKS

En el siguiente ejemplo se muestra una configuración de componentes en formato JSON para el clúster de Amazon EKS.

```
{
    "alarmMetrics":[
       {
          "alarmMetricName": "cluster_failed_node_count",
          "monitor":true
       },
       {
          "alarmMetricName": "node_cpu_reserved_capacity",
          "monitor":true
       },
       {
          "alarmMetricName": "node_cpu_utilization",
          "monitor":true
       },
       {
          "alarmMetricName": "node_filesystem_utilization",
          "monitor":true
       },
       {
          "alarmMetricName": "node_memory_reserved_capacity",
          "monitor":true
       },
       {
          "alarmMetricName": "node_memory_utilization",
          "monitor":true
```

```
        },
        {
           "alarmMetricName": "node_network_total_bytes",
           "monitor":true
        },
        {
           "alarmMetricName": "pod_cpu_reserved_capacity",
           "monitor":true
        },
        {
           "alarmMetricName": "pod_cpu_utilization",
           "monitor":true
        },
        {
           "alarmMetricName": "pod_cpu_utilization_over_pod_limit",
           "monitor":true
        },
        {
           "alarmMetricName": "pod_memory_reserved_capacity",
           "monitor":true
        },
        {
           "alarmMetricName": "pod_memory_utilization",
           "monitor":true
        },
        {
           "alarmMetricName": "pod_memory_utilization_over_pod_limit",
           "monitor":true
        },
        {
           "alarmMetricName": "pod_network_rx_bytes",
           "monitor":true
        },
        {
           "alarmMetricName": "pod_network_tx_bytes",
           "monitor":true
        }
     ],
     "logs":[
        {
           "logGroupName": "/aws/containerinsights/kubernetes/application",
           "logType":"APPLICATION",
           "monitor":true,
           "encoding":"utf-8"
```

```
      }
   ],
   "subComponents":[
      {
         "subComponentType":"AWS::EC2::Instance",
         "alarmMetrics":[
            {
               "alarmMetricName":"CPUUtilization",
               "monitor":true
            },
            {
               "alarmMetricName":"StatusCheckFailed",
               "monitor":true
            },
            {
               "alarmMetricName":"disk_used_percent",
               "monitor":true
            },
            {
               "alarmMetricName":"mem_used_percent",
               "monitor":true
            }
         ],
         "logs":[
            {
               "logGroupName":"APPLICATION-KubernetesClusterOnEC2-IAD",
               "logPath":"",
               "logType":"APPLICATION",
               "monitor":true,
               "encoding":"utf-8"
            }
         ],
         "processes" : [
           {
               "processName" : "my_process",
               "alarmMetrics" : [
                   {
                       "alarmMetricName" : "procstat cpu_usage",
                       "monitor" : true
                   }, {
                       "alarmMetricName" : "procstat memory_rss",
                       "monitor" : true
                   }
               ]
```

```
            }
         ],
           "windowsEvents":[
              {
                 "logGroupName":"my_log_group_2",
                 "eventName":"Application",
                 "eventLevels":[
                    "ERROR",
                    "WARNING",
                    "CRITICAL"
                 ],
                 "monitor":true
              }
           ]
        },
        {
           "subComponentType":"AWS::AutoScaling::AutoScalingGroup",
           "alarmMetrics":[
              {
                 "alarmMetricName":"CPUCreditBalance",
                 "monitor":true
              },
              {
                 "alarmMetricName":"EBSIOBalance%",
                 "monitor":true
              }
           ]
        },
        {
           "subComponentType":"AWS::EC2::Volume",
           "alarmMetrics":[
              {
                 "alarmMetricName":"VolumeReadBytes",
                 "monitor":true
              },
              {
                 "alarmMetricName":"VolumeWriteBytes",
                 "monitor":true
              },
              {
                 "alarmMetricName":"VolumeReadOps",
                 "monitor":true
              },
              {
```

```
                "alarmMetricName":"VolumeWriteOps",
                "monitor":true
            },
            {
                "alarmMetricName":"VolumeQueueLength",
                "monitor":true
            },
            {
                "alarmMetricName":"BurstBalance",
                "monitor":true
            }
        ]
    }
  ]
}
```

> **Note**
>
> - La sección `subComponents` de `AWS::EC2::Instance`, `AWS::EC2::Volume` y `AWS::AutoScaling::AutoScalingGroup` solo aplica al clúster de Amazon EKS que se ejecute en el tipo de lanzamiento de EC2.
> - La sección `windowsEvents` de `AWS::EC2::Instance` en `subComponents` solo aplica a Windows que se ejecute en instancias de Amazon EC2.

Instancia de Amazon Elastic Compute Cloud (EC2)

En el ejemplo siguiente se muestra una configuración de componentes en formato JSON para una instancia de Amazon EC2.

> **Important**
>
> Cuando una instancia de Amazon EC2 ingresa al estado `stopped`, se elimina de la supervisión. Cuando vuelve a un estado `running`, se agrega a la lista de Unmonitored components (Componentes sin monitorear) en la página Application details (Detalles de la aplicación) de la consola de Información de aplicaciones de CloudWatch. Si la supervisión automática de nuevos recursos está habilitada para la aplicación, la instancia se agrega a la lista de Componentes monitoreados. Sin embargo, los registros y las métricas se establecen

en el valor predeterminado de la carga de trabajo. La configuración de registro y métricas anterior no se guarda.

```
{
    "alarmMetrics" : [
      {
        "alarmMetricName" : "CPUUtilization",
        "monitor" : true
      }, {
        "alarmMetricName" : "StatusCheckFailed"
      }
    ],
    "logs" : [
      {
        "logGroupName" : "my_log_group",
        "logPath" : "C:\\LogFolder\\*",
        "logType" : "APPLICATION",
        "monitor" : true
      },
      {
        "logGroupName" : "my_log_group_2",
        "logPath" : "C:\\LogFolder2\\*",
        "logType" : "IIS",
        "encoding" : "utf-8"
      }
    ],
    "processes" : [
        {
            "processName" : "my_process",
            "alarmMetrics" : [
                {
                    "alarmMetricName" : "procstat cpu_usage",
                    "monitor" : true
                }, {
                    "alarmMetricName" : "procstat memory_rss",
                    "monitor" : true
                }
            ]
        }
    ],
    "windowsEvents" : [
      {
```

```
        "logGroupName" : "my_log_group_3",
        "eventName" : "Application",
        "eventLevels" : [ "ERROR", "WARNING", "CRITICAL" ],
        "monitor" : true
      }, {
        "logGroupName" : "my_log_group_4",
        "eventName" : "System",
        "eventLevels" : [ "ERROR", "WARNING", "CRITICAL" ],
        "monitor" : true
    }],
     "alarms" : [
      {
        "alarmName" : "my_instance_alarm_1",
        "severity" : "HIGH"
      },
      {
        "alarmName" : "my_instance_alarm_2",
        "severity" : "LOW"
      }
    ],
     "subComponents" : [
     {
       "subComponentType" : "AWS::EC2::Volume",
       "alarmMetrics" : [
       {
         "alarmMetricName" : "VolumeQueueLength",
         "monitor" : "true"
       },
       {
         "alarmMetricName" : "VolumeThroughputPercentage",
         "monitor" : "true"
       },
       {
         "alarmMetricName" : "BurstBalance",
         "monitor" : "true"
       }
    }]
  }
```

Amazon Elastic Container Service (Amazon ECS)

En el siguiente ejemplo se muestra una configuración de componentes en formato JSON para Amazon Elastic Container Service (Amazon ECS).

```
{
    "alarmMetrics":[
       {
          "alarmMetricName":"CpuUtilized",
          "monitor":true
       },
       {
          "alarmMetricName":"MemoryUtilized",
          "monitor":true
       },
       {
          "alarmMetricName":"NetworkRxBytes",
          "monitor":true
       },
       {
          "alarmMetricName":"NetworkTxBytes",
          "monitor":true
       },
       {
          "alarmMetricName":"RunningTaskCount",
          "monitor":true
       },
       {
          "alarmMetricName":"PendingTaskCount",
          "monitor":true
       },
       {
          "alarmMetricName":"StorageReadBytes",
          "monitor":true
       },
       {
          "alarmMetricName":"StorageWriteBytes",
          "monitor":true
       }
    ],
    "logs":[
       {
          "logGroupName":"/ecs/my-task-definition",
          "logType":"APPLICATION",
          "monitor":true
       }
    ],
    "subComponents":[
```

```
        {
           "subComponentType":"AWS::ElasticLoadBalancing::LoadBalancer",
           "alarmMetrics":[
              {
                 "alarmMetricName":"HTTPCode_Backend_4XX",
                 "monitor":true
              },
              {
                 "alarmMetricName":"HTTPCode_Backend_5XX",
                 "monitor":true
              },
              {
                 "alarmMetricName":"Latency",
                 "monitor":true
              },
              {
                 "alarmMetricName":"SurgeQueueLength",
                 "monitor":true
              },
              {
                 "alarmMetricName":"UnHealthyHostCount",
                 "monitor":true
              }
           ]
        },
        {
           "subComponentType":"AWS::ElasticLoadBalancingV2::LoadBalancer",
           "alarmMetrics":[
              {
                 "alarmMetricName":"HTTPCode_Target_4XX_Count",
                 "monitor":true
              },
              {
                 "alarmMetricName":"HTTPCode_Target_5XX_Count",
                 "monitor":true
              },
              {
                 "alarmMetricName":"TargetResponseTime",
                 "monitor":true
              },
              {
                 "alarmMetricName":"UnHealthyHostCount",
                 "monitor":true
              }
```

```
         ]
      },
      {
         "subComponentType":"AWS::EC2::Instance",
         "alarmMetrics":[
            {
               "alarmMetricName":"CPUUtilization",
               "monitor":true
            },
            {
               "alarmMetricName":"StatusCheckFailed",
               "monitor":true
            },
            {
               "alarmMetricName":"disk_used_percent",
               "monitor":true
            },
            {
               "alarmMetricName":"mem_used_percent",
               "monitor":true
            }
         ],
         "logs":[
            {
               "logGroupName":"my_log_group",
               "logPath":"/mylog/path",
               "logType":"APPLICATION",
               "monitor":true
            }
         ],
         "processes" : [
            {
               "processName" : "my_process",
               "alarmMetrics" : [
                   {
                       "alarmMetricName" : "procstat cpu_usage",
                       "monitor" : true
                   }, {
                       "alarmMetricName" : "procstat memory_rss",
                       "monitor" : true
                   }
                ]
            }
         ],
```

```
         "windowsEvents":[
            {
               "logGroupName":"my_log_group_2",
               "eventName":"Application",
               "eventLevels":[
                  "ERROR",
                  "WARNING",
                  "CRITICAL"
               ],
               "monitor":true
            }
         ]
      },
      {
         "subComponentType":"AWS::EC2::Volume",
         "alarmMetrics":[
            {
               "alarmMetricName":"VolumeQueueLength",
               "monitor":"true"
            },
            {
               "alarmMetricName":"VolumeThroughputPercentage",
               "monitor":"true"
            },
            {
               "alarmMetricName":"BurstBalance",
               "monitor":"true"
            }
         ]
      }
   ]
}
```

> **Note**
>
> - La sección `subComponents` de `AWS::EC2::Instance` y `AWS::EC2::Volume` sólo aplica a los clústeres de Amazon ECS con servicio ECS o tarea ECS que se ejecuten en el tipo de lanzamiento de EC2.
> - La sección `windowsEvents` de `AWS::EC2::Instance` en `subComponents` solo aplica a Windows que se ejecute en instancias de Amazon EC2.

Servicios de Amazon ECS

En el siguiente ejemplo se muestra una configuración de componentes en formato JSON para un servicio de Amazon ECS.

```
{
    "alarmMetrics":[
       {
          "alarmMetricName":"CPUUtilization",
          "monitor":true
       },
       {
          "alarmMetricName":"MemoryUtilization",
          "monitor":true
       },
       {
          "alarmMetricName":"CpuUtilized",
          "monitor":true
       },
       {
          "alarmMetricName":"MemoryUtilized",
          "monitor":true
       },
       {
          "alarmMetricName":"NetworkRxBytes",
          "monitor":true
       },
       {
          "alarmMetricName":"NetworkTxBytes",
          "monitor":true
       },
       {
          "alarmMetricName":"RunningTaskCount",
          "monitor":true
       },
       {
          "alarmMetricName":"PendingTaskCount",
          "monitor":true
       },
       {
          "alarmMetricName":"StorageReadBytes",
          "monitor":true
       },
       {
```

```
            "alarmMetricName":"StorageWriteBytes",
            "monitor":true
         }
      ],
      "logs":[
         {
            "logGroupName":"/ecs/my-task-definition",
            "logType":"APPLICATION",
            "monitor":true
         }
      ],
      "subComponents":[
         {
            "subComponentType":"AWS::ElasticLoadBalancing::LoadBalancer",
            "alarmMetrics":[
               {
                  "alarmMetricName":"HTTPCode_Backend_4XX",
                  "monitor":true
               },
               {
                  "alarmMetricName":"HTTPCode_Backend_5XX",
                  "monitor":true
               },
               {
                  "alarmMetricName":"Latency",
                  "monitor":true
               },
               {
                  "alarmMetricName":"SurgeQueueLength",
                  "monitor":true
               },
               {
                  "alarmMetricName":"UnHealthyHostCount",
                  "monitor":true
               }
            ]
         },
         {
            "subComponentType":"AWS::ElasticLoadBalancingV2::LoadBalancer",
            "alarmMetrics":[
               {
                  "alarmMetricName":"HTTPCode_Target_4XX_Count",
                  "monitor":true
               },
```

```
            {
               "alarmMetricName":"HTTPCode_Target_5XX_Count",
               "monitor":true
            },
            {
               "alarmMetricName":"TargetResponseTime",
               "monitor":true
            },
            {
               "alarmMetricName":"UnHealthyHostCount",
               "monitor":true
            }
         ]
      },
      {
         "subComponentType":"AWS::EC2::Instance",
         "alarmMetrics":[
            {
               "alarmMetricName":"CPUUtilization",
               "monitor":true
            },
            {
               "alarmMetricName":"StatusCheckFailed",
               "monitor":true
            },
            {
               "alarmMetricName":"disk_used_percent",
               "monitor":true
            },
            {
               "alarmMetricName":"mem_used_percent",
               "monitor":true
            }
         ],
         "logs":[
            {
               "logGroupName":"my_log_group",
               "logPath":"/mylog/path",
               "logType":"APPLICATION",
               "monitor":true
            }
         ],
         "processes" : [
            {
```

```
                "processName" : "my_process",
                "alarmMetrics" : [
                {
                    "alarmMetricName" : "procstat cpu_usage",
                    "monitor" : true
                }, {
                    "alarmMetricName" : "procstat memory_rss",
                    "monitor" : true
                }
            ]
        }
    ],
          "windowsEvents":[
             {
                "logGroupName":"my_log_group_2",
                "eventName":"Application",
                "eventLevels":[
                   "ERROR",
                   "WARNING",
                   "CRITICAL"
                ],
                "monitor":true
             }
          ]
       },
       {
          "subComponentType":"AWS::EC2::Volume",
          "alarmMetrics":[
             {
                "alarmMetricName":"VolumeQueueLength",
                "monitor":"true"
             },
             {
                "alarmMetricName":"VolumeThroughputPercentage",
                "monitor":"true"
             },
             {
                "alarmMetricName":"BurstBalance",
                "monitor":"true"
             }
          ]
       }
    ]
```

```
}
```

> **Note**
>
> - La sección `subComponents` de `AWS::EC2::Instance` y `AWS::EC2::Volume` solo aplica a Amazon ECS que se ejecute en el tipo de lanzamiento de EC2.
> - La sección `windowsEvents` de `AWS::EC2::Instance` en `subComponents` solo aplica a Windows que se ejecute en instancias de Amazon EC2.

Tareas de Amazon ECS

En el siguiente ejemplo se muestra una configuración de componentes en formato JSON para una tarea de Amazon ECS.

```
{
    "logs":[
       {
          "logGroupName":"/ecs/my-task-definition",
          "logType":"APPLICATION",
          "monitor":true
       }
    ],
    "processes" : [
        {
            "processName" : "my_process",
            "alarmMetrics" : [
                {
                    "alarmMetricName" : "procstat cpu_usage",
                    "monitor" : true
                }, {
                    "alarmMetricName" : "procstat memory_rss",
                    "monitor" : true
                }
            ]
        }
    ]
}
```

Amazon Elastic File System (Amazon EFS)

En el siguiente ejemplo se muestra una configuración de componentes en formato JSON para Amazon EFS.

```
{
   "alarmMetrics": [
     {
       "alarmMetricName": "BurstCreditBalance",
       "monitor": true
     },
     {
       "alarmMetricName": "PercentIOLimit",
       "monitor": true
     },
     {
       "alarmMetricName": "PermittedThroughput",
       "monitor": true
     },
     {
       "alarmMetricName": "MeteredIOBytes",
       "monitor": true
     },
     {
       "alarmMetricName": "TotalIOBytes",
       "monitor": true
     },
     {
       "alarmMetricName": "DataWriteIOBytes",
       "monitor": true
     },
     {
       "alarmMetricName": "DataReadIOBytes",
       "monitor": true
     },
     {
       "alarmMetricName": "MetadataIOBytes",
       "monitor": true
     },
     {
       "alarmMetricName": "ClientConnections",
       "monitor": true
     },
     {
```

```
        "alarmMetricName": "TimeSinceLastSync",
        "monitor": true
      },
      {
        "alarmMetricName": "Throughput",
        "monitor": true
      },
      {
        "alarmMetricName": "PercentageOfPermittedThroughputUtilization",
        "monitor": true
      },
      {
        "alarmMetricName": "ThroughputIOPS",
        "monitor": true
      },
      {
        "alarmMetricName": "PercentThroughputDataReadIOBytes",
        "monitor": true
      },
      {
        "alarmMetricName": "PercentThroughputDataWriteIOBytes",
        "monitor": true
      },
      {
        "alarmMetricName": "PercentageOfIOPSDataReadIOBytes",
        "monitor": true
      },
      {
        "alarmMetricName": "PercentageOfIOPSDataWriteIOBytes",
        "monitor": true
      },
      {
        "alarmMetricName": "AverageDataReadIOBytesSize",
        "monitor": true
      },
      {
        "alarmMetricName": "AverageDataWriteIOBytesSize",
        "monitor": true
      }
    ],
    "logs": [
     {
     "logGroupName": "/aws/efs/utils",
     "logType": "EFS_MOUNT_STATUS",
```

```
    "monitor": true,
    }
   ]
 }
```

Amazon FSx

En el siguiente ejemplo se muestra una configuración de componentes en formato JSON para Amazon FSx.

```
{
  "alarmMetrics": [
    {
      "alarmMetricName": "DataReadBytes",
      "monitor": true
    },
    {
      "alarmMetricName": "DataWriteBytes",
      "monitor": true
    },
    {
      "alarmMetricName": "DataReadOperations",
      "monitor": true
    },
    {
      "alarmMetricName": "DataWriteOperations",
      "monitor": true
    },
    {
      "alarmMetricName": "MetadataOperations",
      "monitor": true
    },
    {
      "alarmMetricName": "FreeStorageCapacity",
      "monitor": true
    }
  ]
}
```

Aurora MySQL de Amazon Relational Database Service (RDS)

En el siguiente ejemplo se muestra una configuración de componentes en formato JSON para Aurora MySQL de Amazon RDS.

```
{
  "alarmMetrics": [
    {
      "alarmMetricName": "CPUUtilization",
      "monitor": true
    },
    {
      "alarmMetricName": "CommitLatency",
      "monitor": true
    }
  ],
  "logs": [
    {
      "logType": "MYSQL",
      "monitor": true,
    },
    {
      "logType": "MYSQL_SLOW_QUERY",
      "monitor": false
    }
  ]
}
```

Instancia de Amazon Relational Database Service (RDS)

En el ejemplo siguiente se muestra una configuración de componentes en formato JSON para una instancia de Amazon RDS.

```
{
  "alarmMetrics" : [
    {
      "alarmMetricName" : "BurstBalance",
      "monitor" : true
    }, {
      "alarmMetricName" : "WriteThroughput",
      "monitor" : false
    }
  ],

  "alarms" : [
    {
      "alarmName" : "my_rds_instance_alarm",
      "severity" : "MEDIUM"
```

```
    }
  ]
}
```

Comprobación de estado de Amazon Route 53

El siguiente ejemplo muestra la configuración de un componente en formato JSON para comprobar el estado de Amazon Route 53.

```
{
  "alarmMetrics": [
    {
      "alarmMetricName": "ChildHealthCheckHealthyCount",
      "monitor": true
    },
    {
      "alarmMetricName": "ConnectionTime",
      "monitor": true
    },
    {
      "alarmMetricName": "HealthCheckPercentageHealthy",
      "monitor": true
    },
    {
      "alarmMetricName": "HealthCheckStatus",
      "monitor": true
    },
    {
      "alarmMetricName": "SSLHandshakeTime",
      "monitor": true
    },
    {
      "alarmMetricName": "TimeToFirstByte",
      "monitor": true
    }
  ]
}
```

Zona alojada de Amazon Route 53

El siguiente ejemplo muestra la configuración de un componente en formato JSON para una zona alojada en Amazon Route 53.

```
{
  "alarmMetrics": [
    {
      "alarmMetricName": "DNSQueries",
      "monitor": true
    },
    {
      "alarmMetricName": "DNSSECInternalFailure",
      "monitor": true
    },
    {
      "alarmMetricName": "DNSSECKeySigningKeysNeedingAction",
      "monitor": true
    },
    {
      "alarmMetricName": "DNSSECKeySigningKeyMaxNeedingActionAge",
      "monitor": true
    },
    {
      "alarmMetricName": "DNSSECKeySigningKeyAge",
      "monitor": true
    }
  ],
  "logs": [
    {
      "logGroupName":"/hosted-zone/logs",
      "logType": "ROUTE53_DNS_PUBLIC_QUERY_LOGS",
      "monitor": true
    }
  ]
}
```

Punto de conexión de Amazon Route 53 Resolver

En el siguiente ejemplo se muestra la configuración de un componente en formato JSON para un punto de conexión Amazon Route 53 Resolver.

```
{
  "alarmMetrics": [
    {
      "alarmMetricName": "EndpointHealthyENICount",
      "monitor": true
    },
```

```
    {
      "alarmMetricName": "EndpointUnHealthyENICount",
      "monitor": true
    },
    {
      "alarmMetricName": "InboundQueryVolume",
      "monitor": true
    },
    {
      "alarmMetricName": "OutboundQueryVolume",
      "monitor": true
    },
    {
      "alarmMetricName": "OutboundQueryAggregateVolume",
      "monitor": true
    }
  ]
}
```

Configuración de registro de consultas Amazon Route 53 Resolver

En el siguiente ejemplo se muestra la configuración de un componente en formato JSON para la configuración del registro de consultas Amazon Route 53 Resolver.

```
{
  "logs": [
    {
      "logGroupName": "/resolver-query-log-config/logs",
      "logType": "ROUTE53_RESOLVER_QUERY_LOGS",
      "monitor": true
    }
  ]
}
```

Bucket de Amazon S3

En el siguiente ejemplo se muestra una configuración de componentes en formato JSON para el bucket de Amazon S3.

```
{
    "alarmMetrics" : [
        {
```

```
            "alarmMetricName" : "ReplicationLatency",
            "monitor" : true
        }, {
            "alarmMetricName" : "5xxErrors",
            "monitor" : true
        }, {
            "alarmMetricName" : "BytesDownloaded"
            "monitor" : true
        }
    ]
}
```

Amazon Simple Queue Service (SQS)

En los siguientes ejemplos se muestra una configuración de componentes en formato JSON para Amazon Simple Queue Service.

```
{
  "alarmMetrics" : [
    {
      "alarmMetricName" : "ApproximateAgeOfOldestMessage"
    }, {
      "alarmMetricName" : "NumberOfEmptyReceives"
    }
  ],
  "alarms" : [
    {
      "alarmName" : "my_sqs_alarm",
      "severity" : "MEDIUM"
    }
  ]
}
```

Tema de Amazon SNS

En el siguiente ejemplo se muestra una configuración de componentes en formato JSON para el tema de Amazon SNS.

```
{
  "alarmMetrics": [
    {
      "alarmMetricName": "NumberOfNotificationsFailed",
```

```
      "monitor": true
    },
    {
      "alarmMetricName": "NumberOfNotificationsFilteredOut-InvalidAttributes",
      "monitor": true
    },
    {
      "alarmMetricName": "NumberOfNotificationsFilteredOut-NoMessageAttributes",
      "monitor": true
    },
    {
      "alarmMetricName": "NumberOfNotificationsFailedToRedriveToDlq",
      "monitor": true
    }
  ]
}
```

Amazon Virtual Private Cloud (Amazon VPC)

En el siguiente ejemplo se muestra una configuración de componentes en formato JSON para Amazon VPC.

```
{
  "alarmMetrics": [
    {
      "alarmMetricName": "NetworkAddressUsage",
      "monitor": true
    },
    {
      "alarmMetricName": "NetworkAddressUsagePeered",
      "monitor": true
    },
    {
      "alarmMetricName": "VPCFirewallQueryVolume",
      "monitor": true
    }
  ]
}
```

Pasarelas de traducción de direcciones de red (NAT) de Amazon VPC

En el siguiente ejemplo se muestra una configuración de componentes en formato JSON para puertas de enlace NAT.

```
{
  "alarmMetrics": [
    {
      "alarmMetricName": "ErrorPortAllocation",
      "monitor": true
    },
    {
      "alarmMetricName": "IdleTimeoutCount",
      "monitor": true
    }
  ]
}
```

Etapas de la API REST de API Gateway

En el siguiente ejemplo se muestra una configuración de componentes en formato JSON para las etapas de las API REST de API Gateway.

```
{
     "alarmMetrics" : [
         {
             "alarmMetricName" : "4XXError",
             "monitor" : true
         },
         {
             "alarmMetricName" : "5XXError",
             "monitor" : true
         }
     ],
    "logs" : [
        {
            "logType" : "API_GATEWAY_EXECUTION",
            "monitor" : true
        },
        {
            "logType" : "API_GATEWAY_ACCESS",
            "monitor" : true
        }
    ]
}
```

Elastic Load Balancing de aplicaciones

En el siguiente ejemplo se muestra una configuración de componentes en formato JSON para un Application <Elastic Load Balancing (Balanceador de carga elástica de aplicaciones).

```
{
  "alarmMetrics": [
    {
      "alarmMetricName": "ActiveConnectionCount",
    }, {
      "alarmMetricName": "TargetResponseTime"
    }
  ],
  "subComponents": [
    {
      "subComponentType": "AWS::EC2::Instance",
      "alarmMetrics": [
        {
          "alarmMetricName": "CPUUtilization",
        }, {
          "alarmMetricName": "StatusCheckFailed"
        }
      ],
      "logs": [
        {
          "logGroupName": "my_log_group",
          "logPath": "C:\\LogFolder\\*",
          "logType": "APPLICATION",
        }
      ],
      "windowsEvents": [
        {
          "logGroupName": "my_log_group_2",
          "eventName": "Application",
          "eventLevels": [ "ERROR", "WARNING", "CRITICAL" ]
        }
      ]
    }, {
      "subComponentType": "AWS::EC2::Volume",
      "alarmMetrics": [
        {
          "alarmMetricName": "VolumeQueueLength",
        }, {
          "alarmMetricName": "BurstBalance"
```

```
        }
      ]
    }
  ],

  "alarms": [
    {
      "alarmName": "my_alb_alarm",
      "severity": "LOW"
    }
  ]
}
```

AWS Lambda Función

En el siguiente ejemplo se muestra una configuración de componentes en formato JSON para una función de AWS Lambda.

```
{
  "alarmMetrics": [
    {
      "alarmMetricName": "Errors",
      "monitor": true
    },
    {
      "alarmMetricName": "Throttles",
      "monitor": true
    },
    {
      "alarmMetricName": "IteratorAge",
      "monitor": true
    },
    {
      "alarmMetricName": "Duration",
      "monitor": true
    }
  ],
  "logs": [
    {
      "logType": "DEFAULT",
      "monitor": true
    }
  ]
```

```
}
```

Grupo de reglas de AWS Network Firewall

En el siguiente ejemplo se muestra una configuración de componentes en formato JSON para un grupo de reglas AWS Network Firewall.

```
{
  "alarmMetrics": [
    {
      "alarmMetricName": "FirewallRuleGroupQueryVolume",
      "monitor": true
    }
  ]
}
```

Asociación de grupos de reglas AWS Network Firewall

En el siguiente ejemplo se muestra una configuración de componentes en formato JSON para una asociación de grupo de reglas AWS Network Firewall.

```
{
  "alarmMetrics": [
    {
      "alarmMetricName": "FirewallRuleGroupQueryVolume",
      "monitor": true
    }
  ]
}
```

AWS Step Functions

En el siguiente ejemplo se muestra una configuración de componentes en formato JSON para AWS Step Functions.

```
{
  "alarmMetrics": [
    {
      "alarmMetricName": "ExecutionsFailed",
      "monitor": true
    },
    {
```

```
      "alarmMetricName": "LambdaFunctionsFailed",
      "monitor": true
    },
    {
      "alarmMetricName": "ProvisionedRefillRate",
      "monitor": true
    }
  ],
  "logs": [
    {
     "logGroupName": "/aws/states/HelloWorld-Logs",
      "logType": "STEP_FUNCTION",
      "monitor": true,
    }
  ]
}
```

Instancias de Amazon EC2 agrupadas por cliente

En el ejemplo siguiente se muestra una configuración de componentes en formato JSON para instancias de Amazon EC2 agrupadas por clientes.

```
{
    "subComponents": [
      {
        "subComponentType": "AWS::EC2::Instance",
        "alarmMetrics": [
          {
            "alarmMetricName": "CPUUtilization",
          },
          {
            "alarmMetricName": "StatusCheckFailed"
          }
        ],
        "logs": [
          {
            "logGroupName": "my_log_group",
            "logPath": "C:\\LogFolder\\*",
            "logType": "APPLICATION",
          }
        ],
        "processes": [
            {
```

```
                "processName": "my_process",
                "alarmMetrics": [
                    {
                        "alarmMetricName": "procstat cpu_usage",
                        "monitor": true
                    }, {
                        "alarmMetricName": "procstat memory_rss",
                        "monitor": true
                    }
                ]
            }
        ],
        "windowsEvents": [
          {
            "logGroupName": "my_log_group_2",
            "eventName": "Application",
            "eventLevels": [ "ERROR", "WARNING", "CRITICAL" ]
          }
        ]
      }, {
        "subComponentType": "AWS::EC2::Volume",
        "alarmMetrics": [
          {
            "alarmMetricName": "VolumeQueueLength",
          }, {
            "alarmMetricName": "BurstBalance"
          }
        ]
      }
    ],
    "alarms": [
      {
        "alarmName": "my_alarm",
        "severity": "MEDIUM"
      }
    ]
  }
```

Elastic Load Balancing

El siguiente ejemplo muestra una configuración de componentes en formato JSON para Elastic Load Balancing.

```
{
    "alarmMetrics": [
      {
        "alarmMetricName": "EstimatedALBActiveConnectionCount"
      }, {
        "alarmMetricName": "HTTPCode_Backend_5XX"
      }
    ],
    "subComponents": [
      {
        "subComponentType": "AWS::EC2::Instance",
        "alarmMetrics": [
          {
            "alarmMetricName": "CPUUtilization"
          }, {
            "alarmMetricName": "StatusCheckFailed"
          }
        ],
        "logs": [
          {
            "logGroupName": "my_log_group",
            "logPath": "C:\\LogFolder\\*",
            "logType": "APPLICATION"
          }
        ],
        "processes": [
          {
            "processName": "my_process",
            "alarmMetrics": [
                {
                  "alarmMetricName": "procstat cpu_usage",
                  "monitor": true
                }, {
                  "alarmMetricName": "procstat memory_rss",
                  "monitor": true
                }
             ]
          }
        ],
        "windowsEvents": [
          {
            "logGroupName": "my_log_group_2",
            "eventName": "Application",
```

```
            "eventLevels": [ "ERROR", "WARNING", "CRITICAL" ],
            "monitor": true
          }
        ]
      }, {
        "subComponentType": "AWS::EC2::Volume",
        "alarmMetrics": [
          {
            "alarmMetricName": "VolumeQueueLength"
          }, {
            "alarmMetricName": "BurstBalance"
          }
        ]
      }
    ],

    "alarms": [
      {
        "alarmName": "my_elb_alarm",
        "severity": "HIGH"
      }
    ]
  }
```

Java

En el siguiente ejemplo se muestra una configuración de componentes en formato JSON para Java.

```
{
  "alarmMetrics": [ {
    "alarmMetricName": "java_lang_threading_threadcount",
    "monitor": true
  },
  {
    "alarmMetricName": "java_lang_memory_heapmemoryusage_used",
    "monitor": true
  },
  {
    "alarmMetricName": "java_lang_memory_heapmemoryusage_committed",
    "monitor": true
  }],
  "logs": [ ],
  "JMXPrometheusExporter": {
      "hostPort": "8686",
```

```
      "prometheusPort": "9404"
  }
}
```

> **Note**
>
> Información de aplicaciones no admite la configuración de autenticación para el JMX Exporter de Prometheus. Para obtener información acerca de cómo se configura la autenticación, consulte la Prometheus JMX Exporter example configuration (Configuración de ejemplo del exportador JMX de Prometheus).

Kubernetes en Amazon EC2

En el siguiente ejemplo se muestra una configuración de componentes en formato JSON para Kubernetes en Amazon EC2.

```
{
    "alarmMetrics":[
       {
          "alarmMetricName":"cluster_failed_node_count",
          "monitor":true
       },
       {
          "alarmMetricName":"node_cpu_reserved_capacity",
          "monitor":true
       },
       {
          "alarmMetricName":"node_cpu_utilization",
          "monitor":true
       },
       {
          "alarmMetricName":"node_filesystem_utilization",
          "monitor":true
       },
       {
          "alarmMetricName":"node_memory_reserved_capacity",
          "monitor":true
       },
       {
          "alarmMetricName":"node_memory_utilization",
          "monitor":true
```

```
        },
        {
           "alarmMetricName":"node_network_total_bytes",
           "monitor":true
        },
        {
           "alarmMetricName":"pod_cpu_reserved_capacity",
           "monitor":true
        },
        {
           "alarmMetricName":"pod_cpu_utilization",
           "monitor":true
        },
        {
           "alarmMetricName":"pod_cpu_utilization_over_pod_limit",
           "monitor":true
        },
        {
           "alarmMetricName":"pod_memory_reserved_capacity",
           "monitor":true
        },
        {
           "alarmMetricName":"pod_memory_utilization",
           "monitor":true
        },
        {
           "alarmMetricName":"pod_memory_utilization_over_pod_limit",
           "monitor":true
        },
        {
           "alarmMetricName":"pod_network_rx_bytes",
           "monitor":true
        },
        {
           "alarmMetricName":"pod_network_tx_bytes",
           "monitor":true
        }
     ],
     "logs":[
        {
           "logGroupName":"/aws/containerinsights/kubernetes/application",
           "logType":"APPLICATION",
           "monitor":true,
           "encoding":"utf-8"
```

```
      }
   ],
   "subComponents":[
      {
         "subComponentType":"AWS::EC2::Instance",
         "alarmMetrics":[
            {
               "alarmMetricName":"CPUUtilization",
               "monitor":true
            },
            {
               "alarmMetricName":"StatusCheckFailed",
               "monitor":true
            },
            {
               "alarmMetricName":"disk_used_percent",
               "monitor":true
            },
            {
               "alarmMetricName":"mem_used_percent",
               "monitor":true
            }
         ],
         "logs":[
            {
               "logGroupName":"APPLICATION-KubernetesClusterOnEC2-IAD",
               "logPath":"",
               "logType":"APPLICATION",
               "monitor":true,
               "encoding":"utf-8"
            }
         ],
         "processes" : [
           {
               "processName" : "my_process",
               "alarmMetrics" : [
                   {
                       "alarmMetricName" : "procstat cpu_usage",
                       "monitor" : true
                   }, {
                       "alarmMetricName" : "procstat memory_rss",
                       "monitor" : true
                   }
               ]
```

```
            }
         ]
       },
       {
          "subComponentType":"AWS::EC2::Volume",
          "alarmMetrics":[
             {
                "alarmMetricName":"VolumeReadBytes",
                "monitor":true
             },
             {
                "alarmMetricName":"VolumeWriteBytes",
                "monitor":true
             },
             {
                "alarmMetricName":"VolumeReadOps",
                "monitor":true
             },
             {
                "alarmMetricName":"VolumeWriteOps",
                "monitor":true
             },
             {
                "alarmMetricName":"VolumeQueueLength",
                "monitor":true
             },
             {
                "alarmMetricName":"BurstBalance",
                "monitor":true
             }
          ]
       }
    ]
 }
```

RDS MariaDB y RDS MySQL

En el siguiente ejemplo se muestra una configuración de componentes en formato JSON para RDS MariaDB y RDS MySQL

```
{
  "alarmMetrics": [
    {
```

```
      "alarmMetricName": "CPUUtilization",
      "monitor": true
    }
  ],
  "logs": [
    {
      "logType": "MYSQL",
      "monitor": true,
    },
    {
      "logType": "MYSQL_SLOW_QUERY",
      "monitor": false
    }
  ]
}
```

Oracle de RDS

En el siguiente ejemplo se muestra una configuración de componentes en formato JSON para RDS para Oracle.

```
{
  "alarmMetrics": [
    {
      "alarmMetricName": "CPUUtilization",
      "monitor": true
    }
  ],
  "logs": [
    {
      "logType": "ORACLE_ALERT",
      "monitor": true,
    },
    {
      "logType": "ORACLE_LISTENER",
      "monitor": false
    }
  ]
}
```

PostgreSQL de RDS

En el siguiente ejemplo se muestra una configuración de componentes en formato JSON para RDS para PostgreSQL.

```
{
  "alarmMetrics": [
    {
      "alarmMetricName": "CPUUtilization",
      "monitor": true
    }
  ],
  "logs": [
    {
      "logType": "POSTGRESQL",
      "monitor": true
    }
  ]
}
```

SAP ASE en Amazon EC2

En el siguiente ejemplo se muestra una configuración de componentes en formato JSON para SAP ASE en Amazon EC2.

```
{
  "subComponents": [
    {
      "subComponentType": "AWS::EC2::Instance",
      "alarmMetrics": [
        {
          "alarmMetricName": "asedb_database_availability",
          "monitor": true
        },
        {
          "alarmMetricName": "asedb_trunc_log_on_chkpt_enabled",
          "monitor": true
        },
        {
          "alarmMetricName": "asedb_last_db_backup_age_in_days",
          "monitor": true
        },
        {
```

```
          "alarmMetricName": "asedb_last_transaction_log_backup_age_in_hours",
          "monitor": true
        },
        {
          "alarmMetricName": "asedb_suspected_database",
          "monitor": true
        },
        {
          "alarmMetricName": "asedb_db_space_usage_percent",
          "monitor": true
        },
        {
          "alarmMetricName": "asedb_db_log_space_usage_percent",
          "monitor": true
        },
        {
          "alarmMetricName": "asedb_locked_login",
          "monitor": true
        },
        {
          "alarmMetricName": "asedb_data_cache_hit_ratio",
          "monitor": true
        }
      ],
      "logs": [
        {
          "logGroupName": "SAP_ASE_SERVER_LOGS-my-resource-group",
          "logPath": "/sybase/SY2/ASE-*/install/SY2.log",
          "logType": "SAP_ASE_SERVER_LOGS",
          "monitor": true,
          "encoding": "utf-8"
        },
        {
          "logGroupName": "SAP_ASE_BACKUP_SERVER_LOGS-my-resource-group",
          "logPath": "/sybase/SY2/ASE-*/install/SY2_BS.log",
          "logType": "SAP_ASE_BACKUP_SERVER_LOGS",
          "monitor": true,
          "encoding": "utf-8"
        }
    ],
    "sapAsePrometheusExporter": {
      "sapAseSid": "ASE",
      "sapAsePort": "4901",
      "sapAseSecretName": "ASE_DB_CREDS",
```

```
    "prometheusPort": "9399",
    "agreeToEnableASEMonitoring": true
}
```

Alta disponibilidad de SAP ASE en Amazon EC2

En el ejemplo siguiente se muestra una configuración de componentes en formato JSON para una alta disponibilidad de SAP ASE en Amazon EC2.

```
{
  "subComponents": [
    {
      "subComponentType": "AWS::EC2::Instance",
      "alarmMetrics": [
        {
          "alarmMetricName": "asedb_database_availability",
          "monitor": true
        },
        {
          "alarmMetricName": "asedb_trunc_log_on_chkpt_enabled",
          "monitor": true
        },
        {
          "alarmMetricName": "asedb_last_db_backup_age_in_days",
          "monitor": true
        },
        {
          "alarmMetricName": "asedb_last_transaction_log_backup_age_in_hours",
          "monitor": true
        },
        {
          "alarmMetricName": "asedb_suspected_database",
          "monitor": true
        },
        {
          "alarmMetricName": "asedb_db_space_usage_percent",
          "monitor": true
        },
        {
          "alarmMetricName": "asedb_ha_replication_state",
          "monitor": true
        },
        {
```

```
          "alarmMetricName": "asedb_ha_replication_mode",
          "monitor": true
        },
        {
          "alarmMetricName": "asedb_ha_replication_latency_in_minutes",
          "monitor": true
        }
      ],
      "logs": [
        {
          "logGroupName": "SAP_ASE_SERVER_LOGS-my-resource-group",
          "logPath": "/sybase/SY2/ASE-*/install/SY2.log",
          "logType": "SAP_ASE_SERVER_LOGS",
          "monitor": true,
          "encoding": "utf-8"
        },
        {
          "logGroupName": "SAP_ASE_BACKUP_SERVER_LOGS-my-resource-group",
          "logPath": "/sybase/SY2/ASE-*/install/SY2_BS.log",
          "logType": "SAP_ASE_BACKUP_SERVER_LOGS",
          "monitor": true,
          "encoding": "utf-8"
        },
        {
          "logGroupName": "SAP_ASE_REP_SERVER_LOGS-my-resource-group",
          "logPath": "/sybase/SY2/DM/repservername/repservername.log",
          "logType": "SAP_ASE_REP_SERVER_LOGS",
          "monitor": true,
          "encoding": "utf-8"
        },
        {
          "logGroupName": "SAP_ASE_RMA_AGENT_LOGS-my-resource-group",
          "logPath": "/sybase/SY2/DM/RMA-*/instances/AgentContainer/logs/",
          "logType": "SAP_ASE_RMA_AGENT_LOGS",
          "monitor": true,
          "encoding": "utf-8"
        },
        {
          "logGroupName": "SAP_ASE_FAULT_MANAGER_LOGS-my-resource-group",
          "logPath": "/opt/sap/FaultManager/dev_sybdbfm",
          "logType": "SAP_ASE_FAULT_MANAGER_LOGS",
          "monitor": true,
          "encoding": "utf-8"
        }
```

```
  ],
  "sapAsePrometheusExporter": {
    "sapAseSid": "ASE",
    "sapAsePort": "4901",
    "sapAseSecretName": "ASE_DB_CREDS",
    "prometheusPort": "9399",
    "agreeToEnableASEMonitoring": true
}
```

SAP HANA en Amazon EC2

En el siguiente ejemplo se muestra una configuración de componentes en formato JSON para SAP HANA en Amazon EC2.

```
{
  "subComponents": [
    {
      "subComponentType": "AWS::EC2::Instance",
      "alarmMetrics": [
        {
          "alarmMetricName": "hanadb_server_startup_time_variations_seconds",
          "monitor": true
        },
        {
          "alarmMetricName": "hanadb_level_5_alerts_count",
          "monitor": true
        },
        {
          "alarmMetricName": "hanadb_level_4_alerts_count",
          "monitor": true
        },
        {
          "alarmMetricName": "hanadb_out_of_memory_events_count",
          "monitor": true
        },
        {
          "alarmMetricName": "hanadb_max_trigger_read_ratio_percent",
          "monitor": true
        },
        {
          "alarmMetricName": "hanadb_table_allocation_limit_used_percent",
          "monitor": true
        },
```

```
        {
          "alarmMetricName": "hanadb_cpu_usage_percent",
          "monitor": true
        },
        {
          "alarmMetricName": "hanadb_plan_cache_hit_ratio_percent",
          "monitor": true
        },
        {
          "alarmMetricName": "hanadb_last_data_backup_age_days",
          "monitor": true
        }
      ],
      "logs": [
        {
          "logGroupName": "SAP_HANA_TRACE-my-resourge-group",
          "logPath": "/usr/sap/HDB/HDB00/*/trace/*.trc",
          "logType": "SAP_HANA_TRACE",
          "monitor": true,
          "encoding": "utf-8"
        },
        {
          "logGroupName": "SAP_HANA_LOGS-my-resource-group",
          "logPath": "/usr/sap/HDB/HDB00/*/trace/*.log",
          "logType": "SAP_HANA_LOGS",
          "monitor": true,
          "encoding": "utf-8"
        }
      ]
    }
  ],
  "hanaPrometheusExporter": {
    "hanaSid": "HDB",
    "hanaPort": "30013",
    "hanaSecretName": "HANA_DB_CREDS",
    "prometheusPort": "9668"
  }
}
```

Alta disponibilidad de SAP HANA en Amazon EC2

En el ejemplo siguiente se muestra una configuración de componentes en formato JSON para alta disponibilidad de SAP HANA en Amazon EC2.

```
{
  "subComponents": [
    {
      "subComponentType": "AWS::EC2::Instance",
      "alarmMetrics": [
        {
          "alarmMetricName": "hanadb_server_startup_time_variations_seconds",
          "monitor": true
        },
        {
          "alarmMetricName": "hanadb_level_5_alerts_count",
          "monitor": true
        },
        {
          "alarmMetricName": "hanadb_level_4_alerts_count",
          "monitor": true
        },
        {
          "alarmMetricName": "hanadb_out_of_memory_events_count",
          "monitor": true
        },
        {
          "alarmMetricName": "ha_cluster_pacemaker_stonith_enabled",
          "monitor": true
        }
      ],
      "logs": [
        {
          "logGroupName": "SAP_HANA_TRACE-my-resourge-group",
          "logPath": "/usr/sap/HDB/HDB00/*/trace/*.trc",
          "logType": "SAP_HANA_TRACE",
          "monitor": true,
          "encoding": "utf-8"
        },
        {
          "logGroupName": "SAP_HANA_HIGH_AVAILABILITY-my-resource-group",
          "logPath": "/var/log/pacemaker/pacemaker.log",
          "logType": "SAP_HANA_HIGH_AVAILABILITY",
          "monitor": true,
          "encoding": "utf-8"
        }
      ]
    }
```

```
  ],
  "hanaPrometheusExporter": {
    "hanaSid": "HDB",
    "hanaPort": "30013",
    "hanaSecretName": "HANA_DB_CREDS",
    "prometheusPort": "9668"
  },
  "haClusterPrometheusExporter": {
    "prometheusPort": "9664"
  }
}
```

SAP NetWeaver en Amazon EC2

En el siguiente ejemplo, se muestra una configuración de componentes en formato JSON para SAP NetWeaver en Amazon EC2.

```
{
  "subComponents": [
    {
      "subComponentType": "AWS::EC2::Instance",
      "alarmMetrics": [
        {
          "alarmMetricName": "CPUUtilization",
          "monitor": true
        },
        {
          "alarmMetricName": "StatusCheckFailed",
          "monitor": true
        },
        {
          "alarmMetricName": "disk_used_percent",
          "monitor": true
        },
        {
          "alarmMetricName": "mem_used_percent",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_alerts_ResponseTime",
          "monitor": true
        },
        {
```

```
            "alarmMetricName": "sap_alerts_ResponseTimeDialog",
            "monitor": true
          },
          {
            "alarmMetricName": "sap_alerts_ResponseTimeDialogRFC",
            "monitor": true
          },
          {
            "alarmMetricName": "sap_alerts_DBRequestTime",
            "monitor": true
          },
          {
            "alarmMetricName": "sap_alerts_LongRunners",
            "monitor": true
          },
          {
            "alarmMetricName": "sap_alerts_AbortedJobs",
            "monitor": true
          },
          {
            "alarmMetricName": "sap_alerts_BasisSystem",
            "monitor": true
          },
          {
            "alarmMetricName": "sap_alerts_Database",
            "monitor": true
          },
          {
            "alarmMetricName": "sap_alerts_Security",
            "monitor": true
          },
          {
            "alarmMetricName": "sap_alerts_System",
            "monitor": true
          },
          {
            "alarmMetricName": "sap_alerts_QueueTime",
            "monitor": true
          },
          {
            "alarmMetricName": "sap_alerts_Availability",
            "monitor": true
          },
          {
```

```
          "alarmMetricName": "sap_start_service_processes",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_dispatcher_queue_now",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_dispatcher_queue_max",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_enqueue_server_locks_max",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_enqueue_server_locks_now",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_enqueue_server_locks_state",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_enqueue_server_replication_state",
          "monitor": true
        }
      ],
      "logs": [
        {
          "logGroupName": "SAP_NETWEAVER_DEV_TRACE_LOGS-NetWeaver-ML4",
          "logPath": "/usr/sap/ML4/*/work/dev_w*",
          "logType": "SAP_NETWEAVER_DEV_TRACE_LOGS",
          "monitor": true,
          "encoding": "utf-8"
        }
      ]
    }
  ],
  "netWeaverPrometheusExporter": {
    "sapSid": "ML4",
    "instanceNumbers": [
      "00",
      "11"
```

```
    ],
    "prometheusPort": "9680"
  }
}
```

Alta disponibilidad de SAP NetWeaver en Amazon EC2

En el ejemplo siguiente, se muestra una configuración de componentes en formato JSON para alta disponibilidad de SAP NetWeaver en Amazon EC2.

```
{
  "subComponents": [
    {
      "subComponentType": "AWS::EC2::Instance",
      "alarmMetrics": [
        {
          "alarmMetricName": "ha_cluster_corosync_ring_errors",
          "monitor": true
        },
        {
          "alarmMetricName": "ha_cluster_pacemaker_fail_count",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_HA_check_failover_config_state",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_HA_get_failover_config_HAActive",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_alerts_AbortedJobs",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_alerts_Availability",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_alerts_BasisSystem",
          "monitor": true
        },
```

```
        {
          "alarmMetricName": "sap_alerts_DBRequestTime",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_alerts_Database",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_alerts_FrontendResponseTime",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_alerts_LongRunners",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_alerts_QueueTime",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_alerts_ResponseTime",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_alerts_ResponseTimeDialog",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_alerts_ResponseTimeDialogRFC",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_alerts_Security",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_alerts_Shortdumps",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_alerts_SqlError",
          "monitor": true
        },
```

```
        {
          "alarmMetricName": "sap_alerts_System",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_enqueue_server_replication_state",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_start_service_processes",
          "monitor": true
        }
      ],
      "logs": [
        {
          "logGroupName": "SAP_NETWEAVER_DEV_TRACE_LOGS-NetWeaver-PR1",
          "logPath": "/usr/sap/<SID>/D*/work/dev_w*",
          "logType": "SAP_NETWEAVER_DEV_TRACE_LOGS",
          "monitor": true,
          "encoding": "utf-8"
        }
      ]
    }
  ],
  "haClusterPrometheusExporter": {
    "prometheusPort": "9664"
  },
  "netWeaverPrometheusExporter": {
    "sapSid": "PR1",
    "instanceNumbers": [
      "11",
      "12"
    ],
    "prometheusPort": "9680"
  }
}
```

Grupos de disponibilidad siempre activos de SQL

En el siguiente ejemplo se muestra una configuración de componentes en formato JSON para SQL Always On Availability Group.

```
{
```

```
  "subComponents" : [ {
    "subComponentType" : "AWS::EC2::Instance",
    "alarmMetrics" : [ {
      "alarmMetricName" : "CPUUtilization",
      "monitor" : true
    }, {
      "alarmMetricName" : "StatusCheckFailed",
      "monitor" : true
    }, {
      "alarmMetricName" : "Processor % Processor Time",
      "monitor" : true
    }, {
      "alarmMetricName" : "Memory % Committed Bytes In Use",
      "monitor" : true
    }, {
      "alarmMetricName" : "Memory Available Mbytes",
      "monitor" : true
    }, {
      "alarmMetricName" : "Paging File % Usage",
      "monitor" : true
    }, {
      "alarmMetricName" : "System Processor Queue Length",
      "monitor" : true
    }, {
      "alarmMetricName" : "Network Interface Bytes Total/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "PhysicalDisk % Disk Time",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Buffer Manager Buffer cache hit ratio",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Buffer Manager Page life expectancy",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:General Statistics Processes blocked",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:General Statistics User Connections",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Locks Number of Deadlocks/sec",
      "monitor" : true
```

```
    }, {
      "alarmMetricName" : "SQLServer:SQL Statistics Batch Requests/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Database Replica File Bytes Received/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Database Replica Log Bytes Received/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Database Replica Log remaining for undo",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Database Replica Log Send Queue",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Database Replica Mirrored Write Transaction/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Database Replica Recovery Queue",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Database Replica Redo Bytes Remaining",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Database Replica Redone Bytes/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Database Replica Total Log requiring undo",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Database Replica Transaction Delay",
      "monitor" : true
    } ],
    "windowsEvents" : [ {
      "logGroupName" : "WINDOWS_EVENTS-Application-<RESOURCE_GROUP_NAME>",
      "eventName" : "Application",
      "eventLevels" : [ "WARNING", "ERROR", "CRITICAL", "INFORMATION" ],
      "monitor" : true
    }, {
      "logGroupName" : "WINDOWS_EVENTS-System-<RESOURCE_GROUP_NAME>",
      "eventName" : "System",
      "eventLevels" : [ "WARNING", "ERROR", "CRITICAL" ],
      "monitor" : true
```

```
    }, {
      "logGroupName" : "WINDOWS_EVENTS-Security-<RESOURCE_GROUP_NAME>",
      "eventName" : "Security",
      "eventLevels" : [ "WARNING", "ERROR", "CRITICAL" ],
      "monitor" : true
    } ],
    "logs" : [ {
      "logGroupName" : "SQL_SERVER_ALWAYSON_AVAILABILITY_GROUP-<RESOURCE_GROUP_NAME>",
      "logPath" : "C:\\Program Files\\Microsoft SQL Server\\MSSQL**.MSSQLSERVER\\MSSQL\
\Log\\ERRORLOG",
      "logType" : "SQL_SERVER",
      "monitor" : true,
      "encoding" : "utf-8"
    } ]
  }, {
    "subComponentType" : "AWS::EC2::Volume",
    "alarmMetrics" : [ {
      "alarmMetricName" : "VolumeReadBytes",
      "monitor" : true
    }, {
    "alarmMetricName" : "VolumeWriteBytes",
      "monitor" : true
    }, {
    "alarmMetricName" : "VolumeReadOps",
      "monitor" : true
    }, {
    "alarmMetricName" : "VolumeWriteOps",
      "monitor" : true
    }, {
    "alarmMetricName" : "VolumeQueueLength",
      "monitor" : true
    }, {
    "alarmMetricName" : "VolumeThroughputPercentage",
      "monitor" : true
    }, {
    "alarmMetricName" : "BurstBalance",
      "monitor" : true
    } ]
  } ]
}
```

Instancia de clúster de conmutación por error de SQL

En el siguiente ejemplo se muestra una configuración de componentes en formato JSON para la instancia de clúster de conmutación por error de SQL.

```
{
  "subComponents" : [ {
    "subComponentType" : "AWS::EC2::Instance",
    "alarmMetrics" : [ {
      "alarmMetricName" : "CPUUtilization",
      "monitor" : true
    }, {
      "alarmMetricName" : "StatusCheckFailed",
      "monitor" : true
    }, {
      "alarmMetricName" : "Processor % Processor Time",
      "monitor" : true
    }, {
      "alarmMetricName" : "Memory % Committed Bytes In Use",
      "monitor" : true
    }, {
      "alarmMetricName" : "Memory Available Mbytes",
      "monitor" : true
    }, {
      "alarmMetricName" : "Paging File % Usage",
      "monitor" : true
    }, {
      "alarmMetricName" : "System Processor Queue Length",
      "monitor" : true
    }, {
      "alarmMetricName" : "Network Interface Bytes Total/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "PhysicalDisk % Disk Time",
      "monitor" : true
    }, {
      "alarmMetricName" : "Bytes Received/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "Normal Messages Queue Length/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "Urgent Message Queue Length/se",
      "monitor" : true
```

```
    }, {
      "alarmMetricName" : "Reconnect Count",
      "monitor" : true
    }, {
      "alarmMetricName" : "Unacknowledged Message Queue Length/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "Messages Outstanding",
      "monitor" : true
    }, {
      "alarmMetricName" : "Messages Sent/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "Database Update Messages/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "Update Messages/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "Flushes/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "Crypto Checkpoints Saved/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "Crypto Checkpoints Restored/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "Registry Checkpoints Restored/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "Registry Checkpoints Saved/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "Cluster API Calls/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "Resource API Calls/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "Cluster Handles/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "Resource Handles/sec",
```

```
      "monitor" : true
    } ],
    "windowsEvents" : [ {
      "logGroupName" : "WINDOWS_EVENTS-Application-<RESOURCE_GROUP_NAME>",
      "eventName" : "Application",
      "eventLevels" : [ "WARNING", "ERROR", "CRITICAL"],
      "monitor" : true
    }, {
      "logGroupName" : "WINDOWS_EVENTS-System-<RESOURCE_GROUP_NAME>",
      "eventName" : "System",
      "eventLevels" : [ "WARNING", "ERROR", "CRITICAL", "INFORMATION" ],
      "monitor" : true
    }, {
      "logGroupName" : "WINDOWS_EVENTS-Security-<RESOURCE_GROUP_NAME>",
      "eventName" : "Security",
      "eventLevels" : [ "WARNING", "ERROR", "CRITICAL" ],
      "monitor" : true
    } ],
    "logs" : [ {
      "logGroupName" : "SQL_SERVER_FAILOVER_CLUSTER_INSTANCE-<RESOURCE_GROUP_NAME>",
      "logPath" : "\\\\amznfsxjmzbykwn.mydomain.aws\\SQLDB\\MSSQL**.MSSQLSERVER\\MSSQL\
\Log\\ERRORLOG",
      "logType" : "SQL_SERVER",
      "monitor" : true,
      "encoding" : "utf-8"
    } ]
  }, {
    "subComponentType" : "AWS::EC2::Volume",
    "alarmMetrics" : [ {
      "alarmMetricName" : "VolumeReadBytes",
      "monitor" : true
    }, {
    "alarmMetricName" : "VolumeWriteBytes",
      "monitor" : true
    }, {
    "alarmMetricName" : "VolumeReadOps",
      "monitor" : true
    }, {
    "alarmMetricName" : "VolumeWriteOps",
      "monitor" : true
    }, {
    "alarmMetricName" : "VolumeQueueLength",
      "monitor" : true
    }, {
```

```
    "alarmMetricName" : "VolumeThroughputPercentage",
      "monitor" : true
    }, {
    "alarmMetricName" : "BurstBalance",
      "monitor" : true
    } ]
  } ]
}
```

Creación y configuración de la supervisión de Información de aplicaciones de CloudWatch mediante las plantillas de CloudFormation

Puede agregar el monitoreo de Información de aplicaciones, incluidas las métricas clave y la telemetría, a su aplicación, base de datos y servidor web, directamente desde las plantillas de AWS CloudFormation.

En esta sección encontrará un ejemplo de plantillas de AWS CloudFormation en formatos JSON y YAML para ayudarlo a crear y configurar el monitoreo de Información de aplicaciones.

Para ver la referencia de recursos y propiedades de Información de aplicaciones en la Guía del usuario de AWS CloudFormation, consulte Referencia de tipos de recursos de Información de aplicaciones.

Plantillas de de ejemplo

- Creación de una aplicación de Información de aplicaciones para toda la pila de AWS CloudFormation
- Creación de una aplicación de Información de aplicaciones con configuraciones detalladas
- Creación de una aplicación de Información de aplicaciones con la configuración del componente en modo CUSTOM
- Creación de una aplicación de Información de aplicaciones con la configuración del componente en modo DEFAULT
- Creación de una aplicación de Información de aplicaciones con la configuración del componente en modo DEFAULT_WITH_OVERWRITE

Creación de una aplicación de Información de aplicaciones para toda la pila de AWS CloudFormation

Para aplicar la siguiente plantilla, cree recursos de AWS y uno o más grupos de recursos a partir de los cuales pueda crear aplicaciones de Información de aplicaciones para monitorear esos recursos. Para obtener más información, consulte Introducción a AWS Resource Groups.

Las dos primeras partes de la siguiente plantilla especifican un recurso y un grupo de recursos. La última parte de la plantilla crea una aplicación de Información de aplicaciones para el grupo de recursos, pero no configura la aplicación ni aplica monitoreo. Para obtener más información, consulte los detalles del comando CreateApplication en la Referencia de la API de Información de aplicaciones de Amazon CloudWatch.

Plantilla en formato JSON

```
{
    "AWSTemplateFormatVersion": "2010-09-09",
    "Description": "Test Resource Group stack",
    "Resources": {
        "EC2Instance": {
            "Type": "AWS::EC2::Instance",
            "Properties": {
                "ImageId" : "ami-abcd1234efgh5678i",
                "SecurityGroupIds" : ["sg-abcd1234"]
            }
        },
        ...
        "ResourceGroup": {
            "Type": "AWS::ResourceGroups::Group",
            "Properties": {
                "Name": "my_resource_group"
            }
        },
        "AppInsightsApp": {
            "Type": "AWS::ApplicationInsights::Application",
            "Properties": {
                "ResourceGroupName": "my_resource_group"
            },
            "DependsOn" : "ResourceGroup"
        }
    }
}
```

Plantilla en formato YAML

```
---
AWSTemplateFormatVersion: '2010-09-09'
Description: Test Resource Group stack
Resources:
  EC2Instance:
    Type: AWS::EC2::Instance
    Properties:
      ImageId: ami-abcd1234efgh5678i
      SecurityGroupIds:
      - sg-abcd1234
  ...
  ResourceGroup:
    Type: AWS::ResourceGroups::Group
    Properties:
      Name: my_resource_group
  AppInsightsApp:
    Type: AWS::ApplicationInsights::Application
    Properties:
      ResourceGroupName: my_resource_group
    DependsOn: ResourceGroup
```

La siguiente sección de plantilla aplica la configuración de monitoreo predeterminada a la aplicación de Información de aplicaciones. Para obtener más información, consulte los detalles del comando CreateApplication en la Referencia de la API de Información de aplicaciones de Amazon CloudWatch.

Cuando `AutoConfigurationEnabled` se establece en `true`, todos los componentes de la aplicación se configuran con las configuraciones recomendadas de monitoreo para el nivel de aplicación `DEFAULT`. Para obtener más información acerca de estas configuraciones y niveles, consulte DescribeComponentConfigurationRecommendation y UpdateComponentConfiguration en la Referencia de la API de Información de aplicaciones de Amazon CloudWatch.

Plantilla en formato JSON

```
{
    "AWSTemplateFormatVersion": "2010-09-09",
    "Description": "Test Application Insights Application stack",
    "Resources": {
        "AppInsightsApp": {
            "Type": "AWS::ApplicationInsights::Application",
```

```
            "Properties": {
                "ResourceGroupName": "my_resource_group",
                "AutoConfigurationEnabled": true
            }
        }
    }
}
```

Plantilla en formato YAML

```
---
AWSTemplateFormatVersion: '2010-09-09'
Description: Test Application Insights Application stack
Resources:
  AppInsightsApp:
    Type: AWS::ApplicationInsights::Application
    Properties:
      ResourceGroupName: my_resource_group
      AutoConfigurationEnabled: true
```

Creación de una aplicación de Información de aplicaciones con configuraciones detalladas

En la siguiente plantilla se realizan las siguientes acciones:

- Crea una aplicación de Información de aplicaciones con la notificación de Eventos de CloudWatch y Centro de operaciones habilitado. Para obtener más información, consulte los detalles del comando CreateApplication en la Referencia de la API de Información de aplicaciones de Amazon CloudWatch.
- Etiqueta la aplicación con dos etiquetas, una de las cuales no tiene valores de etiqueta. Para obtener más información, consulte TagResource en la Referencia de la API de Información de aplicaciones de Amazon CloudWatch.
- Crea dos componentes de grupo de instancias personalizado. Para obtener más información, consulte CreateComponent en la Referencia de la API de Información de aplicaciones de Amazon CloudWatch.
- Crea dos conjuntos de patrones de registro. Para obtener más información, consulte CreateLogPattern en la Referencia de la API de Información de aplicaciones de Amazon CloudWatch.

- Establece `AutoConfigurationEnabled` en `true`, que configura todos los componentes de la aplicación con la configuración de monitoreo recomendada para el nivel `DEFAULT`. Para obtener más información, consulte DescribeComponentConfigurationRecommendation en la Referencia de la API de Información de aplicaciones de Amazon CloudWatch.

Plantilla en formato JSON

```
{
    "Type": "AWS::ApplicationInsights::Application",
    "Properties": {
        "ResourceGroupName": "my_resource_group",
        "CWEMonitorEnabled": true,
        "OpsCenterEnabled": true,
        "OpsItemSNSTopicArn": "arn:aws:sns:us-east-1:123456789012:my_topic",
        "AutoConfigurationEnabled": true,
        "Tags": [
            {
                "Key": "key1",
                "Value": "value1"
            },
            {
                "Key": "key2",
                "Value": ""
            }
        ],
        "CustomComponents": [
            {
                "ComponentName": "test_component_1",
                "ResourceList": [
                    "arn:aws:ec2:us-east-1:123456789012:instance/i-abcd1234efgh5678i"
                ]
            },
            {
                "ComponentName": "test_component_2",
                "ResourceList": [
                    "arn:aws:ec2:us-east-1:123456789012:instance/i-abcd1234efgh5678i",
                    "arn:aws:ec2:us-east-1:123456789012:instance/i-abcd1234efgh5678i"
                ]
            }
        ],
        "LogPatternSets": [
            {
```

```
                "PatternSetName": "pattern_set_1",
                "LogPatterns": [
                    {
                        "PatternName": "deadlock_pattern",
                        "Pattern": ".*\\sDeadlocked\\sSchedulers(([^\\w].*)|($))",
                        "Rank": 1
                    }
                ]
            },
            {
                "PatternSetName": "pattern_set_2",
                "LogPatterns": [
                    {
                        "PatternName": "error_pattern",
                        "Pattern": ".*[\\s\\[]ERROR[\\s\\]].*",
                        "Rank": 1
                    },
                    {
                        "PatternName": "warning_pattern",
                        "Pattern": ".*[\\s\\[]WARN(ING)?[\\s\\]].*",
                        "Rank": 10
                    }
                ]
            }
        ]
    }
}
```

Plantilla en formato YAML

```
---
Type: AWS::ApplicationInsights::Application
Properties:
  ResourceGroupName: my_resource_group
  CWEMonitorEnabled: true
  OpsCenterEnabled: true
  OpsItemSNSTopicArn: arn:aws:sns:us-east-1:123456789012:my_topic
  AutoConfigurationEnabled: true
  Tags:
  - Key: key1
    Value: value1
  - Key: key2
    Value: ''
```

```
  CustomComponents:
  - ComponentName: test_component_1
    ResourceList:
    - arn:aws:ec2:us-east-1:123456789012:instance/i-abcd1234efgh5678i
  - ComponentName: test_component_2
    ResourceList:
    - arn:aws:ec2:us-east-1:123456789012:instance/i-abcd1234efgh5678i
    - arn:aws:ec2:us-east-1:123456789012:instance/i-abcd1234efgh5678i
  LogPatternSets:
  - PatternSetName: pattern_set_1
    LogPatterns:
    - PatternName: deadlock_pattern
      Pattern: ".*\\sDeadlocked\\sSchedulers(([^\\w].*)|($))"
      Rank: 1
  - PatternSetName: pattern_set_2
    LogPatterns:
    - PatternName: error_pattern
      Pattern: ".*[\\s\\[]ERROR[\\s\\]].*"
      Rank: 1
    - PatternName: warning_pattern
      Pattern: ".*[\\s\\[]WARN(ING)?[\\s\\]].*"
      Rank: 10
```

Creación de una aplicación de Información de aplicaciones con la configuración del componente en modo **CUSTOM**

En la siguiente plantilla se realizan las siguientes acciones:

- Crea una aplicación de Información de aplicaciones. Para obtener más información, consulte CreateApplication en la Referencia de la API de Información de aplicaciones de Amazon CloudWatch.
- El componente `my_component` establece `ComponentConfigurationMode` en `CUSTOM`, lo que hace que este componente se configure como se especifica en `CustomComponentConfiguration`. Para obtener más información, consulte UpdateComponentConfiguration en la Referencia de la API de Información de aplicaciones de Amazon CloudWatch.

Plantilla en formato JSON

```
{
```

```
    "Type": "AWS::ApplicationInsights::Application",
    "Properties": {
        "ResourceGroupName": "my_resource_group,
        "ComponentMonitoringSettings": [
            {
                "ComponentARN": "my_component",
                "Tier": "SQL_SERVER",
                "ComponentConfigurationMode": "CUSTOM",
                "CustomComponentConfiguration": {
                    "ConfigurationDetails": {
                        "AlarmMetrics": [
                            {
                                "AlarmMetricName": "StatusCheckFailed"
                            },
                            ...
                        ],
                        "Logs": [
                            {
                                "LogGroupName": "my_log_group_1",
                                "LogPath": "C:\\LogFolder_1\\*",
                                "LogType": "DOT_NET_CORE",
                                "Encoding": "utf-8",
                                "PatternSet": "my_pattern_set_1"
                            },
                            ...
                        ],
                        "WindowsEvents": [
                            {
                                "LogGroupName": "my_windows_event_log_group_1",
                                "EventName": "Application",
                                "EventLevels": [
                                    "ERROR",
                                    "WARNING",
                                    ...
                                ],
                                "Encoding": "utf-8",
                                "PatternSet": "my_pattern_set_2"
                            },
                            ...
                        ],
                        "Alarms": [
                            {
                                "AlarmName": "my_alarm_name",
                                "Severity": "HIGH"
```

```
                        },
                        ...
                    ]
                },
                "SubComponentTypeConfigurations": [
                    {
                        "SubComponentType": "EC2_INSTANCE",
                        "SubComponentConfigurationDetails": {
                            "AlarmMetrics": [
                                {
                                    "AlarmMetricName": "DiskReadOps"
                                },
                                ...
                            ],
                            "Logs": [
                                {
                                    "LogGroupName": "my_log_group_2",
                                    "LogPath": "C:\\LogFolder_2\\*",
                                    "LogType": "IIS",
                                    "Encoding": "utf-8",
                                    "PatternSet": "my_pattern_set_3"
                                },
                                ...
                            ],
                            "processes" : [
                                {
                                    "processName" : "my_process",
                                    "alarmMetrics" : [
                                {
                                    "alarmMetricName" : "procstat cpu_usage",
                                    "monitor" : true
                                }, {
                                    "alarmMetricName" : "procstat memory_rss",
                                    "monitor" : true
                                }
                            ]
                        }
                    ],
                            "WindowsEvents": [
                                {
                                    "LogGroupName": "my_windows_event_log_group_2",
                                    "EventName": "Application",
                                    "EventLevels": [
                                        "ERROR",
```

```
                                        "WARNING",
                                        ...
                                    ],
                                    "Encoding": "utf-8",
                                    "PatternSet": "my_pattern_set_4"
                                },
                                ...
                            ]
                        }
                    }
                ]
            }
        }
    ]
  }
}
```

Plantilla en formato YAML

```
---
Type: AWS::ApplicationInsights::Application
Properties:
  ResourceGroupName: my_resource_group
  ComponentMonitoringSettings:
  - ComponentARN: my_component
    Tier: SQL_SERVER
    ComponentConfigurationMode: CUSTOM
    CustomComponentConfiguration:
      ConfigurationDetails:
        AlarmMetrics:
        - AlarmMetricName: StatusCheckFailed
        ...
        Logs:
        - LogGroupName: my_log_group_1
          LogPath: C:\LogFolder_1\*
          LogType: DOT_NET_CORE
          Encoding: utf-8
          PatternSet: my_pattern_set_1
        ...
        WindowsEvents:
        - LogGroupName: my_windows_event_log_group_1
          EventName: Application
          EventLevels:
```

```
          - ERROR
          - WARNING
          ...
          Encoding: utf-8
          PatternSet: my_pattern_set_2
        ...
        Alarms:
        - AlarmName: my_alarm_name
          Severity: HIGH
        ...
      SubComponentTypeConfigurations:
      - SubComponentType: EC2_INSTANCE
        SubComponentConfigurationDetails:
          AlarmMetrics:
          - AlarmMetricName: DiskReadOps
          ...
          Logs:
          - LogGroupName: my_log_group_2
            LogPath: C:\LogFolder_2\*
            LogType: IIS
            Encoding: utf-8
            PatternSet: my_pattern_set_3
          ...
          Processes:
          - ProcessName: my_process
            AlarmMetrics:
            - AlarmMetricName: procstat cpu_usage
              ...
            ...
          WindowsEvents:
          - LogGroupName: my_windows_event_log_group_2
            EventName: Application
            EventLevels:
            - ERROR
            - WARNING
            ...
            Encoding: utf-8
            PatternSet: my_pattern_set_4
          ...
```

Creación de una aplicación de Información de aplicaciones con la configuración del componente en modo **DEFAULT**

En la siguiente plantilla se realizan las siguientes acciones:

- Crea una aplicación de Información de aplicaciones. Para obtener más información, consulte CreateApplication en la Referencia de la API de Información de aplicaciones de Amazon CloudWatch.
- Componente `my_component` establece `ComponentConfigurationMode` en `DEFAULT` y `Tier` en `SQL_SERVER`, lo que hace que este componente se configure con los ajustes de configuración que Información de aplicaciones recomienda para el nivel `SQL_Server`. Para obtener más información, consulte DescribeComponentConfiguration y UpdateComponentConfiguration en la Referencia de la API de Información de aplicaciones de Amazon CloudWatch.

Plantilla en formato JSON

```
{
    "Type": "AWS::ApplicationInsights::Application",
    "Properties": {
        "ResourceGroupName": "my_resource_group",
        "ComponentMonitoringSettings": [
            {
                "ComponentARN": "my_component",
                "Tier": "SQL_SERVER",
                "ComponentConfigurationMode": "DEFAULT"
            }
        ]
    }
}
```

Plantilla en formato YAML

```
---
Type: AWS::ApplicationInsights::Application
Properties:
  ResourceGroupName: my_resource_group
  ComponentMonitoringSettings:
  - ComponentARN: my_component
    Tier: SQL_SERVER
    ComponentConfigurationMode: DEFAULT
```

Creación de una aplicación de Información de aplicaciones con la configuración del componente en modo **DEFAULT_WITH_OVERWRITE**

En la siguiente plantilla se realizan las siguientes acciones:

- Crea una aplicación de Información de aplicaciones. Para obtener más información, consulte CreateApplication en la Referencia de la API de Información de aplicaciones de Amazon CloudWatch.
- Componente `my_component` establece `ComponentConfigurationMode` en `DEFAULT_WITH_OVERWRITE` y `tier` en `DOT_NET_CORE`, lo que hace que este componente se configure con los ajustes de configuración que Información de aplicaciones recomienda para el nivel `DOT_NET_CORE`. Los ajustes de configuración sobrescritos se especifican en `DefaultOverwriteComponentConfiguration`:
  - En el nivel de componente, la configuración de `AlarmMetrics` se sobrescribe.
  - En el nivel de subcomponente, para los subcomponentes de tipo `EC2_Instance`, los ajustes de `Logs` se sobrescriben.

  Para obtener más información, consulte UpdateComponentConfiguration en la Referencia de la API de Información de aplicaciones de Amazon CloudWatch.

Plantilla en formato JSON

```
{
    "Type": "AWS::ApplicationInsights::Application",
    "Properties": {
        "ResourceGroupName": "my_resource_group",
        "ComponentMonitoringSettings": [
            {
                "ComponentName": "my_component",
                "Tier": "DOT_NET_CORE",
                "ComponentConfigurationMode": "DEFAULT_WITH_OVERWRITE",
                "DefaultOverwriteComponentConfiguration": {
                    "ConfigurationDetails": {
                        "AlarmMetrics": [
                            {
                                "AlarmMetricName": "StatusCheckFailed"
                            }
                        ]
                    },
```

```
                    "SubComponentTypeConfigurations": [
                        {
                            "SubComponentType": "EC2_INSTANCE",
                            "SubComponentConfigurationDetails": {
                                "Logs": [
                                    {
                                        "LogGroupName": "my_log_group",
                                        "LogPath": "C:\\LogFolder\\*",
                                        "LogType": "IIS",
                                        "Encoding": "utf-8",
                                        "PatternSet": "my_pattern_set"
                                    }
                                ]
                            }
                        }
                    ]
                }
            }
        ]
    }
}
```

Plantilla en formato YAML

```
---
Type: AWS::ApplicationInsights::Application
Properties:
  ResourceGroupName: my_resource_group
  ComponentMonitoringSettings:
  - ComponentName: my_component
    Tier: DOT_NET_CORE
    ComponentConfigurationMode: DEFAULT_WITH_OVERWRITE
    DefaultOverwriteComponentConfiguration:
      ConfigurationDetails:
        AlarmMetrics:
        - AlarmMetricName: StatusCheckFailed
      SubComponentTypeConfigurations:
      - SubComponentType: EC2_INSTANCE
        SubComponentConfigurationDetails:
          Logs:
          - LogGroupName: my_log_group
            LogPath: C:\LogFolder\*
            LogType: IIS
```

```
            Encoding: utf-8
            PatternSet: my_pattern_set
```

Tutorial: configuración de la supervisión para SAP ASE

En este tutorial se detalla la configuración de Información de aplicaciones de CloudWatch para establecer la supervisión de sus bases de datos SAP ASE. Puede utilizar los paneles automáticos de Información de aplicaciones de CloudWatch para visualizar los detalles del problema, acelerar la solución de problemas y facilitar el tiempo medio de resolución (TMR) de sus bases de datos SAP ASE.

Información de aplicaciones para los temas de SAP ASE

- Entornos compatibles
- Sistemas operativos compatibles
- Características
- Requisitos previos
- Configurar la supervisión de la base de datos SAP ASE
- Administrar la supervisión de su base de datos SAP ASE
- Configurar el umbral de la alarma
- Visualizar y solucionar los problemas de SAP ASE que detecte Información de aplicaciones
- Solucionar problemas de Información de aplicaciones para SAP ASE

Entornos compatibles

Información de aplicaciones de CloudWatch es compatible con la implementación de recursos de AWS para los siguientes sistemas y patrones. Proporcione e instale el software de base de datos SAP ASE y el software de aplicación SAP compatible.

- Una o más bases de datos SAP ASE en una única instancia de Amazon EC2: SAP ASE en una arquitectura escalable verticalmente de un solo nodo.
- Configuración de alta disponibilidad de bases de datos SAP ASE entre AZ: SAP ASE con alta disponibilidad configurada en dos zonas de disponibilidad mediante clústeres SUSE/RHEL.

> **Note**
>
> Información de aplicaciones de CloudWatch solo admite entornos ASE HA con un identificador único del sistema SAP (SID). Si se adjuntan varios SID de ASE HA, la supervisión se configurará sola para el primer SID detectado.

Sistemas operativos compatibles

Información de aplicaciones de CloudWatch para SAP ASE es compatible con la arquitectura x86-64 en los siguientes sistemas operativos:

- SuSE Linux 12 SP4
- SuSE Linux 12 SP5
- SuSE Linux 15
- SuSE Linux 15 SP1
- SuSE Linux 15 SP2
- SuSE Linux 15 SP3
- SuSE Linux 15 SP4
- SuSE Linux 15 SP1 para SAP
- SuSE Linux 15 SP2 para SAP
- SuSE Linux 15 SP3 para SAP
- SuSE Linux 15 SP4 para SAP
- SuSE Linux 12 SP4 para SAP
- SuSE Linux 12 SP5 para SAP
- RedHat Linux 7.6
- RedHat Linux 7.7
- RedHat Linux 7.9
- RedHat Linux 8.1
- RedHat Linux 8.4
- RedHat Linux 8.6

Características

Información de aplicaciones de CloudWatch para SAP ASE presenta las siguientes características:

- Detección automática de cargas de trabajo de SAP ASE
- Creación automática de alarmas de SAP ASE basada en un umbral estático
- Creación automática de alarmas de SAP ASE basada en la detección de anomalías
- Reconocimiento automático de patrones de registro de SAP ASE
- Panel de estado de SAP ASE
- Panel de problemas de SAP ASE

Requisitos previos

Debe completar los siguientes requisitos previos para configurar una base de datos SAP ASE con Información de aplicaciones de CloudWatch:

- Parámetros de configuración de SAP ASE: los siguientes parámetros de configuración deben estar habilitados en su base de datos ASE:`"enable monitoring"`, `"sql text pipe max messages"`, `"sql text pipe active"`. Esto permite que Información de aplicaciones de CloudWatch proporcione capacidades de supervisión completas para su base de datos. Si esta configuración no está habilitada en su base de datos ASE, Información de aplicaciones permitirá que se recopilen automáticamente las métricas necesarias para permitir la supervisión.
- Usuario de la base de datos SAP ASE: el usuario de la base de datos proporcionado durante la incorporación de Información de aplicaciones debe tener permiso para acceder a lo siguiente:
  - Las tablas del sistema en la base de datos maestra y en las bases de datos de usuarios (inquilinos)
  - Supervisión de las tablas
- SAPHostCtrl: instale y configure SAPHostCtrl en su instancia de Amazon EC2.
- Agente de Amazon CloudWatch: asegúrese de no ejecutar un agente de CloudWatch preexistente en su instancia de Amazon EC2. Si tiene instalado el agente de CloudWatch, asegúrese de eliminar la configuración de los recursos que utiliza en Información de aplicaciones de CloudWatch del archivo de configuración del agente de CloudWatch existente para evitar un conflicto de fusión. Para obtener más información, consulte Cree o edite de forma manual el archivo de configuración del agente de CloudWatch.

- Habilitación de Systems Manager de AWS: instale SSM Agent en sus instancias y habilite las instancias para SSM. Para obtener información acerca de la instalación de SSM Agent, consulte Trabajo con SSM Agent en la Guía del usuario de AWS Systems Manager.
- Roles de instancias de Amazon EC2: debe adjuntar los siguientes roles de instancias de Amazon EC2 para configurar su base de datos.
  - Debe adjuntar el rol de `AmazonSSMManagedInstanceCore` para habilitar Systems Manager. Para obtener más información, consulte Ejemplos de políticas basadas en identidades de AWS Systems Manager.
  - Debe adjuntar la `CloudWatchAgentServerPolicy` (política del servidor del agente de CloudWatch) para permitir que las métricas de instancias y los registros se emitan a través de CloudWatch. Para obtener más información, consulte Creación de roles de IAM y usuarios para utilizarlos con el agente de Amazon CloudWatch.
  - Debe adjuntar la siguiente política inline de IAM al rol de instancia de Amazon EC2 para leer la contraseña almacenada en AWS Secrets Manager. Para obtener más información acerca de las políticas insertadas, consulte Políticas insertadas en la Guía del usuario de AWS Identity and Access Management.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetSecretValue"
            ],
            "Resource": "arn:aws:secretsmanager:*:*:secret:ApplicationInsights-*"
        }
    ]
}
```

- AWS Resource Groups: para incorporar las aplicaciones a Información de aplicaciones de CloudWatch, debe crear un grupo de recursos que incluya todos los recursos asociados de AWS que se utilicen en la pila de aplicaciones. Esto incluye instancias de Amazon EC2 y volúmenes de Amazon EBS en los que se ejecuta la base de datos SAP ASE. Si hay varias bases de datos por cuenta, recomendamos que cree un grupo de recursos que incluya los recursos de AWS para cada sistema de base de datos SAP ASE.
- Permisos de IAM: para usuarios no administradores:

- Debe crear una política de AWS Identity and Access Management (IAM) a fin de que Información de aplicaciones cree un rol vinculado al servicio y este se asocie a la identidad del usuario. Para obtener información sobre los pasos para asociar la política, consultePolítica de IAM.
- El usuario debe tener permiso para crear un secreto en AWS Secrets Manager para almacenar las credenciales de usuario de la base de datos. Para obtener más información, consulte Ejemplo: permiso para crear secretos.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "secretsmanager:CreateSecret"
      ],
      "Resource": "arn:aws:secretsmanager:*:*:secret:ApplicationInsights-*"
    }
  ]
}
```

- Rol vinculado al servicio: Información de aplicaciones utiliza roles vinculados al servicio de AWS Identity and Access Management (IAM). Al crear su primera aplicación de Información de aplicaciones en la consola, Información de aplicaciones le crea el rol vinculado a un servicio. Para obtener más información, consulte Uso de roles vinculados a un servicio para CloudWatch Application Insights.

Configurar la supervisión de la base de datos SAP ASE

Para configurar la supervisión de la base de datos SAP ASE, siga los pasos a continuación:

1. Abra la consola de CloudWatch.
2. Desde el panel de navegación izquierdo, en Insights, elija Application Insights.
3. En la página de Application Insights (Información de aplicaciones) observará la lista de aplicaciones que se supervisen con Información de aplicaciones y el estado de supervisión de cada aplicación. En la esquina superior derecha, elija Add an application (Agregar una aplicación).

4. En la página Especificar detalles de la aplicación, en la lista desplegable que se encuentra debajo de Grupo de recursos, seleccione el grupo de recursos de AWS que contiene los recursos de la base de datos SAP ASE. Si no ha creado un grupo de recursos para la aplicación, puede crear uno si selecciona Create new resource group (Crear nuevo grupo de recursos) debajo del menú desplegable de Resource Groups (grupo de recursos). Para obtener más información acerca de la creación de grupos de recursos, consulte la Guía del usuario de AWS Resource Groups.
5. Debajo de Monitor de Eventos de CloudWatch, seleccione el casillero para integrar la supervisión de Información de aplicaciones con Eventos de CloudWatch y obtener información de Amazon EBS, Amazon EC2, AWS CodeDeploy, Amazon ECS, API y notificaciones de AWS Health, Amazon RDS, Amazon S3 y AWS Step Functions.
6. En Integrate with AWS Systems Manager OpsCenter (Integrar con el OpsCenter de SYSlong), seleccione la casilla de verificación junto a Generate AWS Systems Manager OpsCenter OpsItems for remedial actions (Generar OpsItems de OpsCenter de SYSlong para obtener acciones correctivas) a fin de ver y recibir notificaciones cuando se detecten problemas en las aplicaciones seleccionadas. Para realizar un seguimiento de las operaciones que se realizan para resolver elementos de trabajo operativos, denominados OpsItems, relacionados con sus recursos de AWS, proporcione el ARN del tema de SNS.
7. Puede optar por ingresar etiquetas como ayuda para identificar y organizar sus recursos. Información de aplicaciones de CloudWatch es compatible con los grupos de recursos basados en etiquetas y en AWS CloudFormation, excepto los grupos de Application Auto Scaling. Para obtener más información, consulte Tag Editor (Editor de etiquetas) en la Guía del usuario de etiquetas y AWS Resource Groups.
8. Elija Next (Siguiente) para continuar con la configuración de la supervisión.
9. En la página Revisar componentes detectados se enumeran los componentes supervisados y sus cargas de trabajo detectadas automáticamente por Información de aplicaciones de CloudWatch.

> **Note**
>
> Los componentes que contienen una carga de trabajo detectada de alta disponibilidad de SAP ASE admiten solo una carga de trabajo en cada componente. Los componentes que contienen una carga de trabajo de un solo nodo de SAP ASE detectada admiten varias cargas de trabajo, pero no puede agregar ni eliminar cargas de trabajo. Se supervisarán todas las cargas de trabajo detectadas automáticamente.

10. Elija Siguiente.
11. En la página Especificar detalles de componentes, introduzca el nombre de usuario y la contraseña de sus bases de datos SAP ASE.
12. Revise la configuración de supervisión de aplicaciones y elija Submit (Enviar).
13. Se abrirá la página de detalles de la aplicación, donde podrá ver el Resumen de aplicaciones, la lista de Componentes y cargas de trabajo supervisados y Componentes y cargas de trabajo no supervisados. Si selecciona el botón de opción situado junto a un componente o carga de trabajo, también podrá visualizar el Historial de configuración, los Patrones de registro y cualquiera de las Etiquetas que ha creado. Una vez que envía la configuración, su cuenta implementa todas las métricas y alarmas de su sistema SAP ASE, lo que puede tardar hasta 2 horas.

Administrar la supervisión de su base de datos SAP ASE

Puede administrar las credenciales de usuario, las métricas y las rutas de registro de la base de datos SAP ASE mediante los siguientes pasos:

1. Abra la consola de CloudWatch.
2. Desde el panel de navegación izquierdo, en Insights, elija Application Insights.
3. En la página de Application Insights observará la lista de aplicaciones que se monitorean con Application Insights y el estado de monitoreo de cada aplicación.
4. Debajo de Monitored components (Componentes supervisados), seleccione el botón de radio situado junto al nombre del componente. A continuación, elija Manage monitoring (Administrar supervisión).
5. Debajo de Amazon EC2 instance group logs (Registros de grupos de instancias de Amazon EC2), puede actualizar la ruta de registro existente, el conjunto de patrones de registro y el nombre del grupo de registros. Además, puede agregar hasta tres Application logs (Registros de aplicaciones) adicionales.
6. En Métricas, puede elegir las métricas de SAP ASE según sus requisitos. Los nombres de las métricas de SAP ASE llevan el prefijo `asedb`. Puede agregar hasta 60 métricas por componente.
7. En Configuración de ASE, ingrese el nombre de usuario y la contraseña de la base de datos SAP ASE. Estos son el nombre de usuario y la contraseña que utiliza el agente de Amazon CloudWatch para conectarse a la base de datos SAP ASE.

8. Debajo de Alarmas personalizadas, puede agregar alarmas adicionales para que Información de aplicaciones de CloudWatch las monitoree.
9. Revise la configuración de supervisión de aplicaciones y elija Submit (Enviar). Una vez que envíe la configuración, su cuenta actualizará todas las métricas y alarmas de su sistema SAP HANA, lo que puede tardar hasta 2 horas.

Configurar el umbral de la alarma

Información de aplicaciones de CloudWatch crea una métrica de Amazon CloudWatch de forma automática para que la alarma controle, junto con el umbral para dicha métrica. La alarma pasa al estado `ALARM` cuando la métrica supera el umbral durante un número específico de periodos de evaluación. Tenga en cuenta que Información de aplicaciones no mantiene esta configuración.

Si desea editar una alarma para una sola métrica, siga los pasos a continuación:

1. Abra la consola de CloudWatch.
2. En el panel de navegación izquierdo, elija Alarms (Alarmas) y luego, All alarms (Todas las alarmas).
3. Seleccione el botón de radio situado junto a la alarma que Información de aplicaciones de CloudWatch creó de forma automática. Luego, elija Actions (Acciones) y Edit (Editar) en el menú desplegable.
4. Edite los siguientes parámetros debajo de Metric (Métrica).
    a. Debajo de Statistic (Estadística), elija una de las estadísticas o percentiles predefinidos o especifique un percentil personalizado. Por ejemplo, `p95.45`.
    b. Debajo de Period (Periodo), elija el periodo de evaluación de la alarma. Al evaluar la alarma, se agrega cada uno de los periodos a un punto de datos.
5. Edite los siguientes parámetros de Conditions (Condiciones).
    a. Especifique si la métrica debe ser mayor, menor o igual al umbral.
    b. Especifique el valor del umbral.
6. Debajo de Additional configuration (Configuración adicional), edite los siguientes parámetros.
    a. Debajo de Datapoints to alarm (Puntos de datos para iniciar la alarma), especifique el número de puntos de datos o periodos de evaluación que deben figurar en el estado `ALARM` para iniciar la alarma. Cuando los dos valores coinciden, se crea una alarma que entra

en estado ALARM si se supera el número designado de periodos consecutivos. Para crear una alarma m de n, especifique un valor que sea menor para el primer punto de datos que para el segundo. Para obtener más información acerca de la evaluación de alarmas, consulteEvaluación de alarmas.

b. Debajo de Missing data treatment (Tratamiento de datos faltantes), elija cómo debe comportarse la alarma cuando falten algunos puntos de datos. Para obtener más información acerca del tratamiento de datos faltantes, consulte Configuring how CloudWatch alarms treat missing data.(Configuración del tratamiento de las alarmas de CloudWatch ante datos faltantes).

c. Si la alarma utiliza un percentil como estadística supervisada, aparece un cuadro Percentiles with low samples (Percentiles con pocas muestras). Decida si evaluar o ignorar los casos con frecuencia de muestreo baja. Si elige ignore (maintain alarm state) (ignorar (mantener el estado de alarma)), el estado de alarma actual se mantiene siempre cuando el tamaño de la muestra es demasiado bajo. Para obtener más información acerca de los percentiles con muestreo bajo, consulte Muestras de datos reducidos y alarmas de CloudWatch basadas en percentiles (Percentiles con muestreo bajo).

7. Elija Siguiente.
8. En Notification (Notificación), seleccione el tema de SNS al que desee enviar la notificación cuando la alarma tenga el estado ALARM, OK o INSUFFICIENT_DATA.
9. Elija Update alarm (Actualizar alarma).

Visualizar y solucionar los problemas de SAP ASE que detecte Información de aplicaciones

Esta sección detalla cómo resolver problemas que ocurren cuando configura la supervisión de SAP ASE en Información de aplicaciones.

Errores del servidor de copia de seguridad de SAP ASE

Puede identificar el mensaje de error consultando el panel creado dinámicamente. El panel muestra el mensaje de error registrado en el servidor de copia de seguridad SAP ASE. Para obtener más información sobre los registros del servidor de copia de seguridad de SAP ASE, consulte la documentación de registro de error del servidor de copia de seguridad de SAP.

Transacciones de SAP ASE de larga duración

Identifique la transacción de larga duración y confirme si se puede detener o si el tiempo de ejecución es intencional. Para obtener más información, consulte 2180410: ¿cómo mostrar los registros del registro de transacciones para las transacciones de larga duración? — SAP ASE.

Conexiones de usuario de SAP ASE

Compruebe si su base de datos SAP ASE tiene el tamaño adecuado para la carga de trabajo que pretende ejecutar en la base de datos. Para obtener más información, consulte Configuración de las conexiones de usuario en la documentación de SAP.

Espacio en disco de SAP ASE

Puede identificar la capa de la base de datos que está causando el problema si observa el panel creado de manera dinámica. El panel muestra las métricas relacionadas y los fragmentos de los archivos de registro. Es importante entender la causa del crecimiento del disco y, cuando proceda, aumentar el tamaño del disco físico, el espacio en disco asignado o ambos. Para obtener más información, consulte la Documentación de SAP sobre el cambio de tamaño del disco en la documentación de SAP.

Solucionar problemas de Información de aplicaciones para SAP ASE

En esta sección se detallan los pasos a seguir para resolver los errores comunes que se presentan en el panel de Información de aplicaciones.

Error	Error devuelto	Causa raíz	Resolución
No se pudieron agregar más de 60 métricas de supervisión.	`Component cannot have more than 60 monitored metric`	El límite de métricas actual es de 60 métricas supervisadas por componente.	Elimine las métricas innecesarias para cumplir con el límite.
No aparecen métricas ni alarmas de SAP después del proceso de incorporación	El comando `run` del `AWS-ConfigureAWSPackage` devolvió un error en AWS Systems Manager. En el resultado, se observa el siguiente error:	Es posible que el nombre de usuario y la contraseña sean incorrectos.	Compruebe que el nombre de usuario y la contraseña sean válidos y vuelva a ejecutar el proceso de incorporación.

Error	Error devuelto	Causa raíz	Resolución
	`CT-LIBRARY error:ct_connect(): protocol specific layer: external error: The attempt to connect to the server failed`		

Tutorial: Configuración de la supervisión para SAP HANA

En este tutorial se detalla la configuración de Información de aplicaciones de CloudWatch para establecer la supervisión de sus bases de datos SAP HANA. Puede utilizar los paneles automáticos de Información de aplicaciones de CloudWatch para visualizar los detalles del problema, acelerar la solución de problemas y facilitar el tiempo medio de resolución (TMR) de sus bases de datos SAP HANA.

Información de aplicaciones para los temas de SAP HANA

- Entornos compatibles
- Sistemas operativos compatibles
- Características
- Requisitos previos
- Configuración de la base de datos SAP HANA para la supervisión
- Administración de la supervisión de su base de datos SAP HANA
- Visualización y solución de los problemas que detecte Información de aplicaciones de CloudWatch
- Detección de anomalías de SAP HANA
- Solución de problemas de Información de aplicaciones para SAP HANA

Entornos compatibles

Información de aplicaciones de CloudWatch es compatible con la implementación de recursos de AWS para los siguientes sistemas y patrones. Proporcione e instale el software de base de datos SAP HANA y el software de aplicación SAP compatible.

- Base de datos SAP HANA en una única instancia de Amazon EC2: SAP HANA en una arquitectura escalable de un solo nodo, con un máximo de 24 TB de memoria.
- Base de datos SAP HANA en varias instancias de Amazon EC2: SAP HANA en una arquitectura multinodo con escalado horizontal.
- Configuración de alta disponibilidad de bases de datos SAP HANA entre AZ: SAP HANA con alta disponibilidad configurada en dos zonas de disponibilidad mediante clústeres SUSE/RHEL.

> **Note**
>
> Información de aplicaciones de CloudWatch solo admite entornos SID HANA únicos. Si se adjuntan varios SID HANA, la supervisión se configurará solo para el primer SID detectado.

Sistemas operativos compatibles

Información de aplicaciones de CloudWatch para SAP HANA es compatible con la arquitectura x86-64 en los siguientes sistemas operativos:

- SuSE Linux 12 SP4 para SAP
- SuSE Linux 12 SP5 para SAP
- SuSE Linux 15
- SuSE Linux 15 SP1
- SuSE Linux 15 SP2
- SuSE Linux 15 para SAP
- SuSE Linux 15 SP1 para SAP
- SuSE Linux 15 SP2 para SAP
- SuSE Linux 15 SP3 para SAP
- SuSE Linux 15 SP4 para SAP
- SuSE Linux 15 SP5 para SAP
- RedHat Linux 8.6 para SAP con servicios de alta disponibilidad y actualización
- RedHat Linux 8.5 para SAP con servicios de alta disponibilidad y actualización
- RedHat Linux 8.4 para SAP con servicios de alta disponibilidad y actualización
- RedHat Linux 8.3 para SAP con servicios de alta disponibilidad y actualización

- RedHat Linux 8.2 para SAP con servicios de alta disponibilidad y actualización
- RedHat Linux 8.1 para SAP con servicios de alta disponibilidad y actualización
- RedHat Linux 7.9 para SAP con servicios de alta disponibilidad y actualización

Características

Información de aplicaciones de CloudWatch para SAP HANA presenta las siguientes características:

- detección automática de cargas de trabajo SAP HANA
- creación automática de alarmas SAP HANA basada en un umbral estático
- creación automática de alarmas SAP HANA basada en la detección de anomalías
- reconocimiento automático de patrones de registro de SAP HANA
- Panel de Health de SAP HANA
- Panel de problemas de SAP HANA

Requisitos previos

Debe completar los siguientes requisitos previos para configurar una base de datos SAP HANA con Información de aplicaciones de CloudWatch:

- SAP HANA: instale una base de datos 2.0 SPS05 SAP HANA en ejecución y accesible en una instancia de Amazon EC2.
- Usuario de base de datos SAP HANA: se debe crear un usuario de base de datos con roles de supervisión en la base de datos SYSTEM y en todos los inquilinos.

  Ejemplo

  Los siguientes comandos SQL crean un usuario con roles de supervisión.

```
su - <sid>adm
hdbsql -u SYSTEM -p <SYSTEMDB password> -d SYSTEMDB
CREATE USER CW_HANADB_EXPORTER_USER PASSWORD <Monitoring user password> NO
 FORCE_FIRST_PASSWORD_CHANGE;
CREATE ROLE CW_HANADB_EXPORTER_ROLE;
GRANT MONITORING TO CW_HANADB_EXPORTER_ROLE;
GRANT CW_HANADB_EXPORTER_ROLE TO CW_HANADB_EXPORTER_USER;
```

- Python 3.8: instale Python 3.8 o versiones posteriores en su sistema operativo. Utilice la última versión de Python. Si Python3 no se detecta en el sistema operativo, se instalará Python 3.6.

  Para obtener más información, consulte installation example.

  > **Note**
  >
  > Se requiere la instalación manual de Python 3.8 o superior para los sistemas operativos SuSE Linux 15 SP4, RedHat Linux 8.6 y versiones posteriores.

- Pip3: instale el programa instalador, pip3, en su sistema operativo. Si no se detecta pip3 en el sistema operativo, se instalará.
- hdbclient: Información de aplicaciones de CloudWatch utiliza el controlador Python para conectarse a la base de datos SAP HANA. Si el cliente no está instalado en python3, asegúrese de tener la versión `2.10 or later` del archivo tar de hdbclient en `/hana/shared/SID/hdbclient/`.
- Agente de Amazon CloudWatch: asegúrese de no ejecutar un agente de CloudWatch preexistente en su instancia de Amazon EC2. Si tiene instalado el agente de CloudWatch, asegúrese de eliminar la configuración de los recursos que utiliza en Información de aplicaciones de CloudWatch del archivo de configuración del agente de CloudWatch existente para evitar un conflicto de fusión. Para obtener más información, consulte Cree o edite de forma manual el archivo de configuración del agente de CloudWatch.
- Habilitación de AWS Systems Manager: instale SSM Agent en sus instancias y las instancias deben estar habilitadas para SSM. Para obtener información acerca de la instalación de SSM Agent, consulte Uso de SSM Agent en la Guía del usuario de AWS Systems Manager.
- Roles de instancias de Amazon EC2: debe adjuntar los siguientes roles de instancias de Amazon EC2 para configurar su base de datos.
  - Debe adjuntar el rol de `AmazonSSMManagedInstanceCore` para habilitar Systems Manager. Para obtener más información, consulte Ejemplos de políticas basadas en identidades de AWS Systems Manager.
  - Debe adjuntar la `CloudWatchAgentServerPolicy` (política del servidor del agente de CloudWatch) para permitir que las métricas de instancias y los registros se emitan a través de CloudWatch. Consulte Creación de roles de IAM y usuarios para utilizarlos con el agente de CloudWatch para obtener más información.
  - Debe adjuntar la siguiente política inline de IAM al rol de instancia de Amazon EC2 para leer la contraseña almacenada en AWS Secrets Manager. Para obtener más información acerca de las

políticas insertadas, consulte Políticas insertadas en la Guía del usuario de AWS Identity and Access Management.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetSecretValue"
            ],
            "Resource": "arn:aws:secretsmanager:*:*:secret:ApplicationInsights-*"
        }
    ]
}
```

- Grupos de recursos de AWS: para incorporar las aplicaciones a Información de aplicaciones de CloudWatch, cree un grupo de recursos que incluya todos los recursos de AWS asociados que se utilicen en la pila de aplicaciones. Esto incluye instancias de Amazon EC2 y volúmenes de Amazon EBS en los que se ejecuta la base de datos SAP HANA. Si hay varias bases de datos por cuenta, le recomendamos que cree un grupo de recursos que incluya los recursos de AWS para cada sistema de base de datos SAP HANA.
- Permisos de IAM: para usuarios no administradores:
  - Debe crear una política de AWS Identity and Access Management (IAM) a fin de que Información de aplicaciones cree un rol vinculado al servicio y este se asocie a la identidad del usuario. Para obtener información sobre los pasos para asociar la política, consultePolítica de IAM.
  - El usuario debe tener permiso para crear un secreto en AWS Secrets Manager para almacenar las credenciales de usuario de la base de datos. Para obtener más información, consulte Ejemplo: permiso para crear secretos.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "secretsmanager:CreateSecret"
      ],
```

```
      "Resource": "arn:aws:secretsmanager:*:*:secret:ApplicationInsights-*"
    }
  ]
}
```

- Rol vinculado al servicio: Información de aplicaciones utiliza roles vinculados al servicio de AWS Identity and Access Management (IAM). Al crear su primera aplicación de Información de aplicaciones en la consola, Información de aplicaciones le crea el rol vinculado a un servicio. Para obtener más información, consulte Uso de roles vinculados a un servicio para CloudWatch Application Insights.

Configuración de la base de datos SAP HANA para la supervisión

Para configurar la supervisión de la base de datos SAP HANA, siga los pasos a continuación

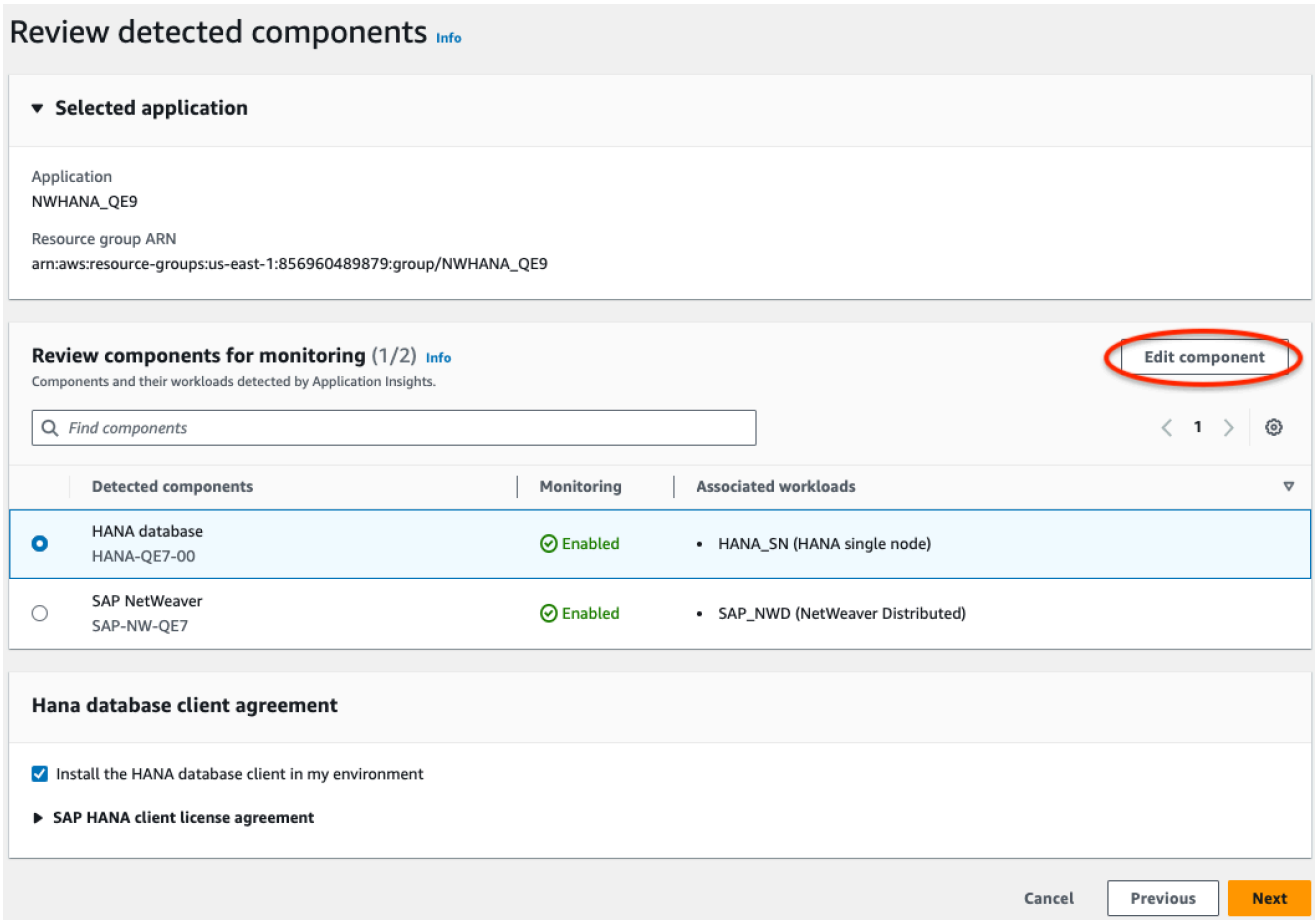
1. Abra la consola de CloudWatch.
2. Desde el panel de navegación izquierdo, en Insights, elija Application Insights.
3. En la página de Application Insights (Información de aplicaciones) observará la lista de aplicaciones que se supervisen con Información de aplicaciones y el estado de supervisión de cada aplicación. En la esquina superior derecha, elija Add an application (Agregar una aplicación).
4. En la página Specify application details (Especificar detalles de la aplicación), en la lista desplegable que se encuentra debajo de Resource Groups (Grupo de recursos), seleccione el grupo de recursos de AWS que contiene los recursos de la base de datos SAP HANA. Si no ha creado un grupo de recursos para la aplicación, puede crear uno si selecciona Create new resource group (Crear nuevo grupo de recursos) debajo del menú desplegable de Resource Groups (grupo de recursos). Para obtener más información acerca de la creación de grupos de recursos, consulte la Guía del usuario de AWS Resource Groups.
5. Debajo de Monitor de Eventos de CloudWatch, seleccione el casillero para integrar la supervisión de Información de aplicaciones con Eventos de CloudWatch y obtener información de Amazon EBS, Amazon EC2, AWS CodeDeploy, Amazon ECS, API y notificaciones de AWS Health, Amazon RDS, Amazon S3 y AWS Step Functions.
6. En Integrate with AWS Systems Manager OpsCenter (Integrar con el OpsCenter de SYSlong), seleccione la casilla de verificación junto a Generate AWS Systems Manager OpsCenter OpsItems for remedial actions (Generar OpsItems de OpsCenter de SYSlong para obtener acciones correctivas) a fin de ver y recibir notificaciones cuando se detecten problemas en las aplicaciones seleccionadas. Para realizar un seguimiento de las operaciones que se realizan

para resolver elementos de trabajo operativos, denominados OpsItems, relacionados con sus recursos de AWS, proporcione el ARN del tema de SNS.

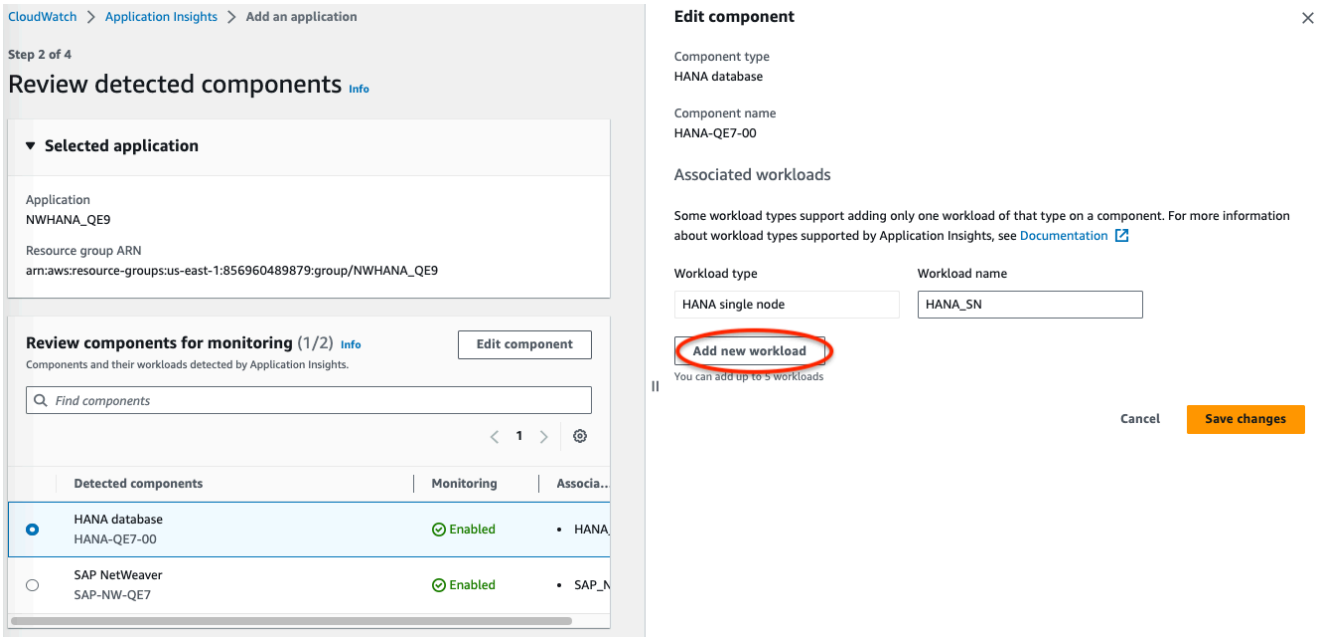
7. Puede optar por ingresar etiquetas como ayuda para identificar y organizar sus recursos. Información de aplicaciones de CloudWatch es compatible con los grupos de recursos basados en etiquetas y en AWS CloudFormation, excepto los grupos de Application Auto Scaling. Para obtener más información, consulte Tag Editor (Editor de etiquetas) en la Guía del usuario de etiquetas y AWS Resource Groups.
8. Elija Next (Siguiente) para continuar con la configuración de la supervisión.
9. En la página Revisar componentes detectados se enumeran los componentes supervisados y sus cargas de trabajo detectadas automáticamente por Información de aplicaciones de CloudWatch.

    a. Para añadir cargas de trabajo a un componente que contiene una carga de trabajo de nodo único de SAP HANA detectada, seleccione el componente y, a continuación, elija Editar componente.

    > **Note**
    >
    > Los componentes que contienen una carga de trabajo detectada de múltiples nodos o de alta disponibilidad de HANA de SAP HANA admiten solo una carga de trabajo en cada componente.



b. Para añadir una nueva carga de trabajo, seleccione Añadir nueva carga de trabajo.



c. Cuando haya terminado de editar cargas de trabajo, elija Guardar cambios.

10. Elija Siguiente.
11. En la página Especificar detalles de componentes, introduzca el nombre y la contraseña del usuario.
12. Revise la configuración de supervisión de aplicaciones y elija Submit (Enviar).
13. Se abrirá la página de detalles de la aplicación, donde podrá ver el Resumen de aplicaciones, la lista de Componentes y cargas de trabajo supervisados y Componentes y cargas de trabajo no supervisados. Si selecciona el botón de opción situado junto a un componente o carga de trabajo, también podrá visualizar el Historial de configuración, los Patrones de registro y cualquiera de las Etiquetas que ha creado. Una vez que envía la configuración, su cuenta implementa todas las métricas y alarmas de su sistema SAP HANA, lo que puede tardar hasta 2 horas.

Administración de la supervisión de su base de datos SAP HANA

Puede administrar las credenciales de usuario, las métricas y las rutas de registro de la base de datos SAP HANA si sigue los siguientes pasos:

1. Abra la consola de CloudWatch.
2. Desde el panel de navegación izquierdo, en Insights, elija Application Insights.
3. En la página de Application Insights observará la lista de aplicaciones que se monitorean con Application Insights y el estado de monitoreo de cada aplicación.
4. Debajo de Monitored components (Componentes supervisados), seleccione el botón de radio situado junto al nombre del componente. A continuación, elija Manage monitoring (Administrar supervisión).
5. Debajo de Amazon EC2 instance group logs (Registros de grupos de instancias de Amazon EC2), puede actualizar la ruta de registro existente, el conjunto de patrones de registro y el nombre del grupo de registros. Además, puede agregar hasta tres Application logs (Registros de aplicaciones) adicionales.
6. Debajo de Metrics (Métricas), puede elegir las métricas de SAP HANA según sus requisitos. Los nombres de métricas de SAP HANA llevan el prefijo `hanadb`. Puede agregar hasta 40 métricas por componente.
7. Debajo de HANA Configuration (Configuración de HANA), ingrese el nombre de usuario y la contraseña de la base de datos SAP HANA. Deberá ingresar el nombre de usuario y la

contraseña que utiliza el agente de Amazon CloudWatch para conectarse a la base de datos SAP HANA.

8. Debajo de Alarmas personalizadas, puede agregar alarmas adicionales para que Información de aplicaciones de CloudWatch las supervise.
9. Revise la configuración de supervisión de aplicaciones y elija Submit (Enviar). Una vez que envíe la configuración, su cuenta actualizará todas las métricas y alarmas de su sistema SAP HANA, lo que puede tardar hasta 2 horas.

Visualización y solución de los problemas que detecte Información de aplicaciones de CloudWatch

En las siguientes secciones se detallan los pasos a seguir para resolver escenarios comunes de solución de problemas que se llevan a cabo al configurar el supervisión de SAP HANA en Información de aplicaciones.

Temas de solución de problemas

- Base de datos SAP HANA con límite de asignación
- Evento de disco lleno
- La copia de seguridad de SAP HANA se detuvo

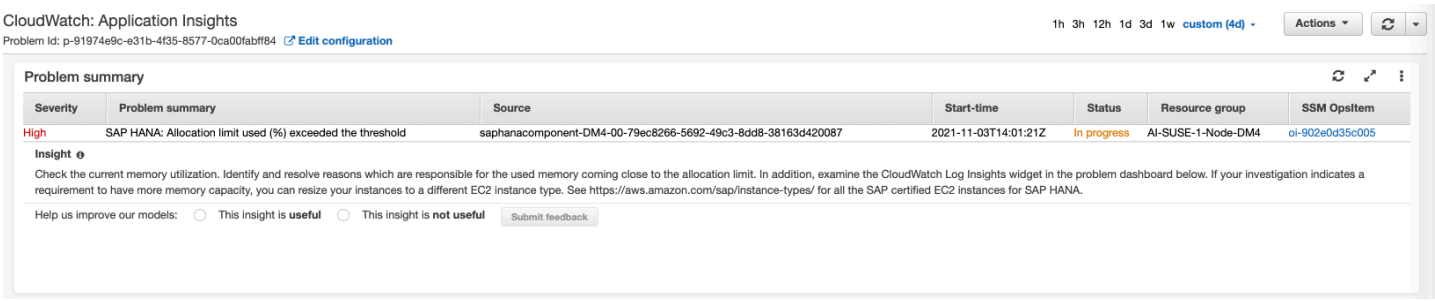
Base de datos SAP HANA con límite de asignación

Descripción

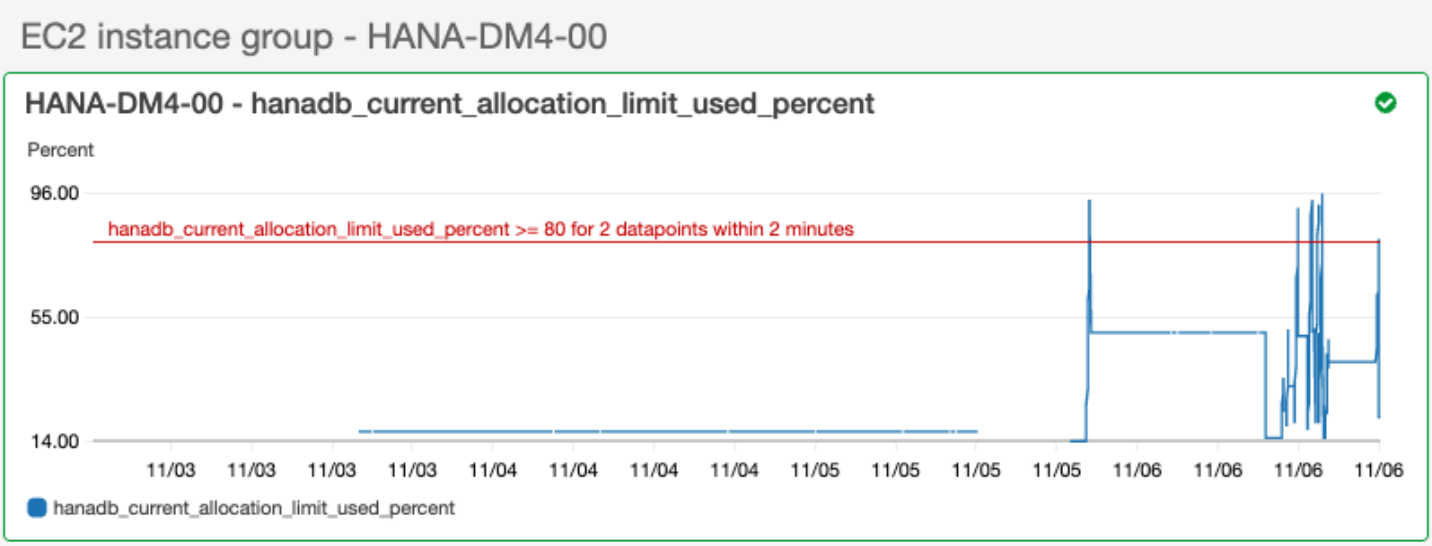
Su aplicación SAP respaldada por una base de datos SAP HANA presenta fallas de funcionamiento debido a la alta presión de la memoria, lo que provoca un deterioro del rendimiento de las aplicaciones.

Resolución

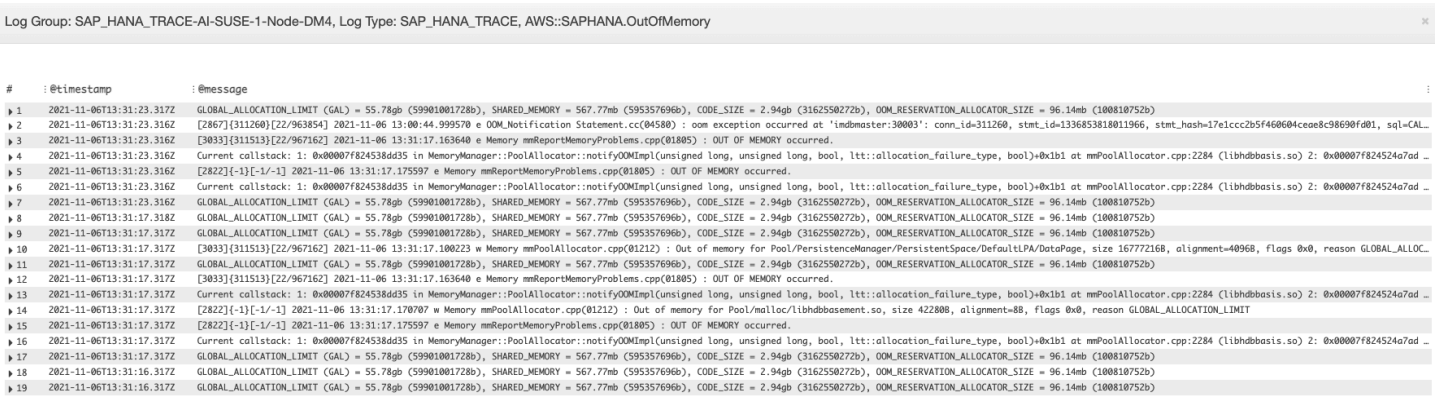
Puede identificar la capa de la aplicación que está causando el problema si observa el panel creado de manera dinámica que muestra las métricas relacionadas y fragmentos de los archivos de registro. En el siguiente ejemplo, el problema puede deberse a una gran carga de datos en el sistema SAP HANA.



La asignación de memoria utilizada supera el umbral del 80 por ciento del límite total de asignación de memoria.



En el grupo de registros se visualiza el esquema `BNR-DATA` y la tabla `IMDBMASTER_30003` que se ha quedado sin memoria. Además, en el grupo de registros se visualiza la hora exacta del problema, el límite de ubicación global actual, la memoria compartida, el tamaño del código y el tamaño de asignación de reservas OOM.



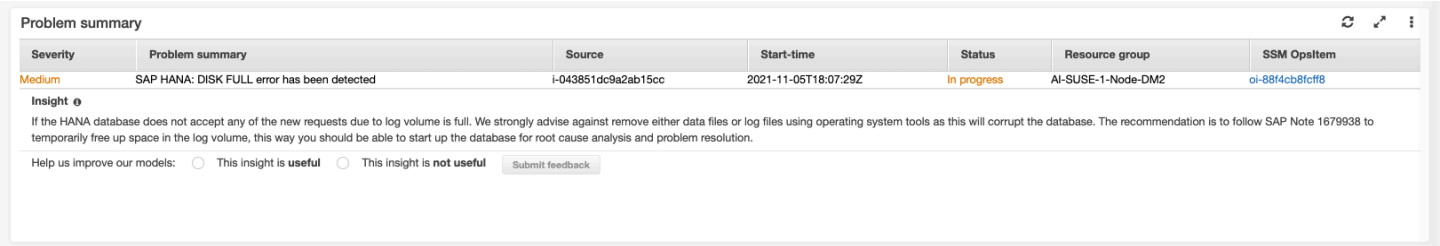
Evento de disco lleno

Descripción

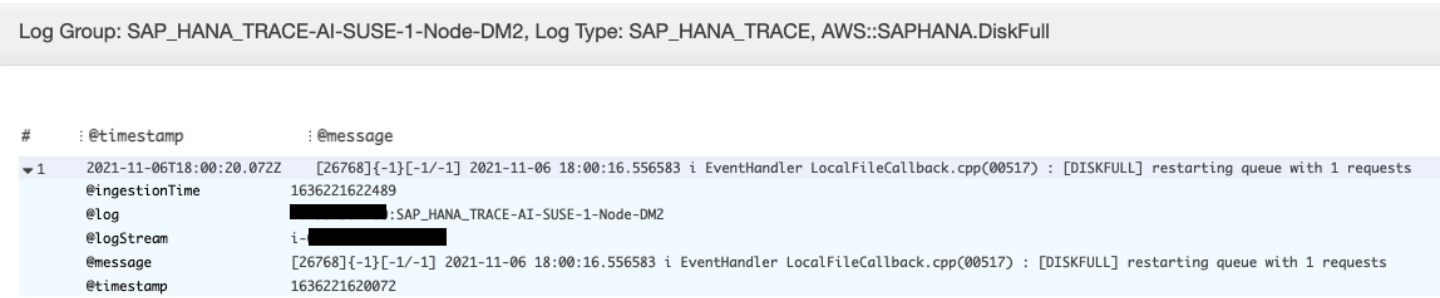
La aplicación SAP respaldada por una base de datos SAP HANA deja de responder, lo que provoca la imposibilidad de acceder a la base de datos.

Resolución

Puede identificar la capa de la base de datos que está causando el problema si observa el panel creado de manera dinámica que muestra las métricas relacionadas y fragmentos de los archivos de registro. En el siguiente ejemplo, el problema puede ser que el administrador no pudo habilitar la copia de seguridad automática de registros, lo que provocó que se llenara el directorio sap/hana/log.



En el widget de grupo de registros en el panel de problemas se visualiza el evento `DISKFULL` (disco lleno).



La copia de seguridad de SAP HANA se detuvo

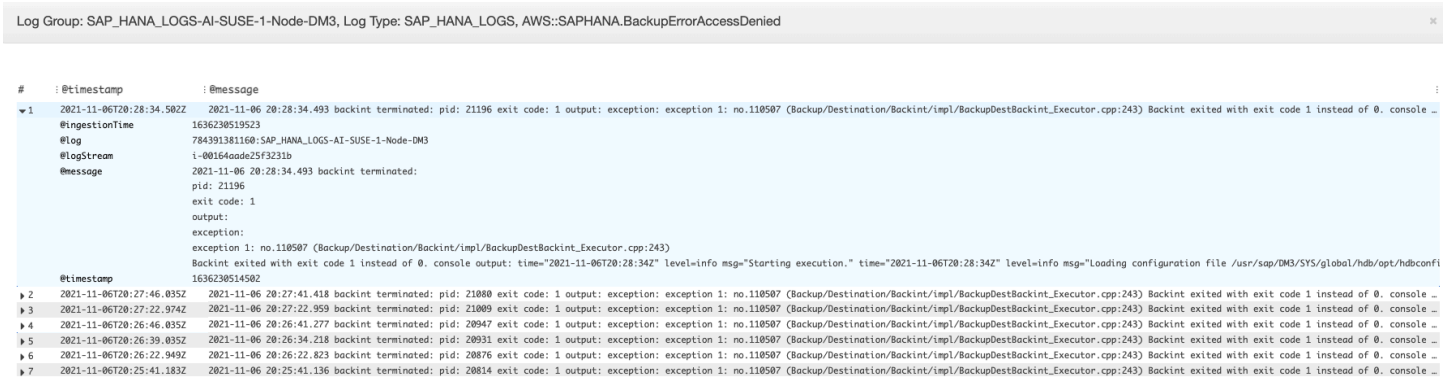
Descripción

La aplicación SAP respaldada por una base de datos SAP HANA dejó de funcionar.

Resolución

Puede identificar la capa de la base de datos que está causando el problema si observa el panel creado de manera dinámica que muestra las métricas relacionadas y fragmentos de los archivos de registro.

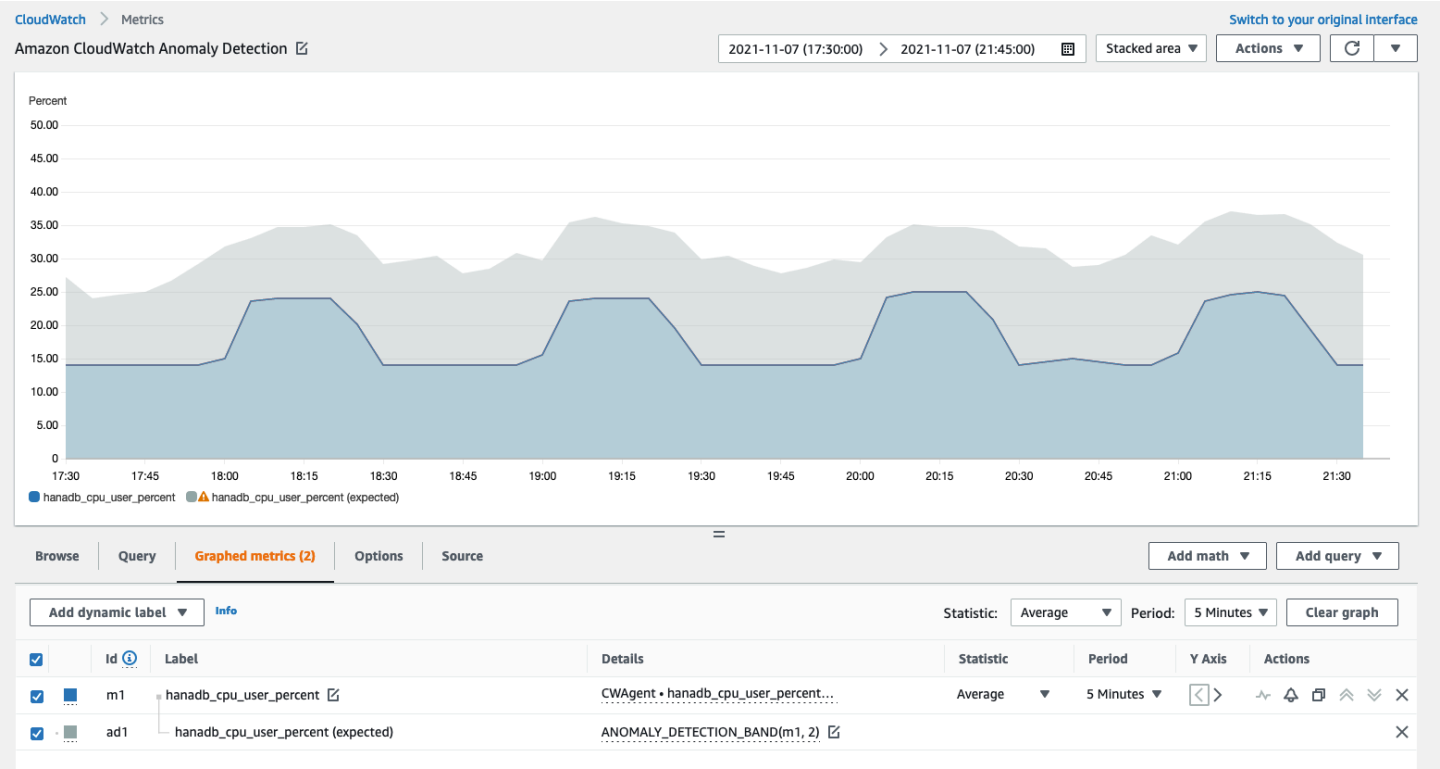
En el widget de grupo de registros en el panel de problemas se visualiza el evento `ACCESS DENIED` (acceso denegado). Esto incluye información adicional, como el bucket de Amazon S3, la carpeta del bucket de Amazon S3 y la región del bucket de Amazon S3.



Detección de anomalías de SAP HANA

En el caso de métricas específicas de SAP HANA, como el número de recuentos de procesos, CloudWatch utiliza algoritmos estadísticos y de machine learning para definir el umbral. Con estos algoritmos se analizan las métricas de la base de datos SAP HANA, se determinan los valores de referencia normales y se detectan anomalías con una intervención mínima del usuario de forma continua. Los algoritmos generan un modelo de detección de anomalías, lo que genera un intervalo de valores esperados que representan el comportamiento normal de la métrica.

Los algoritmos de detección de anomalías dan cuenta de la estacionalidad y los cambios de tendencia de las métricas. Los cambios de estacionalidad pueden ser por hora, por día o por semana, como se muestra en los siguientes ejemplos del uso de la CPU de SAP HANA.



Después de crear un modelo, la detección de anomalías de CloudWatch evalúa el modelo y realiza ajustes de forma continua para garantizar que sea lo más preciso posible. Esto incluye volver a entrenar el modelo para ajustarlo si los valores de las métricas evolucionan con el tiempo o experimentan cambios repentinos. También incluye indicadores para mejorar los modelos de métricas estacionales, con picos o estables.

Solución de problemas de Información de aplicaciones para SAP HANA

En esta sección se detallan los pasos a seguir para resolver los errores comunes que se presentan en el panel de Información de aplicaciones.

No se pueden agregar más de 60 métricas supervisadas

En el resultado, se observa el siguiente error.

```
Component cannot have more than 60 monitored metrics
```

Causa raíz: el límite de métricas actual es de 60 métricas supervisadas por componente.

Resolución: para mantenerse por debajo del límite, elimine las métricas que no sean necesarias.

No aparecen métricas de SAP después del proceso de incorporación

Utilice la siguiente información para descubrir por qué las métricas de SAP no aparecen en el panel después del proceso de incorporación. El primer paso es solucionar el problema por el que las métricas de SAP no aparecen utilizando los registros de la AWS Management Console o del exportador de una instancia de Amazon EC2. A continuación, revise el resultado del error para encontrar una solución.

Solución del problema por el que las métricas de SAP no aparecen después de la incorporación

Puede utilizar los registros de la AWS Management Console o los registros del exportador de una instancia de Amazon EC2 para solucionar problemas.

AWS Management Console

Solución del problema por el que las métricas de SAP no aparecen después de la incorporación usando la consola

1. Abra la consola de AWS Systems Manager en https://console.aws.amazon.com/systems-manager/.
2. En el panel de navegación izquierdo, elija Administrador de estado.
3. En Asociaciones, compruebe el estado del documento `AWSEC2-ApplicationInsightsCloudwatchAgentInstallAndConfigure`. Si el estado es `Failed`, en ID de ejecución, seleccione el identificador fallido y visualice el resultado.
4. En Asociaciones, compruebe el estado del documento `AWS-ConfigureAWSPackage`. Si el estado es `Failed`, en ID de ejecución, seleccione el identificador fallido y visualice el resultado.

Exporter logs from Amazon EC2 instance

Solución del problema por el que las métricas de SAP no aparecen después de la incorporación utilizando los registros del exportador

1. Conecte la instancia de Amazon EC2 en la que se ejecuta su base de datos SAP HANA.
2. Encuentre la convención de nomenclatura correcta de `WORKLOAD_SHORT_NAME` mediante el siguiente comando. Utilizará este nombre corto en los dos pasos siguientes.

```
sudo systemctl | grep exporter
```

> **Note**
>
> Información de aplicaciones agrega el sufijo `WORKLOAD_SHORT_NAME` al nombre del servicio en función de la carga de trabajo que se esté ejecutando. Los nombres abreviados de las implementaciones de nodo único, múltiples nodos y alta disponibilidad de SAP HANA son `HANA_SN`, `HANA_MN` y `HANA_HA`.

3. Para verificar si hay errores en los registros del servicio del administrador del exportador, ejecute el siguiente comando reemplazando `WORKLOAD_SHORT_NAME` por el nombre corto que encontró en Step 2.

```
sudo journalctl -e --unit=prometheus-
hanadb_exporter_manager_WORKLOAD_SHORT_NAME.service
```

4. Si los registros del servicio del administrador del exportador no muestran ningún error, ejecute el siguiente comando para comprobar si hay errores en los registros del servicio del exportador.

```
sudo journalctl -e --unit=prometheus-hanadb_exporter_WORKLOAD_SHORT_NAME.service
```

Resolución de las causas principales más comunes por las que las métricas de SAP no aparecen después de la incorporación

Los siguientes ejemplos describen cómo resolver las causas principales comunes por las que las métricas de SAP no aparecen después de la incorporación.

- En el resultado, se observa el siguiente error.

```
Reading json config file path: /opt/aws/amazon-cloudwatch-agent/etc/amazon-
cloudwatch-agent.d/default ...
Reading json config file path: /opt/aws/amazon-cloudwatch-agent/etc/
amazon-cloudwatch-agent.d/ssm_AmazonCloudWatch-ApplicationInsights-
SSMParameterForTESTCWEC2INSTANCEi0d88867f1f3e36285.tmp ...
2023/11/30 22:25:17 Failed to merge multiple json config files.
2023/11/30 22:25:17 Failed to merge multiple json config files.
2023/11/30 22:25:17 Under path : /metrics/append_dimensions | Error : Different
 values are specified for append_dimensions
2023/11/30 22:25:17 Under path : /metrics/metrics_collected/disk | Error : Different
 values are specified for disk
```

```
2023/11/30 22:25:17 Under path : /metrics/metrics_collected/mem | Error : Different
 values are specified for mem
2023/11/30 22:25:17 Configuration validation first phase failed. Agent version: 1.0.
 Verify the JSON input is only using features supported by this version.
```

Resolución: Información de aplicaciones está intentando configurar las mismas métricas que están preconfiguradas como parte del archivo de configuración del agente de CloudWatch existente. Elimine los archivos existentes en `/opt/aws/amazon-cloudwatch-agent/etc/amazon-cloudwatch-agent.d/` o elimine las métricas que están causando el conflicto del archivo de configuración del agente de CloudWatch existente.

- En el resultado, se observa el siguiente error.

```
Unable to find a host with system database, for more info rerun using -v
```

Resolución: es posible que el nombre de usuario, la contraseña o el puerto de la base de datos sean incorrectos. Compruebe que el nombre de usuario, la contraseña y el puerto sean válidos y luego vuelva a ejecutar el proceso de incorporación.

- En el resultado, se observa el siguiente error.

```
This hdbcli installer is not compatible with your Python interpreter
```

Resolución: actualice pip3 y wheel como se muestra en el siguiente ejemplo para Python 3.6.

```
python3.6 -m pip install --upgrade pip setuptools wheel
```

- En el resultado, se observa el siguiente error.

```
Unable to install hdbcli using pip3. Please try to install it
```

Resolución: asegúrese de cumplir con los requisitos previos de `hdbclient` o instale `hdbclient` manualmente en pip3.

- En el resultado, se observa el siguiente error.

```
Package 'boto3' requires a different Python: 3.6.15 not in '>= 3.7'
```

Resolución: se requiere Python 3.8 o superior para esta versión del sistema operativo. Compruebe los requisitos previos de Python 3.8 e instálelo.

- En el resultado se observa uno de los siguientes errores de instalación.

```
Can not execute `setup.py` since setuptools is not available in the build environment
```

o

```
[SSL: CERTIFICATE_VERIFY_FAILED]
```

Resolución: instale Python utilizando los comandos de SUSE Linux como se muestra en el siguiente ejemplo. El ejemplo siguiente instala la versión más reciente de Python 3.8.

```
wget https://www.python.org/ftp/python/3.8.<LATEST_RELEASE>/
Python-3.8.<LATEST_RELEASE>.tgz
tar xf Python-3.*
cd Python-3.*/
sudo zypper install make gcc-c++ gcc automake autoconf libtool
sudo zypper install zlib-devel
sudo zypper install libopenssl-devel libffi-devel
./configure --with-ensurepip=install
sudo make
sudo make install
sudo su
python3.8 -m pip install --upgrade pip setuptools wheel
```

Tutorial: Configuración de la supervisión para SAP NetWeaver

En este tutorial se detalla la configuración de Información de aplicaciones de Amazon CloudWatch para establecer la supervisión de sus bases de datos SAP NetWeaver. Puede usar los paneles automáticos de Información de aplicaciones de CloudWatch para visualizar los detalles del problema, acelerar la solución de problemas y facilitar el tiempo medio de resolución (TMR) de los servidores de aplicación de SAP NetWeaver.

Información de aplicaciones de CloudWatch para los temas de SAP NetWeaver

- Entornos compatibles
- Sistemas operativos compatibles
- Características

- Requisitos previos
- Configuración de los servidores de aplicaciones de SAP NetWeaver para la supervisión
- Administración de la supervisión de los servidores de aplicaciones de SAP NetWeaver
- Visualización y solución de los problemas de SAP NetWeaver que detecte Información de aplicaciones de CloudWatch
- Solución de problemas de Información de aplicaciones para SAP NetWeaver

Entornos compatibles

Información de aplicaciones de CloudWatch es compatible con la implementación de recursos de AWS para los siguientes sistemas y patrones.

- Implementación del sistema SAP NetWeaver Standard.
- Implementaciones distribuidas de SAP NetWeaver en varias instancias de Amazon EC2.
- Configuración de alta disponibilidad de SAP NetWeaver entre zonas de disponibilidad: SAP NetWeaver con alta disponibilidad configurada entre dos zonas de disponibilidad mediante clústeres SUSE/RHEL.

Sistemas operativos compatibles

Información de aplicaciones de CloudWatch para SAP NetWeaver es compatible con los siguientes sistemas operativos:

- Oracle Linux 8
- Red Hat Enterprise Linux 7.6
- Red Hat Enterprise Linux 7.7
- Red Hat Enterprise Linux 7.9
- Red Hat Enterprise Linux 8.1
- Red Hat Enterprise Linux 8.2
- Red Hat Enterprise Linux 8.4
- Red Hat Enterprise Linux 8.6
- SUSE Linux Enterprise Server 15 para SAP
- SUSE Linux Enterprise Server 15 SP1 para SAP

- SUSE Linux Enterprise Server 15 SP2 para SAP
- SUSE Linux Enterprise Server 15 SP3 para SAP
- SUSE Linux Enterprise Server 15 SP4 para SAP
- SUSE Linux Enterprise Server 12 SP4 para SAP
- SUSE Linux Enterprise Server 12 SP5 para SAP
- SUSE Enterprise Server 15, excepto patrones de alta disponibilidad
- SUSE Enterprise Server 15 SP1, excepto patrones de alta disponibilidad
- SUSE Enterprise Server 15 SP2, excepto patrones de alta disponibilidad
- SUSE Enterprise Server 15 SP3, excepto patrones de alta disponibilidad
- SUSE Enterprise Server 15 SP4, excepto patrones de alta disponibilidad
- SUSE Enterprise Server 12 SP4, excepto patrones de alta disponibilidad
- SUSE Enterprise Server 12 SP5, excepto patrones de alta disponibilidad

Características

Información de aplicaciones de CloudWatch para SAP NetWeaver 7.0x-7.5x (incluida la plataforma ABAP) ofrece las siguientes características:

- Detección automática de cargas de trabajo de SAP NetWeaver
- Creación automática de alarmas de SAP NetWeaver basada en umbrales estáticos
- Reconocimiento automático de patrones de registro de SAP NetWeaver
- Panel de estado de SAP NetWeaver
- Panel de problemas de SAP NetWeaver

Requisitos previos

Debe completar los siguientes requisitos previos para configurar una base de datos SAP NetWeaver con Información de aplicaciones de CloudWatch:

- Habilitación de Systems Manager de AWS: instale SSM Agent en sus instancias de Amazon EC2 y habilite las instancias para SSM. Para obtener información acerca de la instalación de SSM Agent, consulte Configurar AWS Systems Manager en la Guía de usuario de AWSSystems Manager.
- Roles de instancias de Amazon EC2: debe adjuntar los siguientes roles de instancia de Amazon EC2 para configurar la supervisión de SAP NetWeaver.

  - Debe adjuntar el rol de `AmazonSSMManagedInstanceCore` para habilitar Systems Manager. Para obtener más información, consulte Ejemplos de políticas basadas en identidades de AWS Systems Manager.
  - Debe adjuntar la política `CloudWatchAgentServerPolicy` para permitir que las métricas y registros de instancias se emitan a través de CloudWatch. Consulte Creación de roles de IAM y usuarios para utilizarlos con el agente de CloudWatch para obtener más información.
- Grupos de recursos de AWS: para incorporar las aplicaciones a Información de aplicaciones de CloudWatch, cree un grupo de recursos que incluya todos los recursos de AWS asociados que se utilicen en la pila de aplicaciones. Esto incluye instancias de Amazon EC2 y volúmenes de Amazon EFS y de Amazon EBS en los que se ejecuten los servidores de aplicación de SAP NetWeaver. Si hay varios sistemas de SAP NetWeaver por cuenta, le recomendamos que cree un grupo de recursos que incluya los recursos de AWS para cada sistema de SAP NetWeaver. Para obtener más información acerca de la creación de grupos de recursos, consulte la Guía del usuario de grupos de recursos y etiquetas de AWS.
- Permisos de IAM: en el caso de los usuarios que no sean administradores, debe crear una política de AWS Identity and Access Management (IAM) a fin de que Información de aplicaciones cree un rol vinculado al servicio y este se asocie a la identidad del usuario. Para obtener información acerca de la creación de una política, consulte Política de IAM).
- Rol vinculado al servicio: Información de aplicaciones utiliza roles vinculados al servicio de AWS Identity and Access Management (IAM). Al crear su primera aplicación de Información de aplicaciones en la consola, Información de aplicaciones le crea el rol vinculado a un servicio. Para obtener más información, consulte Uso de roles vinculados a un servicio para CloudWatch Application Insights.
- Agente de Amazon CloudWatch: Información de aplicaciones instala y configura el agente de CloudWatch. Si tiene instalado el agente de CloudWatch, Información de aplicaciones conserva su configuración. Para evitar un conflicto de fusión, elimine la configuración de los recursos que desee utilizar en Información de aplicaciones del archivo de configuración del agente de CloudWatch existente. Para obtener más información, consulte Cree o edite de forma manual el archivo de configuración del agente de CloudWatch.

Configuración de los servidores de aplicaciones de SAP NetWeaver para la supervisión

Siga los siguientes pasos para configurar la supervisión de los servidores de aplicaciones de SAP NetWeaver.

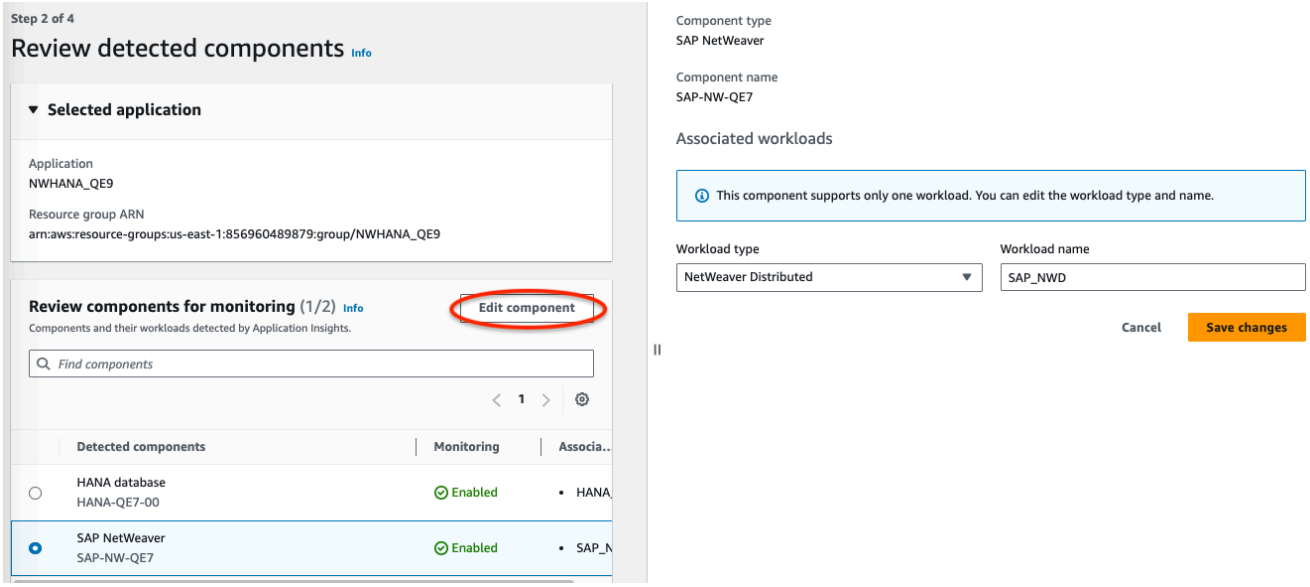
Configuración de la supervisión

1. Abra la consola de CloudWatch.
2. En el panel de navegación izquierdo, en Insights (Información), seleccione Application Insights (Información de aplicaciones).
3. En la página de Información de aplicaciones observará la lista de aplicaciones que se supervisen con Información de aplicaciones y el estado de supervisión de cada aplicación. En la esquina superior derecha, seleccione Add an application (Agregar una aplicación).
4. En la página Specify application details (Especificar detalles de la aplicación), en la lista desplegable que se encuentra debajo de Resource Groups (Grupos de recursos), seleccione el grupo de recursos de AWS que creó y que contiene los recursos de SAP NetWeaver. Si no ha creado un grupo de recursos para la aplicación, puede crear uno si selecciona Create new resource group (Crear nuevo grupo de recursos) en el menú desplegable de Resource Groups.
5. En Automatic monitoring of new resources (Monitoreo automático de nuevos recursos), active la casilla de verificación para permitir que Información de la aplicación supervise automáticamente los recursos que se agregan al grupo de recursos de la aplicación tras realizar la incorporación.
6. En Monitor EventBridge Events (Monitor de EventBridge Events), seleccione el casillero para integrar la supervisión de Información de aplicaciones con CloudWatch Events y obtener información de Amazon EBS, Amazon EC2, AWS CodeDeploy, Amazon ECS, API y notificaciones de AWS Health, Amazon RDS, Amazon S3 y AWS Step Functions.
7. En Integrate with AWS Systems Manager OpsCenter (Integrar con el OpsCenter de SYSlong), seleccione la casilla de verificación junto a Generate AWS Systems Manager OpsCenter OpsItems for remedial actions (Generar OpsItems de OpsCenter de SYSlong para obtener acciones correctivas) a fin de ver y recibir notificaciones cuando se detecten problemas en las aplicaciones seleccionadas. Para realizar un seguimiento de las operaciones que se lleven a cabo para resolver elementos de trabajo operativos, denominados OpsItems, relacionados con sus recursos de AWS, proporcione el ARN del tema de SNS.
8. Puede optar por ingresar etiquetas como ayuda para identificar y organizar sus recursos. Información de aplicaciones de CloudWatch es compatible con los grupos de recursos basados en etiquetas y en AWS CloudFormation, excepto los grupos de Application Auto Scaling. Para obtener más información, consulte Tag Editor (Editor de etiquetas) en la Guía del usuario de etiquetas y AWS Resource Groups.
9. Para revisar los componentes detectados, seleccione Siguiente.

10. En la página Revisar componentes detectados se enumeran los componentes supervisados y sus cargas de trabajo detectadas automáticamente por Información de aplicaciones de CloudWatch.

    - Para editar el tipo y el nombre de la carga de trabajo, elija Editar componente.

    > **Note**
    >
    > Los componentes que contienen una carga de trabajo NetWeaver Distributed o NetWeaver High Availability detectada admiten solo una carga de trabajo en un componente.



11. Elija Siguiente.
12. En la página Specify component details (Especificación de los detalles del componente), elija Next (Siguiente).
13. Revise la configuración de supervisión de aplicaciones y elija Enviar.
14. Se abrirá la página de detalles de la aplicación, donde podrá ver el Resumen de la aplicación, el Panel, los Componentes y las Cargas de trabajo. También podrá ver el Configuration history (Historial de configuración), los Log patterns (Patrones de registro) y cualquiera de las Tags (Etiquetas) que haya creado. Una vez que envíe la solicitud, Información de aplicaciones de CloudWatch implementa todas las métricas y alarmas de su sistema SAP NetWeaver, lo que puede tardar hasta una hora.

Administración de la supervisión de los servidores de aplicaciones de SAP NetWeaver

Siga los siguientes pasos para administrar la supervisión de los servidores de aplicaciones de SAP NetWeaver.

Administración de la supervisión

1. Abra la consola de CloudWatch.
2. En el panel de navegación izquierdo, en Insights (Información), seleccione Application Insights (Información de aplicaciones).
3. Seleccione la pestaña List view (Vista de lista).
4. En la página de Información de aplicaciones observará la lista de aplicaciones que se supervisen con Información de aplicaciones y el estado de supervisión de cada aplicación.
5. Seleccione la aplicación.
6. Elija la pestaña Components (Componentes).
7. Debajo de Monitored components (Componentes supervisados), seleccione el botón de radio situado junto al nombre del componente. A continuación, seleccione Manage monitoring (Administrar el monitoreo).
8. En Instance logs (Registros de instancias), puede actualizar la ruta de registro existente, el conjunto de patrones de registro y el nombre del grupo de registro. Además, puede agregar hasta tres Application logs (Registros de aplicaciones) adicionales.
9. En Metrics (Métricas), puede elegir las métricas de SAP NetWeaver según sus requisitos. Los nombres de métricas de SAP NetWeaver llevan el prefijo `sap`. Puede agregar hasta 40 métricas por componente.
10. Debajo de Custom alarms (Alarmas personalizadas), puede agregar alarmas adicionales para que Información de aplicaciones de CloudWatch las monitoree.
11. Revise la configuración de supervisión de aplicaciones y elija Save (Guardar). Una vez que envíe la configuración, su cuenta actualizará todas las métricas y alarmas de sus sistemas SAP NetWeaver.

Visualización y solución de los problemas de SAP NetWeaver que detecte Información de aplicaciones de CloudWatch

En las siguientes secciones se detallan los pasos que se tienen que seguir para resolver escenarios comunes de solución de problemas que se llevan a cabo al configurar la supervisión de SAP NetWeaver en Información de aplicaciones.

Temas de solución de problemas

- Problemas de conectividad de bases de datos de SAP NetWeaver
- Problemas de disponibilidad de las aplicaciones de SAP NetWeaver

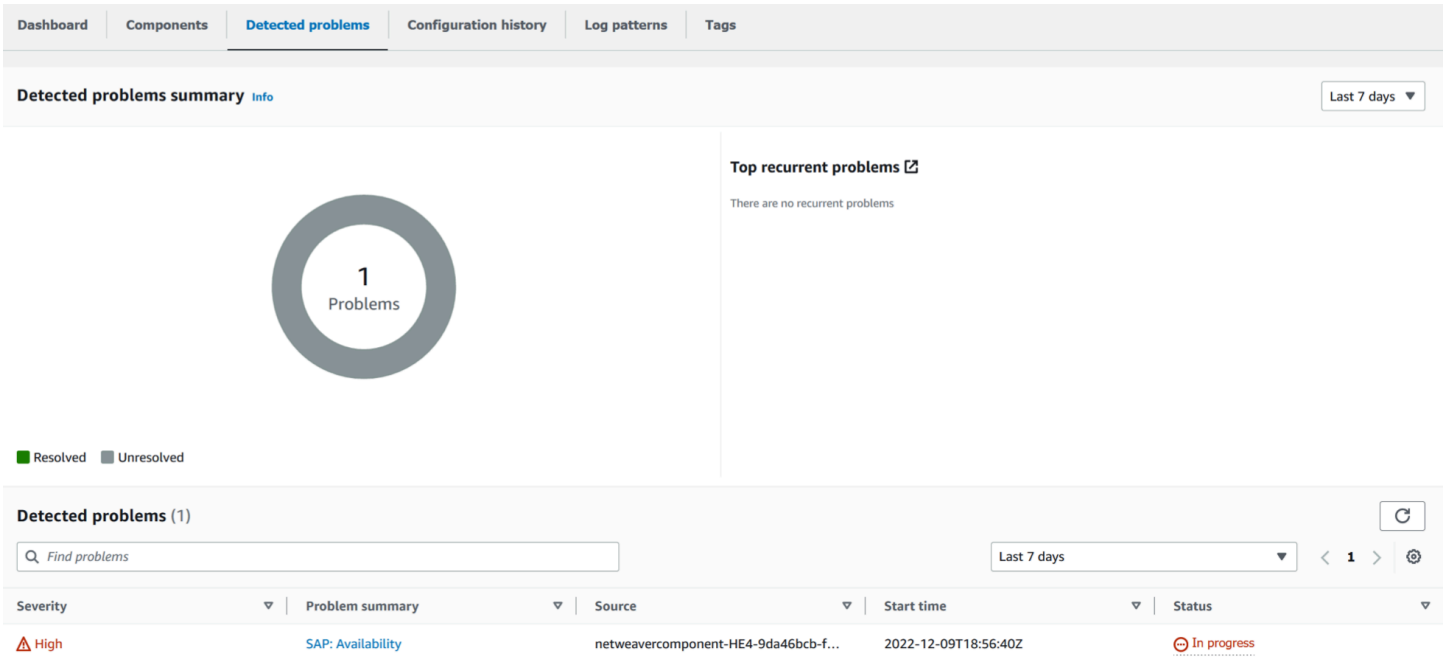
Problemas de conectividad de bases de datos de SAP NetWeaver

Descripción

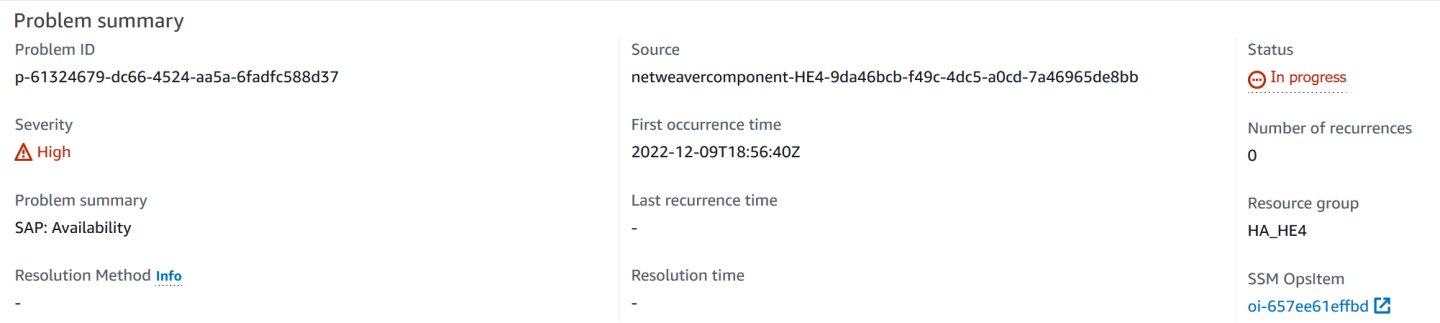
Su aplicación SAP NetWeaver tiene problemas de conectividad con la base de datos.

Causa

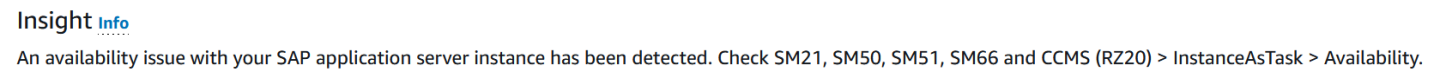
Para identificar el problema de conectividad, vaya a la consola de Información de aplicaciones de CloudWatch y consulte el panel de problemas de Información de aplicaciones de SAP NetWeaver. Seleccione el enlace en Problem summary (Resumen del problema) para ver el problema específico.



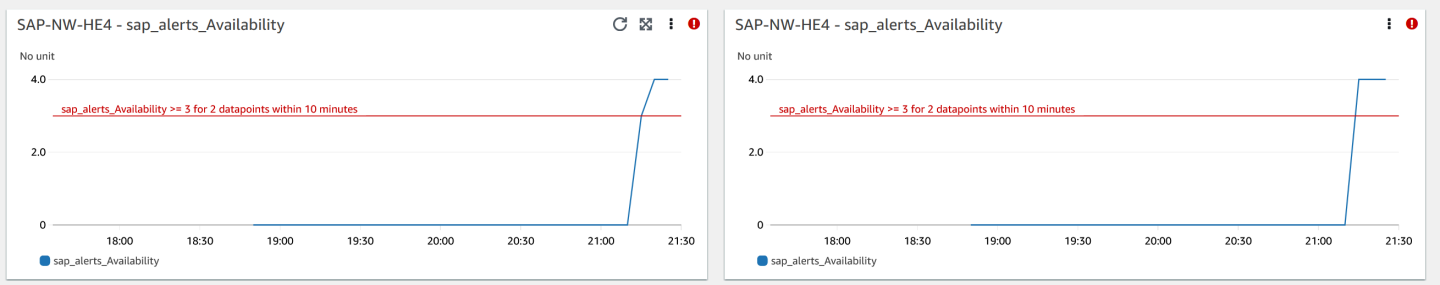
En el siguiente ejemplo, en Problem summary (Resumen del problema), el problema es SAP: Disponibilidad.



Inmediatamente después de la opción Problem summary, la sección Insight (Información) proporciona más contexto sobre el error y donde puede obtener más información sobre las causas del problema.

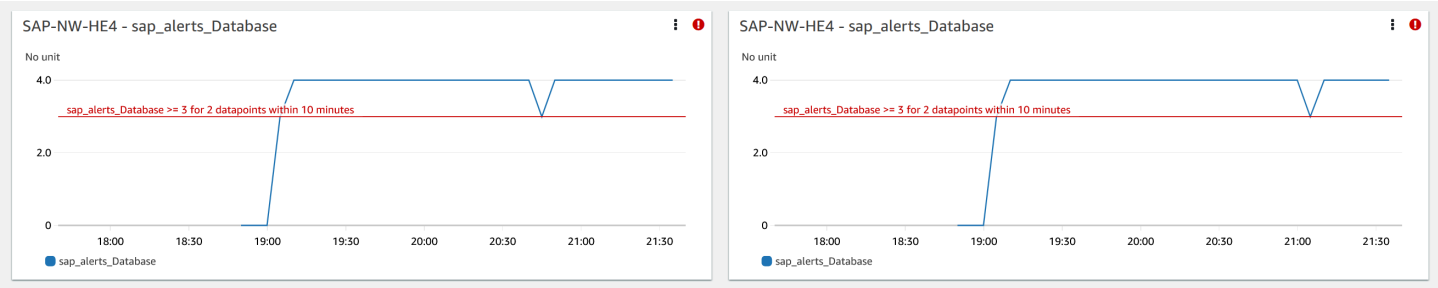


En el mismo panel de problemas, puede ver las métricas y los registros relacionados que la opción de detección de problemas agrupó para que pueda aislar la causa del error. La métrica `sap_alerts_Availability` hace un seguimiento de la disponibilidad del sistema SAP NetWeaver a lo largo del tiempo. Puede utilizar el seguimiento histórico para correlacionar el momento en que la métrica inició un estado de error o superó el umbral de alarma. En el siguiente ejemplo, hay un problema de disponibilidad con el sistema SAP NetWeaver. El ejemplo muestra dos alarmas porque hay dos instancias del servidor de aplicaciones SAP y, debido a ello, se creó una alarma para cada instancia.



Para obtener más información acerca de cada alarma, pase el ratón sobre el nombre de la métrica `sap_alerts_Availability`.

En el siguiente ejemplo, la métrica `sap_alerts_Database` muestra que la capa de base de datos tiene un problema o un error. Esta alarma indica que SAP NetWeaver ha tenido problemas para conectarse a su base de datos o para comunicarse con ella.

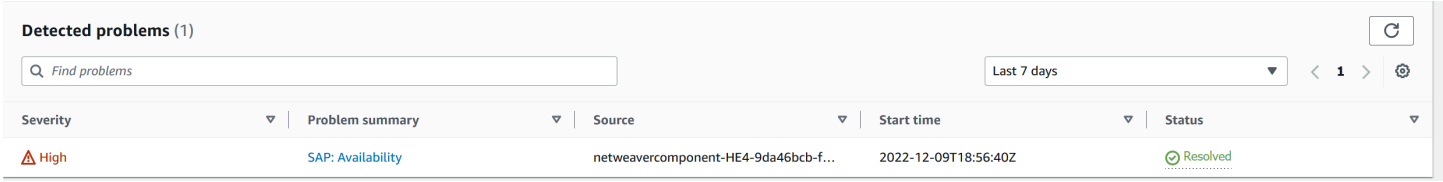


Dado que la base de datos es un recurso clave para SAP NetWeaver, es posible que reciba varias alarmas relacionadas cuando la base de datos tenga un problema o un error. En el siguiente ejemplo, las métricas `sap_alerts_FrontendResponseTime` y `sap_alerts_LongRunners` se inician porque la base de datos no está disponible.

Resolución

Información de aplicaciones supervisa el problema detectado cada hora. Si no hay nuevas entradas de registro relacionadas en sus archivos de registro de SAP NetWeaver, las entradas de registro más antiguas se considerarán resueltas. Debe corregir cualquier condición de error relacionada con las alarmas de CloudWatch. Una vez corregidas las condiciones de error, la alarma se resuelve cuando se recuperan las alarmas y los registros. Cuando se resuelvan todos los errores de registro y las alarmas de CloudWatch, Información de aplicaciones deja de detectar errores y el problema se resuelve automáticamente en una hora. Le recomendamos que resuelva todas las condiciones de error y las alarmas del registro para ver los problemas más recientes en el panel de problemas.

En el siguiente ejemplo, el problema de disponibilidad de SAP se resolvió.



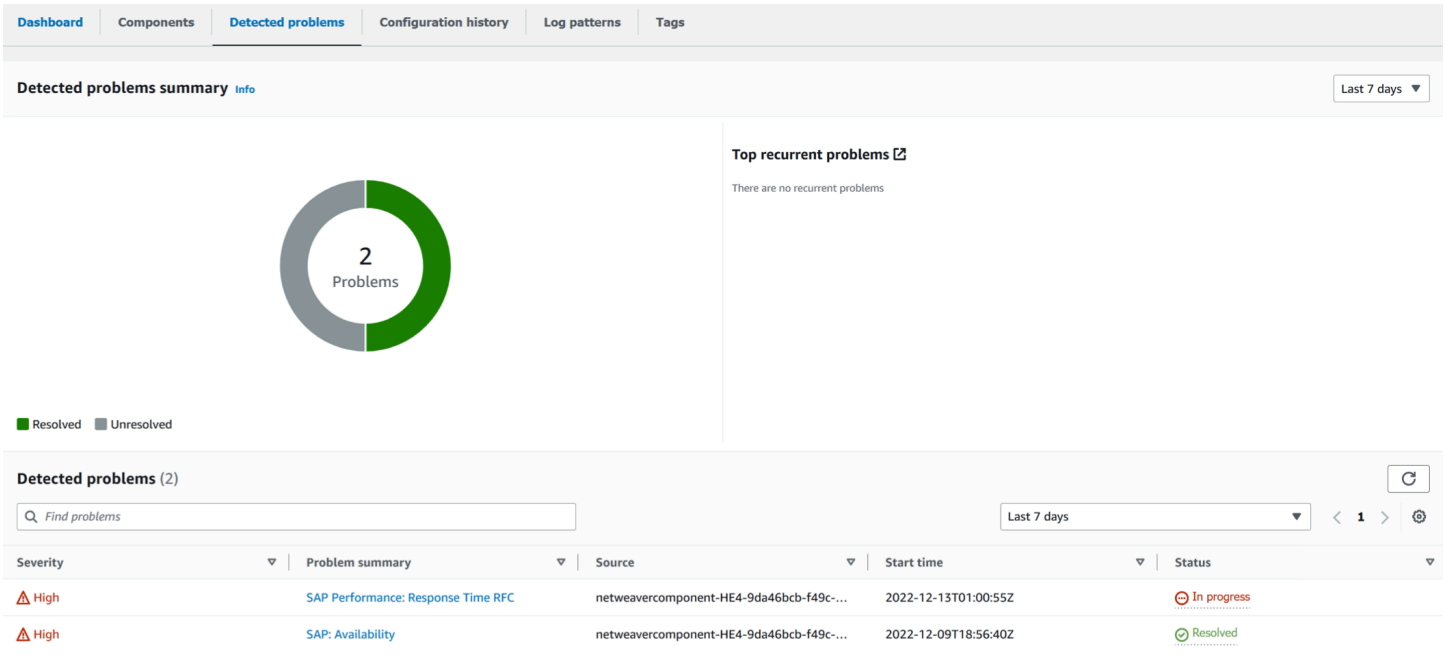
Problemas de disponibilidad de las aplicaciones de SAP NetWeaver

Descripción

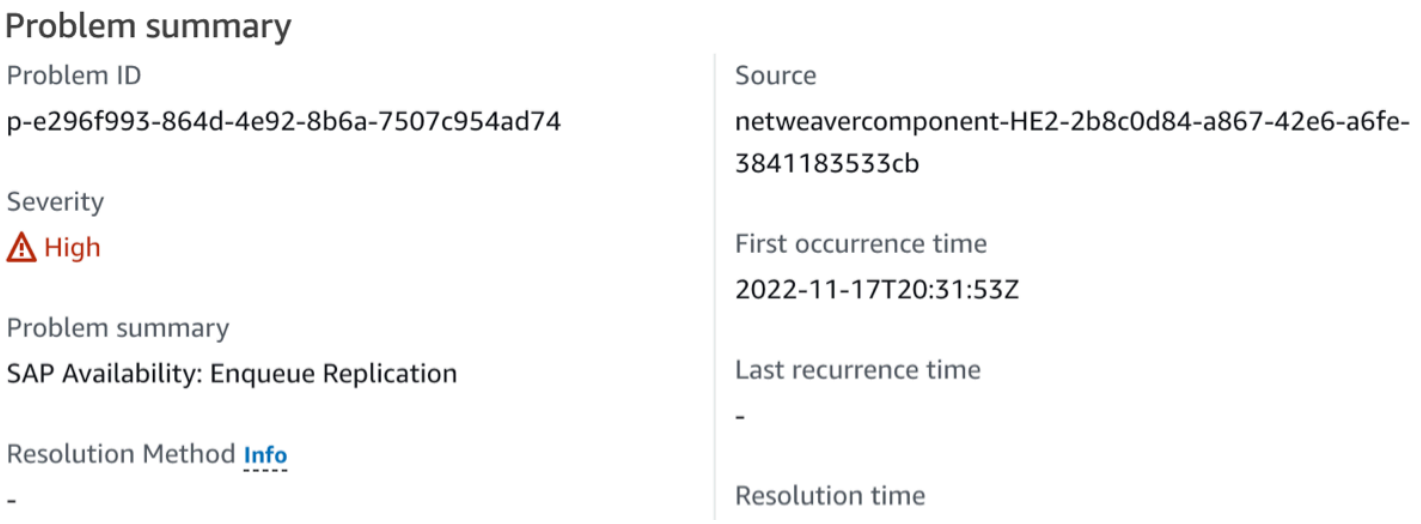
Su replicación en cola de alta disponibilidad de SAP NetWeaver dejó de funcionar.

Causa

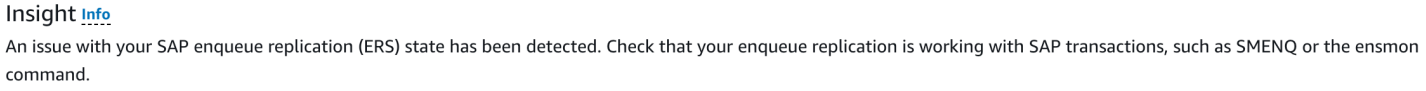
Para identificar el problema de conectividad, vaya a la consola de Información de aplicaciones de CloudWatch y consulte el panel de problemas de Información de aplicaciones de SAP NetWeaver. Seleccione el enlace en Problem summary (Resumen del problema) para ver el problema específico.



En el siguiente ejemplo, en Problem summary, el problema es la replicación en cola de alta disponibilidad.

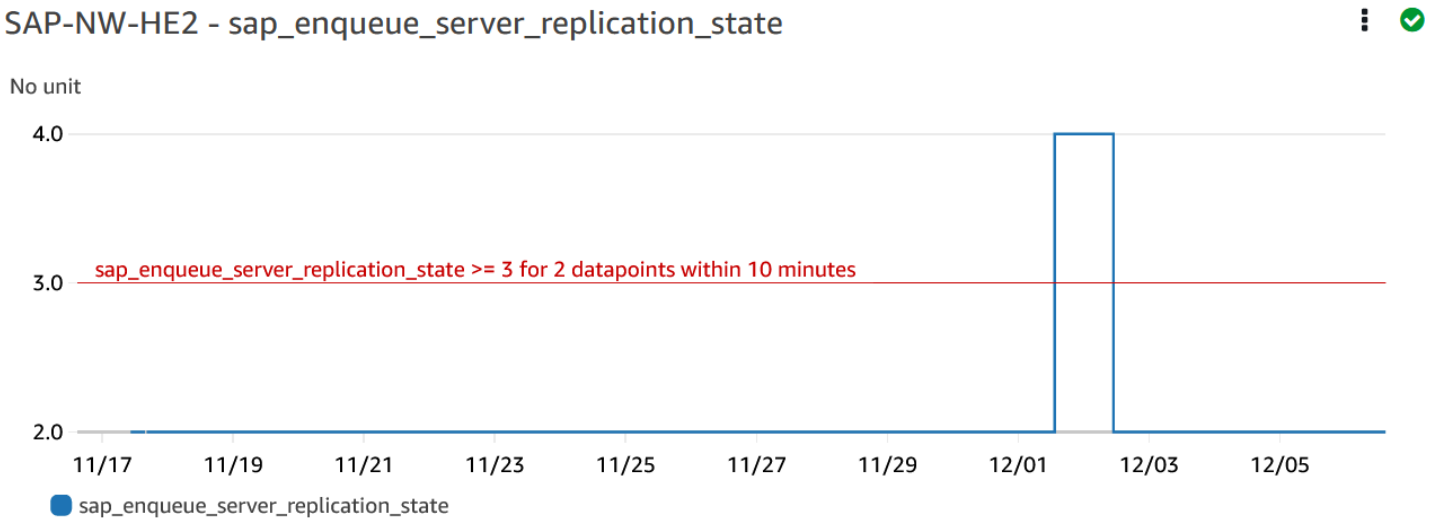


Inmediatamente después de la opción Problem summary, la sección Insight (Información) proporciona más contexto sobre el error y donde puede obtener más información sobre las causas del problema.

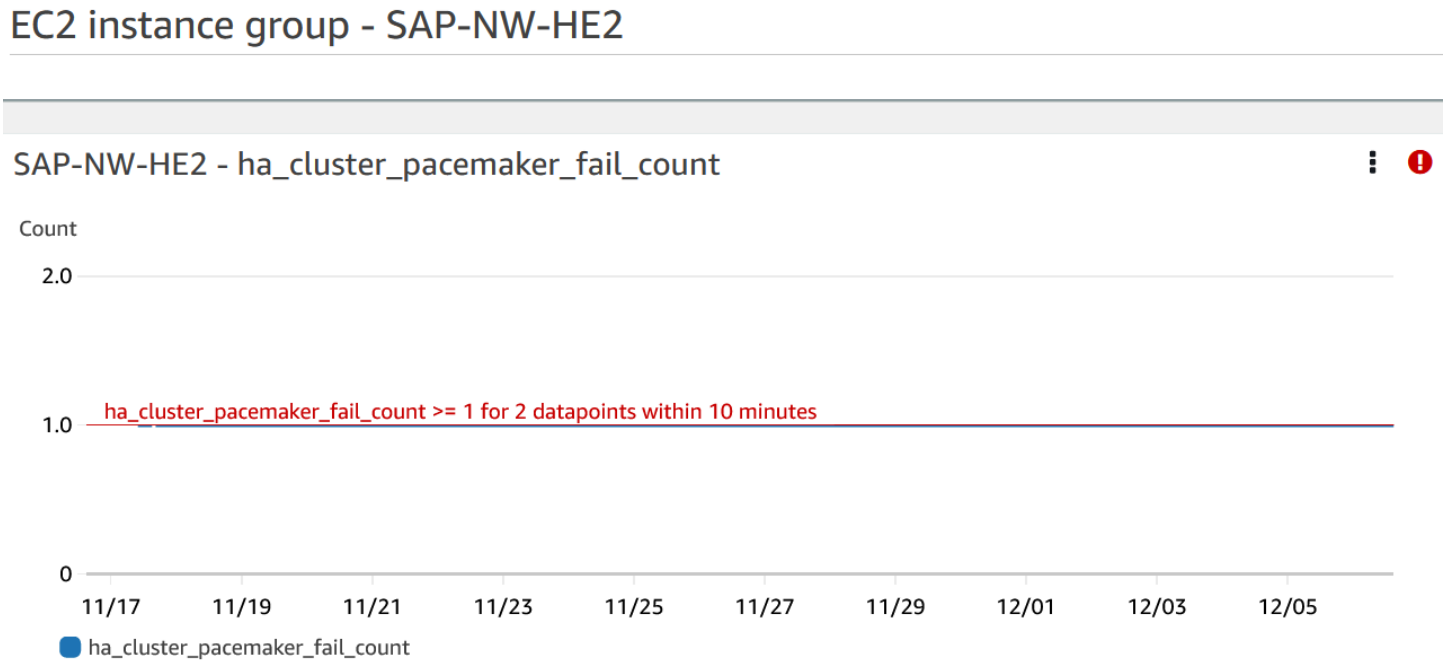


En el siguiente ejemplo se muestra el panel de problemas, donde puede ver los registros y las métricas que se hayan agrupado para que pueda aislar las causas del error. La métrica

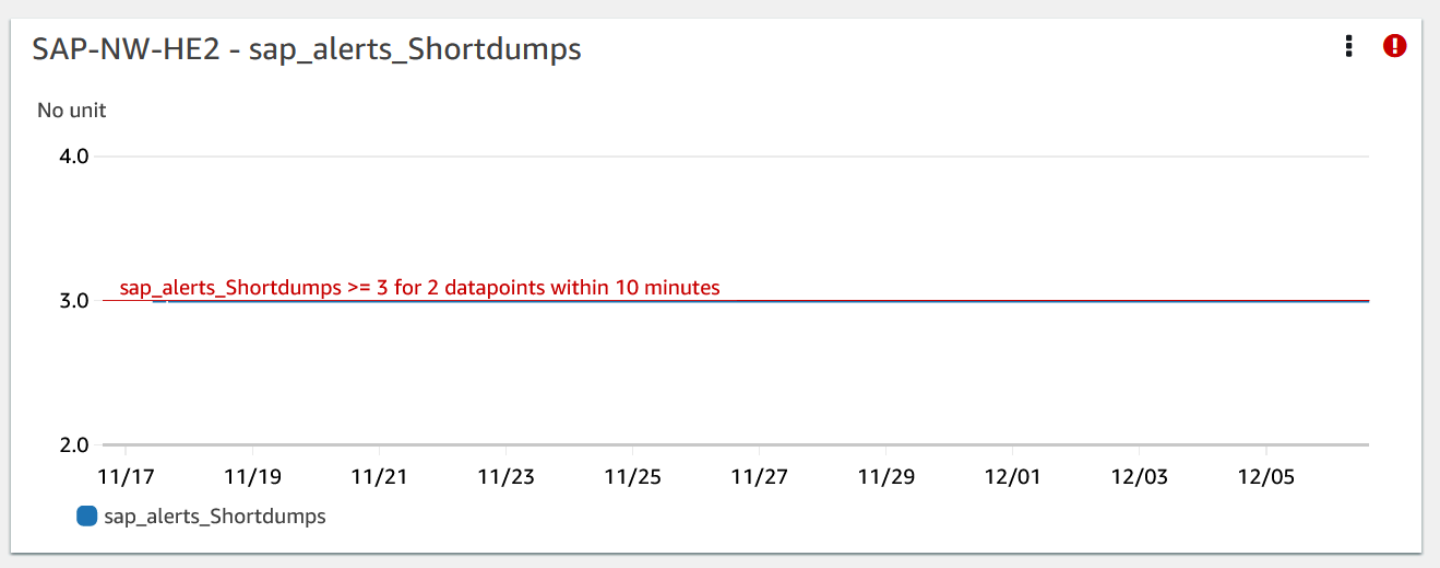
`sap_enqueue_server_replication_state` realiza un seguimiento del valor a lo largo del tiempo. Puede utilizar el seguimiento histórico para correlacionar el momento en que la métrica inició un estado de error o superó el umbral de alarma.



En el siguiente ejemplo, la métrica `ha_cluster_pacemaker_fail_count` muestra que el clúster de Pacemaker de alta disponibilidad sufrió un error en los recursos. Los recursos específicos de Pacemaker que tuvieron un recuento de errores superior o igual a uno se identifican en el panel de componentes.

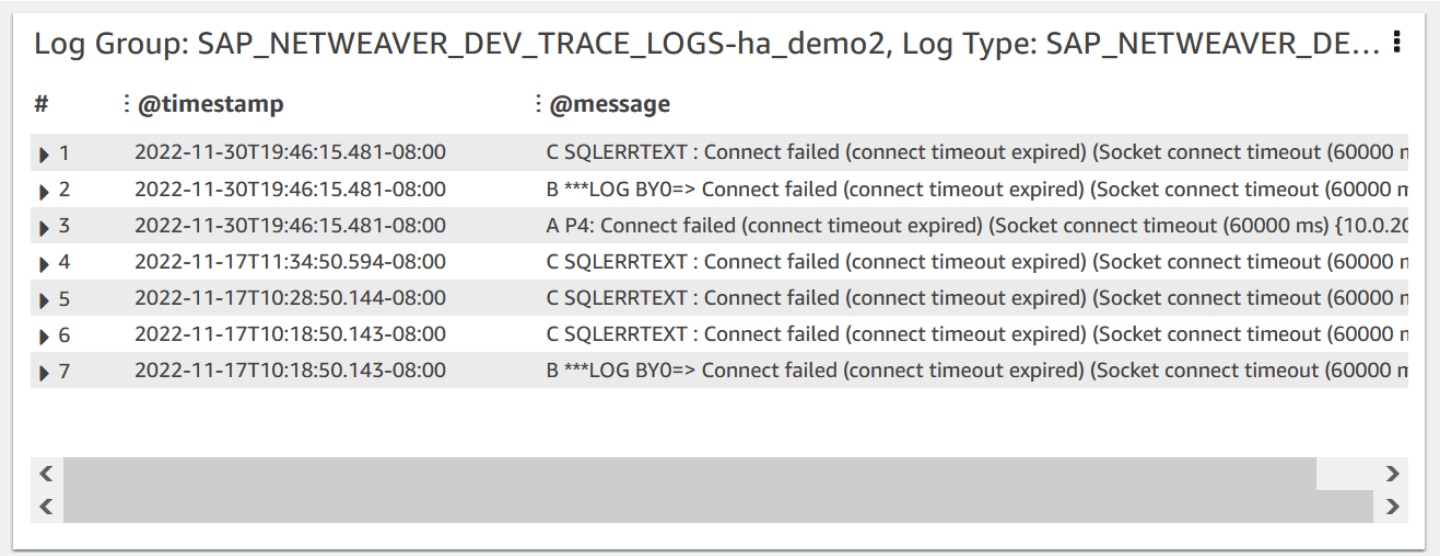


En el siguiente ejemplo se muestra la métrica `sap_alerts_Shortdumps`, que indica que el rendimiento de la aplicación SAP se redujo cuando se detectó el problema.

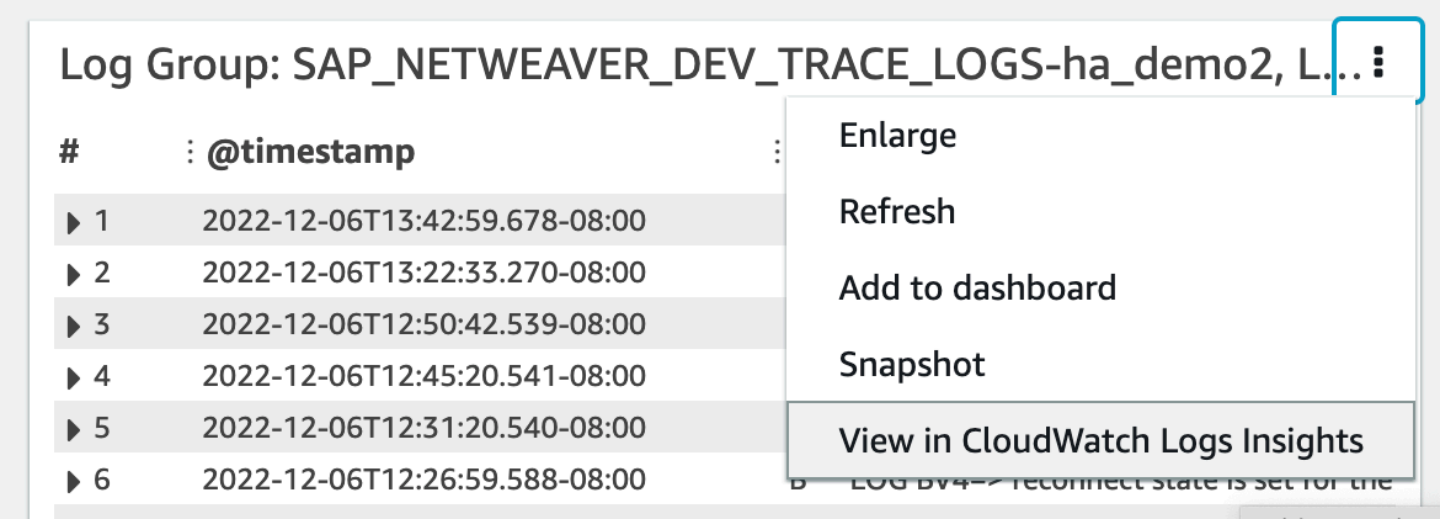


Registros

Las entradas de registro son útiles para comprender mejor los problemas que se produjeron en la capa de SAP NetWeaver cuando se detectó el problema. En el widget del grupo de registros en el panel de problemas, puede ver la hora específica del problema.



Para ver información detallada sobre los registros, seleccione los tres puntos verticales de la esquina superior derecha y seleccione View in CloudWatch Logs Insights (Ver en Información de registros de Amazon).

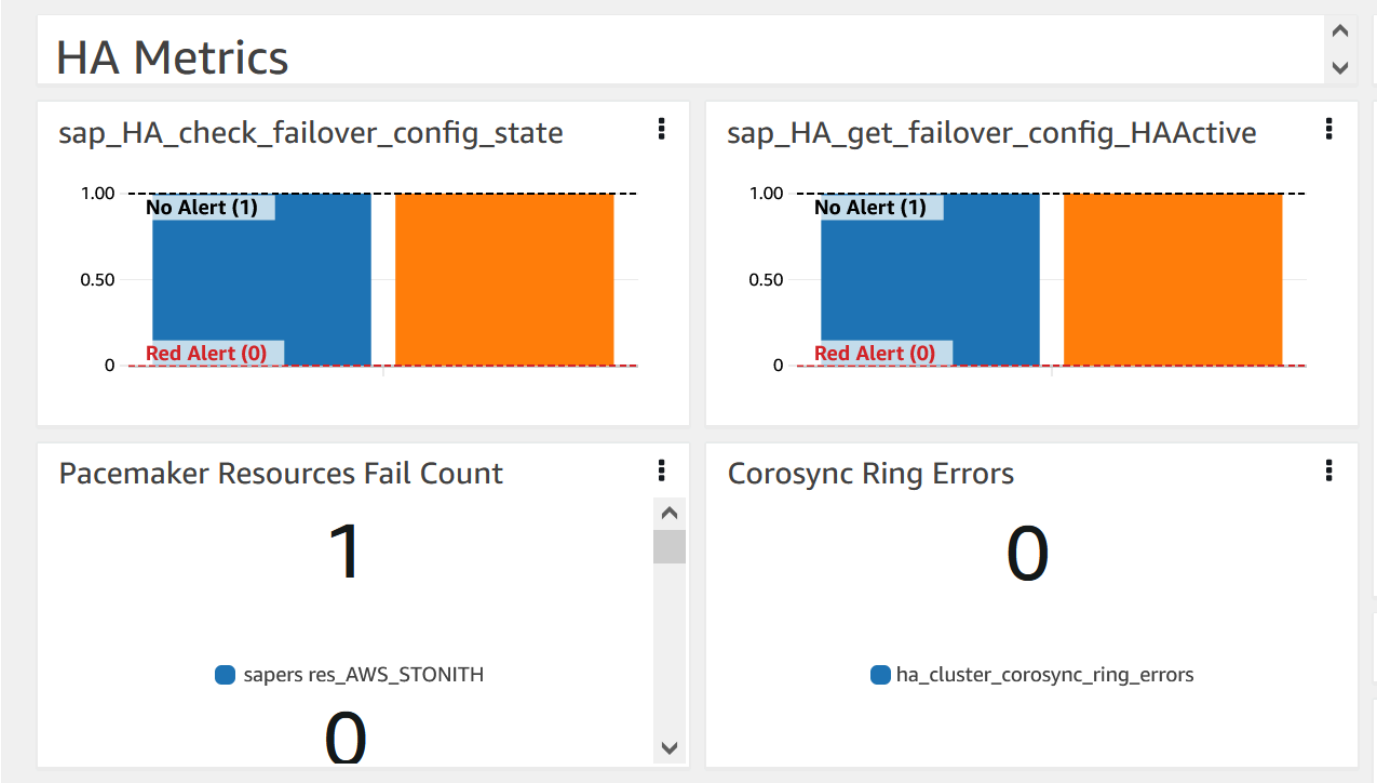


Siga los siguientes pasos para obtener más información sobre las métricas y las alarmas que se muestran en el panel de problemas.

Obtención de más información sobre las métricas y alarmas

1. Abra la consola de CloudWatch.
2. En el panel de navegación izquierdo, en Insights (Información), seleccione Application Insights (Información de aplicaciones). A continuación, elija la pestaña List view (Vista de lista) y seleccione su aplicación.
3. Seleccione la pestaña Components (Componentes). A continuación, seleccione el componente de SAP NetWeaver del que quiera obtener más información.

   En el siguiente ejemplo se muestra la sección HA Metrics (Métricas de alta disponibilidad) con la métrica `ha_cluster_pacemaker_fail_count` que se mostró en el panel de problemas.

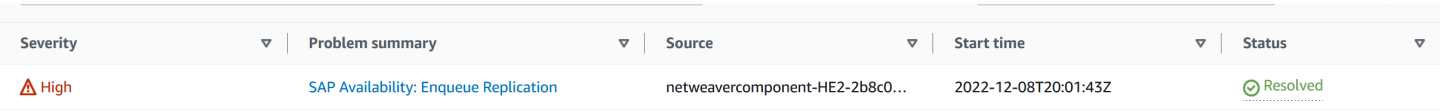


Resolución

Información de aplicaciones supervisa el problema detectado cada hora. Si no hay nuevas entradas de registro relacionadas en sus archivos de registro de SAP NetWeaver, las entradas de registro más antiguas se considerarán resueltas. Debe corregir cualquier condición de error relacionada con este problema.

En el caso de la alarma `sap_alerts_Shortdumps`, debe resolver la alerta en el sistema SAP NetWeaver mediante el código de transacción `RZ20 # R3Abap # Shortdumps` para acceder a la alerta del CCMS. Para más información acerca de las alertas CCMS, consulte el sitio web de SAP. Resuelva todas las alertas de CCMS en el árbol de Shortdumps. Una vez resueltas todas las alertas en el sistema SAP NetWeaver, CloudWatch ya no indica que la métrica está en estado de alarma.

Cuando se resuelvan todos los errores de registro y las alarmas de CloudWatch, Información de aplicaciones deja de detectar errores y el problema se resuelve automáticamente en una hora. Le recomendamos que resuelva todas las condiciones de error y las alarmas del registro para ver los problemas más recientes en el panel de problemas. En el siguiente ejemplo, se resuelve el problema de la replicación en cola de alta disponibilidad de SAP NetWeaver.



Solución de problemas de Información de aplicaciones para SAP NetWeaver

En esta sección se detallan los pasos a seguir para resolver los errores comunes que se presentan en el panel de Información de aplicaciones.

No se han podido agregar más de 60 métricas de monitoreo

Error devuelto: `Component cannot have more than 60 monitored metrics.`

Causa raíz: `The current metric limit is 60 monitor metrics per component.`

Resolución: elimine las métricas que no sean necesarias para cumplir con el límite.

Las métricas de SAP no aparecen en el panel después del proceso de incorporación

Causa principal: el panel de componentes utiliza un período métrico de cinco minutos para agregar los puntos de datos.

Resolución: todas las métricas deberían aparecer en el panel transcurridos cinco minutos.

Las métricas y alarmas de SAP no aparecen en el panel

Siga los siguientes pasos para saber por qué las métricas y alarmas de SAP no aparecen en el panel después del proceso de incorporación.

Identificación del problema con métricas y alarmas

1. Abra la consola de CloudWatch.
2. En el panel de navegación izquierdo, en Insights (Información), seleccione Application Insights (Información de aplicaciones). A continuación, elija la pestaña List view (Vista de lista) y seleccione su aplicación.
3. Elija la pestaña Configuration history (Historial de configuración).
4. Si ve que faltan puntos de datos de métricas, compruebe si hay errores relacionados con `prometheus-sap_host_exporter`.
5. Si no encuentra ningún error en el paso anterior, conéctese a su instancia de Linux. En el caso de implementaciones de alta disponibilidad, conéctese a la instancia de Amazon EC2 del clúster principal.

6. En la instancia, compruebe que el exportador se esté ejecutando mediante el siguiente comando. El puerto predeterminado es 9680. Si usa un puerto diferente, sustituya 9680 por el puerto que esté utilizando.

```
curl localhost:9680/metrics
```

Si no se devuelve ningún dato, quiere decir que el exportador no pudo iniciarse.

7. Para encontrar la convención de nomenclatura correcta de WORKLOAD_SHORT_NAME para utilizarla en los dos pasos siguientes, ejecute el siguiente comando.

> **Note**
>
> Información de aplicaciones agrega el sufijo WORKLOAD_SHORT_NAME al nombre del servicio en función de la carga de trabajo que se esté ejecutando. Los nombres abreviados de las implementaciones distribuidas, estándar y de alta disponibilidad de NetWeaver son SAP_NWD, SAP_NWS y SAP_NWH.

```
sudo systemctl | grep exporter
```

8. Ejecute el siguiente comando para verificar si hay errores en los registros de los servicios del exportador:

```
sudo journalctl -e --unit=prometheus-sap_host_exporter_WORKLOAD_SHORT_NAME.service
```

9. Ejecute el siguiente comando para verificar si hay errores en los registros de los servicios del administrador:

```
sudo journalctl -e --unit=prometheus-
sap_host_exporter_manager_WORKLOAD_SHORT_NAME.service
```

> **Note**
>
> Este servicio debe estar activo y en funcionamiento en todo momento.

Si este comando no devuelve ningún error, vaya al siguiente paso.

10. Ejecute el siguiente comando para iniciar el exportador de forma manual. A continuación, compruebe la salida del exportador.

    ```
    sudo /opt/aws/sap_host_exporter/sap_host_exporter
    ```

    Puede salir del proceso de exportación después de comprobar si hay errores.

Causa raíz: existen varias causas posibles para este problema. Una causa común es que el exportador no puede conectarse a una de las instancias del servidor de aplicaciones.

Resolución

Siga los pasos siguientes para conectar el exportador a las instancias del servidor de aplicaciones. Podrá comprobar que la instancia de la aplicación SAP esté en ejecución y tendrá que usar SAPControl para conectarse a esta.

Conexión del exportador a las instancias del servidor de aplicaciones

1. En la instancia de Amazon EC2, ejecute el siguiente comando para comprobar que la aplicación SAP se ejecute.

   ```
   sapcontrol -nr <App_InstNo> -function GetProcessList
   ```

2. Debe establecer una conexión SAPControl que funcione. Si la conexión de SAPControl no funciona, busque la causa raíz del problema en la instancia de la aplicación SAP correspondiente.
3. Para iniciar manualmente el exportador después de solucionar el problema de conexión a SAP Control, ejecute el siguiente comando:

   ```
   sudo systemctl start prometheus-sap_host_exporter.service
   ```

4. Si no puede resolver el problema con la conexión de SAPControl, use el procedimiento siguiente como solución temporal.

   a. Abra la consola de AWS Systems Manager.

   b. En el panel de navegación izquierdo, elija State Manager (Administrador de estado).

   c. En Associations (Asociaciones), busque la asociación del sistema SAP NetWeaver.

```
Association Name: Equal: AWS-ApplicationInsights-
SSMSAPHostExporterAssociationForCUSTOMSAPNW<SID>-1
```

d. Seleccione el Association id (ID de asociación).

e. Elija la pestaña Parameters (Parámetros) y elimine el número del servidor de la aplicación del elemento AdditionalArguments.

f. Seleccione Apply Association Now (Aplicar asociación ahora).

> **Note**
>
> Recuerde que se trata de una solución temporal. Si se realizan actualizaciones en las configuraciones de supervisión del componente, la instancia se volverá a agregar.

Visualización y solución de los problemas que Información de aplicaciones de Amazon CloudWatch haya detectado

En los temas de esta sección se observa información detallada sobre los problemas detectados y las observaciones que se muestran en Información de aplicaciones. También encontrará soluciones sugeridas para problemas detectados con su cuenta o su configuración.

Temas de solución de problemas

- Información general de la consola de CloudWatch
- Página de resumen de problemas de Información de aplicaciones
- Fallos en los conflictos de fusión de agentes de CloudWatch
- Las alarmas no se crean
- Comentarios
- Errores de configuración

Información general de la consola de CloudWatch

En el panel de Información de aplicaciones de CloudWatch que se encuentre en la página de información general de la consola de Información de aplicaciones de CloudWatch, encontrará

información general sobre los problemas que afecten a las aplicaciones monitoreadas. Para obtener más información, consulte Introducción a Información de aplicaciones de Amazon CloudWatch.

En el panel de información general de Información de aplicaciones de CloudWatch observará lo siguiente:

- la gravedad de los problemas detectados: alta/media/baja
- Un breve resumen del problema
- el origen del problema
- La hora a la que comenzó el problema
- El estado de resolución del problema
- el grupo de recursos afectado.

Para ver los detalles de un problema específico, seleccione la descripción del problema en Problem Summary (Resumen del problema). Un panel detallado muestra información acerca del problema, las anomalías de las métricas relacionadas y fragmentos de los errores de registro. Puede brindar comentarios sobre la relevancia de la información si indica si le resulta útil.

En caso de que se detecte un nuevo recurso sin configurar, la descripción del resumen del problema lo llevará al asistente Edit configuration (Editar configuración) para que configure el nuevo recurso. Puede ver o editar la configuración de su grupo de recursos si selecciona View/edit configuration (Ver/editar configuración) en la esquina superior derecha del panel detallado.

Para volver a la información general, elija Back to overview (Volver a la información general), situada junto a la cabecera del panel detallado de Información de aplicaciones de CloudWatch.

Página de resumen de problemas de Información de aplicaciones

Página de resumen de problemas de Información de aplicaciones

En la página de resumen de problemas de Información de aplicaciones de CloudWatch, encontrará la siguiente información sobre los problemas detectados:

- Un breve resumen del problema
- La hora de inicio y la fecha del problema
- La gravedad del problema: alta/media/baja
- El estado del problema detectado: en curso/resuelto

- Información: información generada automáticamente sobre el problema detectado y su posible causa raíz
- Comentarios sobre la información: comentarios que haya proporcionado sobre la utilidad de la información que Información de aplicaciones de CloudWatch genera.
- Observaciones relacionadas: una vista detallada de las anomalías de las métricas y fragmentos de errores de los registros pertinentes relacionados con el problema en los distintos componentes de la aplicación

Fallos en los conflictos de fusión de agentes de CloudWatch

Información de aplicaciones de CloudWatch instala y configura el agente CloudWatch en las instancias del cliente. Esto incluye la creación de un archivo de configuración del agente de CloudWatch con configuraciones para métricas o registros. Se puede producir un conflicto de fusión si la instancia de un cliente ya tiene un archivo de configuración del agente de CloudWatch con diferentes configuraciones definidas para las mismas métricas o registros. Para resolver el conflicto de fusión, siga estos pasos:

1. Identifique los archivos de configuración del agente de CloudWatch en su sistema. Para obtener más información acerca de las ubicaciones de los archivos, consulte Archivos y ubicaciones del agente de CloudWatch.
2. Elimine la configuración de los recursos que desee utilizar en Información de aplicaciones del archivo de configuración del agente de CloudWatch existente. Si solo desea utilizar las configuraciones de Información de aplicaciones, elimine los archivos de configuración del agente CloudWatch existentes.

Las alarmas no se crean

Para algunas métricas, Información de aplicaciones predice el umbral de alarma en función de los puntos de datos anteriores de la métrica. Para habilitar esta predicción, se deben cumplir los siguientes criterios.

- Puntos de datos recientes: debe haber un mínimo de 100 puntos de datos de las últimas 24 horas. Los puntos de datos no necesitan ser continuos y pueden estar dispersos a lo largo de un período de 24 horas.

- Datos históricos: debe haber un mínimo de 100 puntos de datos que abarquen el período comprendido entre 15 días antes de la fecha actual y 1 día antes de la fecha actual. Los puntos de datos no necesitan ser continuos y pueden estar dispersos a lo largo de un período de 15 días.

> **Note**
>
> En el caso de algunas métricas, Información de aplicaciones retrasa la creación de alarmas hasta que se cumplan las condiciones anteriores. En este caso, aparece un evento en el historial de configuración en el que la métrica carece de puntos de datos suficientes para establecer el umbral de alarma.

Comentarios

Comentarios

Puede proporcionar comentarios sobre la información generada automáticamente sobre los problemas detectados indicando si es útil o no es útil. Sus comentarios sobre la información, junto con los diagnósticos de la aplicación (anomalías de métricas y excepciones de registro), se utilizan para mejorar la detección de problemas similares en el futuro.

Errores de configuración

Información de aplicaciones de CloudWatch utiliza su configuración para crear telemetrías de supervisión para los componentes. Cuando Información de aplicaciones detecta un problema con su cuenta o con la configuración, podrá encontrar información en el campo Observaciones sobre cómo resolver el problema de configuración de su aplicación.

En la siguiente tabla se muestran las soluciones recomendadas para observaciones específicas.

Observaciones	Resolución que se sugiere	Notas adicionales
La cuota de CloudFormation ya se ha alcanzado.	Información de aplicaciones crea una pila de CloudFormation para cada aplicación para administrar la instalación y configuración de CloudWatch para todos los component	n/a

Observaciones	Resolución que se sugiere	Notas adicionales
	es de la aplicación. De forma predeterminada, cada cuenta de AWS puede tener 2000 pilas. Consulte Límites de AWS CloudFormation. Para solucionar este problema, aumente el límite de pilas de CloudFormation.	
No hay ningún rol de instancia de SSM en las siguientes instancias.	Para que Información de aplicaciones pueda instalar y configurar el agente de CloudWatch en las instancias de la aplicación, las políticas de AmazonSSMManagedInstanceCore y CloudWatchAgentServerPolicy deben estar adjuntadas al rol de la instancia.	Información de aplicaciones llama a la API DescribeInstanceInformation de SSM para obtener la lista de instancias con permiso de SSM. Una vez asociado el rol a la instancia, SSM tarda algún tiempo en incluir la instancia en el resultado de DescribeInstanceInformation. Hasta que SSM incluye la instancia en el resultado, aparece el error NO_SSM_INSTANCE_ROLE para la aplicación.
Los nuevos componentes pueden necesitar configuración.	Información de aplicaciones detecta que hay nuevos componentes en el grupo de recursos de la aplicación. Para solucionar esto, configure los nuevos componentes según corresponda.	n/a

Registros y métricas que Información de aplicaciones de Amazon CloudWatch admite

A continuación, se indican los registros y las métricas que se admiten para Información de aplicaciones de Amazon CloudWatch.

Información de aplicaciones de CloudWatch admite los siguientes registros:

- Registros de Microsoft Internet Information Services (IIS)
- Registro de errores de SQL Server en EC2
- Registros de aplicaciones .NET personalizados, como Log4Net
- Registros de eventos de Windows, incluidos registros de Windows (sistema, aplicación y seguridad) y registro de aplicaciones y servicios
- Amazon CloudWatch Logs para AWS Lambda
- Registro de errores y registro lento para RDS de MySQL, MySQL de Aurora y MySQL en EC2
- Registro de Postgresql para RDS de PostgreSQL y PostgreSQL en EC2
- Amazon CloudWatch Logs para AWS Step Functions
- Registros de ejecución y registros de acceso (JSON, CSV y XML, pero no CLF) para etapas de la API REST de API Gateway
- Registros del JMX Exporter de Prometheus (EMF)
- Registros de alertas y registros de agente de escucha para Oracle en Amazon RDS y Oracle en Amazon EC2
- Los contenedores registran el enrutamiento desde los contenedores de Amazon ECS a CloudWatch mediante `awslogs` log driver (controlador de registro).
- Los contenedores registran el enrutamiento desde los contenedores de Amazon ECS a CloudWatch mediante FireLens container log router (Enrutador de registro de contenedores FireLens).
- El contenedor registra el enrutamiento desde Amazon EKS o Kubernetes, que se ejecutan en Amazon EC2, a CloudWatch mediante Procesador de registro Fluentd o Bit Fluentd con Información de contenedores.
- Registros de seguimiento y errores de SAP HANA
- Registros de Pacemaker de alta disponibilidad
- Registros del servidor de SAP ASE
- Registros del servidor de copia de seguridad de SAP ASE

- Registros del servidor de replicación de SAP ASE
- Registros del agente RMA SAP ASE
- Registros del administrador de errores de SAP ASE
- Registros de seguimiento de desarrolladores de SAP NetWeaver
- Métricas de procesos para procesos de Windows mediante complemento proctstat para el agente CloudWatch
- Registros de consultas de DNS públicos para la zona alojada
- Registros de consultas de DNS de Amazon Route 53 Resolver

Información de aplicaciones de CloudWatch admite las siguientes clases de registro:

- Estándar: Información de aplicaciones de Amazon CloudWatch requiere que los grupos de registros se configuren con la clase de registro estándar de los Registros de CloudWatch para permitir el monitoreo.

Información de aplicaciones de CloudWatch admite métricas para los siguientes componentes de aplicaciones:

- Amazon Elastic Compute Cloud (EC2)
  - Métricas integradas de CloudWatch
  - Métricas del agente de CloudWatch (servidor de Windows)
  - Métricas de procesos del agente de CloudWatch (servidor de Windows)
  - Métricas del agente de CloudWatch (servidor Linux)
- Elastic Block Store (EBS)
- Amazon Elastic File System (Amazon EFS)
- Elastic Load Balancer (ELB)
- Application ELB
- Grupos de Amazon EC2 Auto Scaling
- Amazon Simple Queue Server (SQS)
- Amazon Relational Database Service (RDS)
  - Instancias de base de datos de RDS
  - Clústeres de base de datos de RDS
- Función AWS Lambda

- Tabla de Amazon DynamoDB.
- Bucket de Amazon S3
- AWS Step Functions
  - Nivel de ejecución
  - Actividad
  - Función de Lambda
  - Integración con los servicios
  - API de Step Functions
- Etapas de la API de REST de API Gateway
- SAP HANA
- SAP ASE
- Alta disponibilidad de SAP ASE en Amazon EC2
- SAP NetWeaver
- Clúster de alta disponibilidad
- Java
- Amazon Elastic Container Service (Amazon ECS)
  - Métricas integradas de CloudWatch
  - Métricas de Información de contenedores
  - Métricas de Prometheus de Información de contenedores
- Kubernetes en AWS
  - Métricas de Información de contenedores
  - Métricas de Prometheus de Información de contenedores
- Amazon FSx
- Amazon VPC
- Puerta de enlace NAT de Amazon VPC
- Comprobación de estado de Amazon Route 53
- Zona alojada de Amazon Route 53
- Punto de conexión de Amazon Route 53 Resolver
- Grupo de reglas de AWS Network Firewall
- Asociación de grupos de reglas AWS Network Firewall

- Métricas con requisitos de puntos de datos
  - AWS/ApplicationELB
  - AWS/AutoScaling
  - AWS/EC2
  - Elastic Block Store (EBS)
  - AWS/ELB
  - AWS/RDS
  - AWS/Lambda
  - AWS/SQS
  - AWS/CWAgent
  - AWS/DynamoDB
  - AWS/S3
  - Estados de AWS
  - ApiGateway de AWS
  - AWS/SNS
- Métricas recomendadas
- Métricas de contador de rendimiento

Amazon Elastic Compute Cloud (EC2)

Información de aplicaciones de CloudWatch es compatible con las siguientes métricas:

Métricas

- Métricas integradas de CloudWatch
- Métricas del agente de CloudWatch (servidor de Windows)
- Métricas de procesos del agente de CloudWatch (servidor de Windows)
- Métricas del agente de CloudWatch (servidor Linux)

Métricas integradas de CloudWatch

CPUCreditBalance

CPUCreditUsage

CPUSurplusCreditBalance

CPUSurplusCreditsCharged

CPUUtilization

DiskReadBytes

DiskReadOps

DiskWriteBytes

DiskWriteOps

EBSByteBalance%

EBSIOBalance%

EBSReadBytes

EBSReadOps

EBSWriteBytes

EBSWriteOps

NetworkIn

NetworkOut

NetworkPacketsIn

NetworkPacketsOut

StatusCheckFailed

StatusCheckFailed_Instance

StatusCheckFailed_System

Métricas del agente de CloudWatch (servidor de Windows)

N.º de excepciones de CLR de.NET lanzadas

.NET CLR Exceptions: n.º de excepciones generadas/segundo

.NET CLR Exceptions: n.º de filtros/segundo

.NET CLR Exceptions: n.º de instrucciones finally/segundo

.NET CLR Exceptions: generación para capturar profundidad/segundo

.NET CLR Interop: n.º de CCWs

.NET CLR Interop: n.º de stubs

.NET CLR Interop: n.º de exportaciones TLB/segundo

.NET CLR Interop: n.º de importaciones TLB/segundo

.NET CLR Interop: n.º de serialización

.NET CLR Jit: % de tiempo en Jit

.NET CLR Jit: errores de Jit estándar

.NET CLR Loading: % de tiempo de carga

.NET CLR Loading: errores de velocidad de carga

.NET CLR LocksAndThreads: tasa de contención/segundo

.NET CLR LocksAndThreads: longitud de cola/segundo

.NET CLR Memory: n.º de bytes confirmados totales

Tiempo de % de memoria de CLR de .NET en GC

.NET CLR Networking 4.0.0.0: tiempo medio de cola de HttpWebRequest

.NET CLR Networking 4.0.0.0: solicitudes HttpWebRequest anuladas/segundo

.NET CLR Networking 4.0.0.0: solicitudes HttpWebRequest con errores/segundo

.NET CLR Networking 4.0.0.0: solicitudes HttpWebRequest en cola/segundo

APP_POOL_WAS Errores totales de ping en el proceso de trabajo

Reinicios de aplicaciones ASP.NET

Porcentaje de tiempo del procesador administrado (estimado) de las aplicaciones ASP.NET

Errores de aplicaciones ASP.NET totales/sec

Errores de aplicaciones ASP.NET no controlados durante la ejecución/s

Solicitudes de aplicaciones ASP.NET en la cola de aplicaciones

Solicitudes de aplicaciones ASP.NET/s

Tiempo de espera de solicitudes ASP.NET

Solicitudes ASP.NET en cola

Colas de solicitudes de servicio HTTP CurrentQueueSize

Porcentaje de espacio libre en disco lógico

Porcentaje de bytes confirmados en uso en memoria

Mbytes disponibles de memoria

Páginas de memoria/segundo

Total de bytes de interfaz de red/segundo

Porcentaje de uso de archivo de paginación

Porcentaje de tiempo de disco en disco físico

Promedio de disco físico Longitud de la cola de disco

Promedio de disco físico Lectura de disco/segundo

Promedio de disco físico Escritura en disco/segundo

Bytes leídos/segundo en disco físico

Lecturas/segundo en disco físico

Bytes de escritura/segundo en disco físico

Escrituras/segundo en disco físico

Porcentaje de tiempo de inactividad del procesador

Porcentaje de horas de interrupción del procesador

Porcentaje de tiempo de procesador del procesador

Porcentaje de tiempo de usuario del procesador

SQLServer: métodos de acceso: registros reenviados/segundo

SQLServer: métodos de acceso: exploraciones completas/segundo

SQLServer: métodos de acceso: divisiones de página/segundo

SQLServer: administrador del búfer: porcentaje de aciertos de caché del búfer

SQLServer: administrador del búfer: duración prevista de página

SQLServer: estadísticas generales: procesos bloqueados

SQLServer: estadísticas generales: conexiones de usuario

SQLServer: bloqueos temporales: tiempo medio de espera de bloqueo temporal (ms)

SQLServer: bloqueos: tiempo medio de espera de bloqueo (ms)

SQLServer: bloqueos: tiempos de espera de bloqueo agotados/segundo

SQLServer: bloqueos: esperas de bloqueo/segundo

SQLServer: bloqueos: número de interbloqueos/segundo

SQLServer: administrador de memoria: concesiones de memoria pendientes

SQLServer: estadísticas de SQL: solicitudes por lotes/segundo

SQLServer: estadísticas de SQL: compilaciones de SQL/segundo

SQLServer: estadísticas de SQL: recopilaciones de SQL/segundo

Longitud de cola del procesador del sistema

Conexiones TCPv4 establecidas

Conexiones TCPv6 establecidas

W3SVC_W3WP: descargas de caché de archivos

W3SVC_W3WP: errores de caché de archivos

W3SVC_W3WP: solicitudes/segundo

W3SVC_W3WP: descargas de caché de URI

W3SVC_W3WP: errores de caché de URI

Bytes de servicios web recibidos/s

Bytes de servicios web enviados/s

Intentos de conexión de servicio web/segundo

Conexiones actuales de servicios web

Solicitudes de recepción de servicio web/segundo

Solicitudes de publicación de servicio web/segundo

Bytes recibidos por segundo

Longitud de cola de mensajes normales por segundo

Longitud de cola de mensajes urgentes por segundo

Recuento de reconexiones

Longitud de cola de mensajes no reconocidos por segundo

Mensajes pendientes

Mensajes enviados por segundo

Mensajes de actualización de base de datos por segundo

Mensajes de actualización por segundo

Vaciados por segundo

Puntos de control criptográfico guardados por segundo

Puntos de control criptográfico restaurados por segundo

Puntos de control del registro restaurados por segundo

Puntos de control del registro guardados por segundo

Llamadas a la API de clústeres por segundo

Llamadas a la API de recursos por segundo

Controladores de clústeres por segundo

Controladores de recursos por segundo

Métricas de procesos del agente de CloudWatch (servidor de Windows)

Las métricas de procesos se recopilan mediante el complemento procstat del agente de CloudWatch. Solo las instancias de Amazon EC2 que ejecutan cargas de trabajo Windows admiten métricas de procesos.

procstat cpu_time_system

procstat cpu_time_user

procstat cpu_usage

procstat memory_rss

procstat memory_vms

procstat read_bytes

procstat write_bytes

.procstat read_count

procstat write_count

Métricas del agente de CloudWatch (servidor Linux)

cpu_time_active

cpu_time_guest

cpu_time_guest_nice

cpu_time_idle

cpu_time_iowait

cpu_time_irq

cpu_time_nice

cpu_time_softirq

cpu_time_steal

cpu_time_system

cpu_time_user

cpu_usage_active

cpu_usage_guest

cpu_usage_guest_nice

cpu_usage_idle

cpu_usage_iowait

cpu_usage_irq

cpu_usage_nice

cpu_usage_softirq

cpu_usage_steal

cpu_usage_system

cpu_usage_user

disk_free

disk_inodes_free

disk_inodes_used

disk_used

disk_used_percent

diskio_io_time

diskio_iops_in_progress

diskio_read_bytes

diskio_read_time

diskio_reads

diskio_write_bytes

diskio_write_time

diskio_writes

mem_active

mem_available

mem_available_percent

mem_buffered

mem_cached

mem_free

mem_inactive

mem_used

mem_used_percent

net_bytes_recv

net_bytes_sent

net_drop_in

net_drop_out

net_err_in

net_err_out

net_packets_recv

net_packets_sent

netstat_tcp_close

netstat_tcp_close_wait

netstat_tcp_closing

netstat_tcp_established

netstat_tcp_fin_wait1

netstat_tcp_fin_wait2

netstat_tcp_last_ack

netstat_tcp_listen

netstat_tcp_none

netstat_tcp_syn_recv

netstat_tcp_syn_sent

netstat_tcp_time_wait

netstat_udp_socket

processes_blocked

processes_dead

processes_idle

processes_paging

processes_running

processes_sleeping

processes_stopped

processes_total

processes_total_threads

processes_wait

processes_zombies

swap_free

swap_used

swap_used_percent

Elastic Block Store (EBS)

Información de aplicaciones de CloudWatch es compatible con las siguientes métricas:

VolumeReadBytes

VolumeWriteBytes

VolumeReadOps

VolumeWriteOps

VolumeTotalReadTime

VolumeTotalWriteTime

VolumeIdleTime

VolumeQueuelength

VolumeThroughputPercentage

VolumenConsumedReadWriteOps

BurstBalance

Amazon Elastic File System (Amazon EFS)

Información de aplicaciones de CloudWatch es compatible con las siguientes métricas:

BurstCreditBalance

PercentIOLimit

PermittedThroughput

MeteredIOBytes

TotalIOBytes

DataWriteIOBytes

DataReadIOBytes

MetadataIOBytes

ClientConnections

TimeSinceLastSync

StorageBytes

Rendimiento

PercentageOfPermittedThroughputUtilization

ThroughputIOPS

PercentThroughputDataReadIOByte

PercentThroughputDataWriteIOBytes

PercentageOfIOPSDataReadIOBytes

PercentageOfIOPSDataWriteIOBytes

AverageDataReadIOBytesSize

AverageDataWriteIOBytesSize

Elastic Load Balancer (ELB)

Información de aplicaciones de CloudWatch es compatible con las siguientes métricas:

EstimatedALBActiveConnectionCount

EstimatedALBConsumedLCUs

EstimatedALBNewConnectionCount

EstimatedProcessedBytes

HTTPCode_Backend_4XX

HTTPCode_Backend_5XX

HealthyHostCount

RequestCount

UnHealthyHostCount

Application ELB

Información de aplicaciones de CloudWatch es compatible con las siguientes métricas:

EstimatedALBActiveConnectionCount

EstimatedALBConsumedLCUs

EstimatedALBNewConnectionCount

EstimatedProcessedBytes

HTTPCode_Backend_4XX

HTTPCode_Backend_5XX

HealthyHostCount

Latencia

RequestCount

SurgeQueueLength

UnHealthyHostCount

Grupos de Amazon EC2 Auto Scaling

Información de aplicaciones de CloudWatch es compatible con las siguientes métricas:

CPUCreditBalance

CPUCreditUsage

CPUSurplusCreditBalance

CPUSurplusCreditsCharged

CPUUtilization

DiskReadBytes

DiskReadOps

DiskWriteBytes

DiskWriteOps

EBSByteBalance%

EBSIOBalance%

EBSReadBytes

EBSReadOps

EBSWriteBytes

EBSWriteOps

NetworkIn

NetworkOut

NetworkPacketsIn

NetworkPacketsOut

StatusCheckFailed

StatusCheckFailed_Instance

StatusCheckFailed_System

Amazon Simple Queue Server (SQS)

Información de aplicaciones de CloudWatch es compatible con las siguientes métricas:

ApproximateAgeOfOldestMessage

ApproximateNumberOfMessagesDelayed

ApproximateNumberOfMessagesNotVisible

ApproximateNumberOfMessagesVisible

NumberOfEmptyReceives

NumberOfMessagesDeleted

NumberOfMessagesReceived

NumberOfMessagesSent

Amazon Relational Database Service (RDS)

Información de aplicaciones de CloudWatch es compatible con las siguientes métricas:

Métricas

- Instancias de base de datos de RDS
- Clústeres de base de datos de RDS

Instancias de base de datos de RDS

BurstBalance

CPUCreditBalance

CPUUtilization

DatabaseConnections

DiskQueueDepth

FailedSQLServerAgentJobsCount

FreeStorageSpace

FreeableMemory

NetworkReceiveThroughput

NetworkTransmitThroughput

ReadIOPS

ReadLatency

ReadThroughput

WriteIOPS

WriteLatency

WriteThroughput

Clústeres de base de datos de RDS

ActiveTransactions

AuroraBinlogReplicaLag

AuroraReplicaLag

BackupRetentionPeriodStorageUsed

BinLogDiskUsage

BlockedTransactions

BufferCacheHitRatio

CPUUtilization

CommitLatency

CommitThroughput

DDLLatency

DDLThroughput

DMLLatency

DMLThroughput

DatabaseConnections

Interbloqueos

DeleteLatency

DeleteThroughput

EngineUptime

FreeLocalStorage

FreeableMemory

InsertLatency

InsertThroughput

LoginFailures

NetworkReceiveThroughput

NetworkThroughput

NetworkTransmitThroughput

Consultas

ResultSetCacheHitRatio

SelectLatency

SelectThroughput

SnapshotStorageUsed

TotalBackupStorageBilled

UpdateLatency

UpdateThroughput

VolumeBytesUsed

VolumeReadIOPs

VolumeWriteIOPs

Función AWS Lambda

Información de aplicaciones de CloudWatch es compatible con las siguientes métricas:

Errores

DeadLetterErrors

Duración

Limitaciones

IteratorAge

ProvisionedConcurrencySpilloverInvocations

Tabla de Amazon DynamoDB.

Información de aplicaciones de CloudWatch es compatible con las siguientes métricas:

SystemErrors

UserErrors

ConsumedReadCapacityUnits

ConsumedWriteCapacityUnits

ReadThrottleEvents

WriteThrottleEvents

TimeToLiveDeletedItemCount

CondicionalCheckFailedRequests

TransactionConflict

ReturnedRecordsCount

PendingReplicationCount

ReplicationLatency

Bucket de Amazon S3

Información de aplicaciones de CloudWatch es compatible con las siguientes métricas:

ReplicationLatency

BytesPendingReplication

OperationsPendingReplication

4xxErrors

5xxErrors

AllRequests

GetRequests

PutRequests

DeleteRequests

HeadRequests

PostRequests

SelectRequests

ListRequests

SelectScannedBytes

SelectReturnedBytes

FirstByteLatency

TotalRequestLatency

BytesDownloaded

BytesUploaded

AWS Step Functions

Información de aplicaciones de CloudWatch es compatible con las siguientes métricas:

Métricas

- Nivel de ejecución
- Actividad
- Función de Lambda
- Integración con los servicios

- API de Step Functions

Nivel de ejecución

ExecutionTime

ExecutionThrottled

ExecutionsFailed

ExecutionsTimedOut

ExecutionsAborted

ExecutionsSucceeded

ExecutionsStarted

Actividad

ActivityRunTime

ActivityScheduleTime

ActivityTime

ActivitiesFailed

ActivitiesHeartbeatTimedOut

ActivitiesTimedOut

ActivitiesScheduled

ActivitiesSucceeded

ActivitiesStarted

Función de Lambda

LambdaFunctionRuntime

LambdaFunctionScheduleTime

LambdaFunctionTime

LambdaFunctionsFailed

LambdaFunctionsTimedOut

LambdaFunctionsScheduled

LambdaFunctionsSucceeded

LambdaFunctionsStarted

Integración con los servicios

ServiceIntegrationRuntime

ServiceIntegrationScheduleTime

ServiceIntegrationTime

ServiceIntegrationsFailed

ServiceIntegrationsTimedOut

ServiceIntegrationsScheduled

ServiceIntegrationsSucceeded

ServiceIntegrationsStarted

API de Step Functions

ThrottledEvents

ProvisionedBucketSize

ProvisionedRefillRate

ConsumedCapacity

Etapas de la API de REST de API Gateway

Información de aplicaciones de CloudWatch es compatible con las siguientes métricas:

4XXError

5XXError

IntegrationLatency

Latencia

CacheHitCount

CacheMissCount

SAP HANA

> **Note**
>
> Información de aplicaciones de CloudWatch solo admite entornos SID HANA únicos. Si se adjuntan varios SID HANA, el monitoreo se configurará solo para el primer SID detectado.

CloudWatch Application Insights es compatible con las siguientes métricas:

hanadb_every_service_started_status

hanadb_daemon_service_started_status

hanadb_preprocessor_service_started_status

hanadb_webdispatcher_service_started_status

hanadb_compileserver_service_started_status

hanadb_nameserver_service_started_status

hanadb_server_startup_time_variations_seconds

hanadb_level_5_alerts_count

hanadb_level_4_alerts_count

hanadb_out_of_memory_events_count

hanadb_max_trigger_read_ratio_percent

hanadb_max_trigger_write_ratio_percent

hanadb_log_switch_wait_ratio_percent

hanadb_log_switch_race_ratio_percent

hanadb_time_since_last_savepoint_seconds

hanadb_disk_usage_highlevel_percent

hanadb_max_converter_page_number_count

hanadb_long_running_savepoints_count

hanadb_failed_io_reads_count

hanadb_failed_io_writes_count

hanadb_disk_data_unused_percent

hanadb_current_allocation_limit_used_percent

hanadb_table_allocation_limit_used_percent

hanadb_host_total_physical_memory_mb

hanadb_host_physical_memory_used_mb

hanadb_host_physical_memory_free_mb

hanadb_swap_memory_free_mb

hanadb_swap_memory_used_mb

hanadb_host_allocation_limit_mb

hanadb_host_total_memory_used_mb

hanadb_host_total_peak_memory_used_mb

hanadb_host_total_allocation_limit_mb

hanadb_host_code_size_mb

hanadb_host_shared_memory_allocation_mb

hanadb_cpu_usage_percent

hanadb_cpu_user_percent

hanadb_cpu_system_percent

hanadb_cpu_waitio_percent

hanadb_cpu_busy_percent

hanadb_cpu_idle_percent

hanadb_long_delta_merge_count

hanadb_unsuccesful_delta_merge_count

hanadb_succesful_delta_merge_count

hanadb_row_store_allocated_size_mb

hanadb_row_store_free_size_mb

hanadb_row_store_used_size_mb

hanadb_temporary_tables_count

hanadb_large_non_compressed_tables_count

hanadb_total_non_compressed_tables_count

hanadb_longest_running_job_seconds

hanadb_average_commit_time_milliseconds

hanadb_suspended_sql_statements_count

hanadb_plan_cache_hit_ratio_percent

hanadb_plan_cache_lookup_count

hanadb_plan_cache_hit_count

hanadb_plan_cache_total_execution_microseconds

hanadb_plan_cache_cursor_duration_microseconds

hanadb_plan_cache_preparation_microseconds

hanadb_plan_cache_evicted_count

hanadb_plan_cache_evicted_microseconds

hanadb_plan_cache_evicted_preparation_count

hanadb_plan_cache_evicted_execution_count

hanadb_plan_cache_evicted_preparation_microseconds

hanadb_plan_cache_evicted_cursor_duration_microseconds

hanadb_plan_cache_evicted_total_execution_microseconds

hanadb_plan_cache_evicted_plan_size_mb

hanadb_plan_cache_count

hanadb_plan_cache_preparation_count

hanadb_plan_cache_execution_count

hanadb_network_collision_rate

hanadb_network_receive_rate

hanadb_network_transmit_rate

hanadb_network_packet_receive_rate

hanadb_network_packet_transmit_rate

hanadb_network_transmit_error_rate

hanadb_network_receive_error_rate

hanadb_time_until_license_expires_days

hanadb_is_license_valid_status

hanadb_local_running_connections_count

hanadb_local_idle_connections_count

hanadb_remote_running_connections_count

hanadb_remote_idle_connections_count

hanadb_last_full_data_backup_age_days

hanadb_last_data_backup_age_days

hanadb_last_log_backup_age_hours

hanadb_failed_data_backup_past_7_days_count

hanadb_failed_log_backup_past_7_days_count

hanadb_oldest_backup_in_catalog_age_days

hanadb_backup_catalog_size_mb

hanadb_hsr_replication_status

hanadb_hsr_log_shipping_delay_seconds

hanadb_hsr_secondary_failover_count

hanadb_hsr_secondary_reconnect_count

hanadb_hsr_async_buffer_used_mb

hanadb_hsr_secondary_active_status

hanadb_handle_count

hanadb_ping_time_milliseconds

hanadb_connection_count

hanadb_internal_connection_count

hanadb_external_connection_count

hanadb_idle_connection_count

hanadb_transaction_count

hanadb_internal_transaction_count

hanadb_external_transaction_count

hanadb_user_transaction_count

hanadb_blocked_transaction_count

hanadb_statement_count

hanadb_active_commit_id_range_count

hanadb_mvcc_version_count

hanadb_pending_session_count

hanadb_record_lock_count

hanadb_read_count

hanadb_write_count

hanadb_merge_count

hanadb_unload_count

hanadb_active_thread_count

hanadb_waiting_thread_count

hanadb_total_thread_count

hanadb_active_sql_executor_count

hanadb_waiting_sql_executor_count

hanadb_total_sql_executor_count

hanadb_data_write_size_mb

hanadb_data_write_time_milliseconds

hanadb_log_write_size_mb

hanadb_log_write_time_milliseconds

hanadb_data_read_size_mb

hanadb_data_read_time_milliseconds

hanadb_log_read_size_mb

hanadb_log_read_time_milliseconds

hanadb_data_backup_write_size_mb

hanadb_data_backup_write_time_milliseconds

hanadb_log_backup_write_size_mb

hanadb_log_backup_write_time_milliseconds

hanadb_mutex_collision_count

hanadb_read_write_lock_collision_count

hanadb_admission_control_admit_count

hanadb_admission_control_reject_count

hanadb_admission_control_queue_size_mb

hanadb_admission_control_wait_time_milliseconds

SAP ASE

Información de aplicaciones de CloudWatch es compatible con las siguientes métricas:

asedb_database_availability

asedb_trunc_log_on_chkpt_enabled

asedb_last_db_backup_age_in_days

asedb_last_transaction_log_backup_age_in_hours

asedb_suspected_database

asedb_db_space_usage_percent

asedb_db_log_space_usage_percent

asedb_locked_login

asedb_has_mixed_log_and_data

asedb_runtime_for_open_transactions

asedb_data_cache_hit_ratio

asedb_data_cache_usage

asedb_sql_cache_hit_ratio

asedb_cache_usage

asedb_run_queue_length

asedb_number_of_rollbacks

asedb_number_of_commits

asedb_number_of_transactions

asedb_outstanding_disk_io

asedb_percent_io_busy

asedb_percent_system_busy

asedb_percent_locks_active

asedb_scheduled_jobs_failed_percent

asedb_user_connections_percent

asedb_query_logical_reads

asedb_query_physical_reads

asedb_query_cpu_time

asedb_query_memory_usage

Alta disponibilidad de SAP ASE en Amazon EC2

Información de aplicaciones de CloudWatch es compatible con las siguientes métricas:

asedb_ha_replication_state

asedb_ha_replication_mode

asedb_ha_replication_latency_in_minutes

SAP NetWeaver

Información de aplicaciones de CloudWatch es compatible con las siguientes métricas:

Métrica	Descripción
sap_alerts_ResponseTime	La alerta de tiempo de respuesta de SAP de CCMS (RZ20)>R3Services>Dialog>ResponseTime.
sap_alerts_ResponseTimeDialog	La alerta del diálogo de tiempo de respuesta de SAP de CCMS (RZ20)>R3Services>Dialog>ResponseTimeDialog.
sap_alerts_ResponseTimeDialogRFC	La alerta de tiempo de respuesta de SAP de CCMS (RZ20)>R3Services> Dialog>ResponseTimeDialogRFC.
sap_alerts_DBRequestTime	La alerta de tiempo de respuesta de SAP de CCMS (RZ20)>R3Services>Dialog>DBRequestTime.
sap_alerts_FrontendResponseTime	La alerta de tiempo de respuesta de SAP de CCMS (RZ20)>R3Services > Dialog>FrontEndResponseTime.
sap_alerts_Database	El sistema SAP ha registrado errores relacionados con la base de datos. Alerta desde SM21 o CCMS (RZ20)>R3Syslog>Database.
sap_alerts_QueueTime	La alerta de tiempo de espera de cola de SAP desde CCMS (RZ20)>R3Services>Dialog>QueueTime.
sap_alerts_AbortedJobs	Trabajos en segundo plano erróneos en el sistema SAP. Alerta desde (RZ20)>R3Services > Background>AbortedJobs.

Métrica	Descripción
sap_alerts_BasisSystem	El sistema SAP registró errores a nivel de sistema. Alerta desde SM21 o CCMS (RZ20)>R3Syslog>BasisSystem.
sap_alerts_Security	El sistema SAP registró mensajes relacionados con la seguridad. Alerta desde SM21 o CCMS (RZ20)>R3Syslog>Security.
sap_alerts_System	El sistema SAP registró mensajes relacionados con la seguridad o la auditoría. Alerta desde SM21 o CCMS (RZ20)>Security>System.
sap_alerts_LongRunners	Hay programas de larga ejecución en su sistema SAP. Alerta de CCMS (RZ20)>R3 Services > Dialog>LongRunners.
sap_alerts_SqlError	Hay registros de errores de la capa de cliente de la base de datos SAP. Alerta de CCMS(RZ20)>DatabaseClient>AbapSql>SqlError.
sap_alerts_State	Alerta de estado de CCMS (RZ20)>OS Collector>State.
sap_alerts_Shortdumps	Alerta de Shortdumps de ST22 y CCMS (RZ20)>R3Abap>Shortdumps.
sap_alerts_Availability	Alerta de disponibilidad para una instancia de servidor de aplicaciones SAP de SM21, SM50, SM51, SM66 y CCMS (RZ20)>InstanceAsTask>Availability.
sap_dispatcher_queue_high	La función `GetQueueStatistic` del servicio web de SAPControl proporciona un recuento alto de colas del distribuidor.

Métrica	Descripción
sap_dispatcher_queue_max	La función `GetQueueStatistic` del servicio web de SAPControl proporciona el recuento máximo de colas del distribuidor.
sap_dispatcher_queue_now	La función `GetQueueStatistic` del servicio web de SAPControl proporciona el recuento actualizado de colas del distribuidor.
sap_dispatcher_queue_reads	La función `GetQueueStatistic` del servicio web de SAPControl proporciona el recuento de lecturas de cola del distribuidor.
sap_dispatcher_queue_writes	La función `GetQueueStatistic` del servicio web de SAPControl proporciona el recuento de escrituras de cola del distribuidor.
sap_enqueue_server_arguments_high	La función `EnqGetStatistic` del servicio web de SAPControl proporciona el valor más alto de los argumentos en cola.
sap_enqueue_server_arguments_max	La función `EnqGetStatistic` del servicio web de SAPControl proporciona el valor máximo de los argumentos en cola.
sap_enqueue_server_arguments_now	La función `EnqGetStatistic` del servicio web de SAPControl proporciona los argumento s que haya actualmente en cola.
sap_enqueue_server_arguments_state	La función `EnqGetStatistic` del servicio web de SAPControl proporciona el estado de los argumentos en cola.
sap_enqueue_server_backup_requests	La función `EnqGetStatistic` del servicio web de SAPControl proporciona las solicitudes de copia de seguridad en cola.

Métrica	Descripción
sap_enqueue_server_cleanup_requests	La función `EnqGetStatistic` del servicio web de SAPControl proporciona las solicitudes de limpieza en cola.
sap_enqueue_server_dequeue_all_requests	La función `EnqGetStatistic` del servicio web de SAPControl proporciona todas las solicitudes que se han quitado de la cola.
sap_enqueue_server_dequeue_errors	La función `EnqGetStatistic` del servicio web de SAPControl proporciona todos los errores que se hayan quitado de la cola.
sap_enqueue_server_dequeue_requests	La función `EnqGetStatistic` del servicio web de SAPControl proporciona las solicitudes que se hayan quitado de la cola.
sap_enqueue_server_enqueue_errors	La función `EnqGetStatistic` del servicio web de SAPControl proporciona los errores en cola.
sap_enqueue_server_enqueue_rejects	La función `EnqGetStatistic` del servicio web de SAPControl proporciona los elementos rechazados en cola.
sap_enqueue_server_enqueue_requests	La función `EnqGetStatistic` del servicio web de SAPControl proporciona las solicitudes en cola.
sap_enqueue_server_lock_time	La función `EnqGetStatistic` del servicio web de SAPControl proporciona el tiempo de bloqueo en cola.
sap_enqueue_server_lock_wait_time	La función `EnqGetStatistic` del servicio web de SAPControl proporciona el tiempo de espera de bloqueo en cola.

Métrica	Descripción
sap_enqueue_server_locks_high	La función `EnqGetStatistic` del servicio web de SAPControl proporciona el valor más alto de los bloqueos en cola.
sap_enqueue_server_locks_max	La función `EnqGetStatistic` del servicio web de SAPControl proporciona el máximo de bloqueos en cola.
sap_enqueue_server_locks_now	La función `EnqGetStatistic` del servicio web de SAPControl proporciona los bloqueos actuales en cola.
sap_enqueue_server_locks_state	La función `EnqGetStatistic` del servicio web de SAPControl proporciona el estado de los bloqueos en cola.
sap_enqueue_server_owner_high	La función `EnqGetStatistic` del servicio web de SAPControl proporciona el valor más alto de propietarios en cola.
sap_enqueue_server_owner_max	La función `EnqGetStatistic` del servicio web de SAPControl proporciona el máximo de propietarios en cola.
sap_enqueue_server_owner_now	La función `EnqGetStatistic` del servicio web de SAPControl proporciona los propietar ios actuales en cola.
sap_enqueue_server_owner_state	La función `EnqGetStatistic` del servicio web de SAPControl proporciona estado de los propietarios en cola.
sap_enqueue_server_replication_state	La función `EnqGetStatistic` del servicio web de SAPControl proporciona estado de replicación en cola.

Métrica	Descripción
sap_enqueue_server_reporting_requests	La función `EnqGetStatistic` del servicio web de SAPControl proporciona el estado de las solicitudes de informes.
sap_enqueue_server_server_time	La función `EnqGetStatistic` del servicio web de SAPControl proporciona el tiempo del servidor en cola.
sap_HA_check_failover_config_state	La función `HACheckFailoverConfig` del servicio web de SAPControl proporciona el estado de alta disponibilidad de SAP.
sap_HA_get_failover_config_HAActive	La función `HAGetFailoverConfig` del servicio web de SAPControl proporciona el estado y configuración del clúster de alta disponibilidad de SAP.
sap_start_service_processes	La función `GetProcessList` del servicio web de SAPControl proporciona el estado de los procesos de disp+work, IGS, gwrd, icman, servidor de mensajes y servidor en cola.

Clúster de alta disponibilidad

Información de aplicaciones de CloudWatch es compatible con las siguientes métricas:

ha_cluster_pacemaker_stonith_enabled

ha_cluster_corosync_quorate

hanadb_webdispatcher_service_started_status

ha_cluster_pacemaker_nodes

ha_cluster_corosync_ring_errors

ha_cluster_pacemaker_fail_count

Java

Información de aplicaciones de CloudWatch es compatible con las siguientes métricas:

java_lang_memory_heapmemoryusage_used

java_lang_memory_heapmemoryusage_committed

java_lang_operatingsystem_openfiledescriptorcount

java_lang_operatingsystem_maxfiledescriptorcount

java_lang_operatingsystem_freephysicalmemorysize

java_lang_operatingsystem_freeswapspacesize

java_lang_threading_threadcount

java_lang_threading_daemonthreadcount

java_lang_classloading_loadedclasscount

java_lang_garbagecollector_collectiontime_copy

java_lang_garbagecollector_collectiontime_ps_scavenge

java_lang_garbagecollector_collectiontime_parnew

java_lang_garbagecollector_collectiontime_marksweepcompact

java_lang_garbagecollector_collectiontime_ps_marksweep

java_lang_garbagecollector_collectiontime_concurrentmarksweep

java_lang_garbagecollector_collectiontime_g1_young_generation

java_lang_garbagecollector_collectiontime_g1_old_generation

java_lang_garbagecollector_collectiontime_g1_mixed_generation

java_lang_operatingsystem_committedvirtualmemorysize

Amazon Elastic Container Service (Amazon ECS)

Información de aplicaciones de CloudWatch es compatible con las siguientes métricas:

Métricas

- Métricas integradas de CloudWatch
- Métricas de Información de contenedores
- Métricas de Prometheus de Información de contenedores

Métricas integradas de CloudWatch

CPUReservation

CPUUtilization

MemoryReservation

MemoryUtilization

GPUReservation

Métricas de Información de contenedores

ContainerInstanceCount

CpuUtilized

CpuReserved

DeploymentCount

DesiredTaskCount

MemoryUtilized

MemoryReserved

NetworkRxBytes

NetworkTxBytes

PendingTaskCount

RunningTaskCount

ServiceCount

StorageReadBytes

StorageWriteBytes

TaskCount

TaskSetCount

instance_cpu_limit

instance_cpu_reserved_capacity

instance_cpu_usage_total

instance_cpu_utilization

instance_filesystem_utilization

instance_memory_limit

instance_memory_reserved_capacity

instance_memory_utilization

instance_memory_working_set

instance_network_total_bytes

instance_number_of_running_tasks

Métricas de Prometheus de Información de contenedores

Métricas de Java JMX

java_lang_memory_heapmemoryusage_used

java_lang_memory_heapmemoryusage_committed

java_lang_operatingsystem_openfiledescriptorcount

java_lang_operatingsystem_maxfiledescriptorcount

java_lang_operatingsystem_freephysicalmemorysize

java_lang_operatingsystem_freeswapspacesize

java_lang_threading_threadcount

java_lang_classloading_loadedclasscount

java_lang_threading_daemonthreadcount

java_lang_garbagecollector_collectiontime_copy

java_lang_garbagecollector_collectiontime_ps_scavenge

java_lang_garbagecollector_collectiontime_parnew

java_lang_garbagecollector_collectiontime_marksweepcompact

java_lang_garbagecollector_collectiontime_ps_marksweep

java_lang_garbagecollector_collectiontime_concurrentmarksweep

java_lang_garbagecollector_collectiontime_g1_young_generation

java_lang_garbagecollector_collectiontime_g1_old_generation

java_lang_garbagecollector_collectiontime_g1_mixed_generation

java_lang_operatingsystem_committedvirtualmemorysize

Kubernetes en AWS

Información de aplicaciones de CloudWatch es compatible con las siguientes métricas:

Métricas

- Métricas de Información de contenedores
- Métricas de Prometheus de Información de contenedores

Métricas de Información de contenedores

cluster_failed_node_count

cluster_node_count

namespace_number_of_running_pods

node_cpu_limit

node_cpu_reserved_capacity

node_cpu_usage_total

node_cpu_utilization

node_filesystem_utilization

node_memory_limit

node_memory_reserved_capacity

node_memory_utilization

node_memory_working_set

node_network_total_bytes

node_number_of_running_containers

node_number_of_running_pods

pod_cpu_reserved_capacity

pod_cpu_utilization

pod_cpu_utilization_over_pod_limit

pod_memory_reserved_capacity

pod_memory_utilization

pod_memory_utilization_over_pod_limit

pod_network_rx_bytes

pod_network_tx_bytes

service_number_of_running_pods

Métricas de Prometheus de Información de contenedores

Métricas de Java JMX

java_lang_memory_heapmemoryusage_used

java_lang_memory_heapmemoryusage_committed

java_lang_operatingsystem_openfiledescriptorcount

java_lang_operatingsystem_maxfiledescriptorcount

java_lang_operatingsystem_freephysicalmemorysize

java_lang_operatingsystem_freeswapspacesize

java_lang_threading_threadcount

java_lang_classloading_loadedclasscount

java_lang_threading_daemonthreadcount

java_lang_garbagecollector_collectiontime_copy

java_lang_garbagecollector_collectiontime_ps_scavenge

java_lang_garbagecollector_collectiontime_parnew

java_lang_garbagecollector_collectiontime_marksweepcompact

java_lang_garbagecollector_collectiontime_ps_marksweep

java_lang_garbagecollector_collectiontime_concurrentmarksweep

java_lang_garbagecollector_collectiontime_g1_young_generation

java_lang_garbagecollector_collectiontime_g1_old_generation

java_lang_garbagecollector_collectiontime_g1_mixed_generation

java_lang_operatingsystem_committedvirtualmemorysize

Amazon FSx

Información de aplicaciones de CloudWatch es compatible con las siguientes métricas:

DataReadBytes

DataWriteBytes

DataReadOperations

DataWriteOperations

MetadataOperations

FreeStorageCapacity

FreeDataStorageCapacity

LogicalDiskUsage

PhysicalDiskUsage

Amazon VPC

Información de aplicaciones de CloudWatch es compatible con las siguientes métricas:

NetworkAddressUsage

NetworkAddressUsagePeered

VPCFirewallQueryVolume

Puerta de enlace NAT de Amazon VPC

Información de aplicaciones de CloudWatch es compatible con las siguientes métricas:

ErrorPortAllocation

IdleTimeoutCount

Comprobación de estado de Amazon Route 53

Información de aplicaciones de CloudWatch es compatible con las siguientes métricas:

ChildHealthCheckHealthyCount

ConnectionTime

HealthCheckPercentageHealthy

HealthCheckStatus

SSLHandshakeTime

TimeToFirstByte

Zona alojada de Amazon Route 53

Información de aplicaciones de CloudWatch es compatible con las siguientes métricas:

DNSQueries

DNSSECInternalFailure

DNSSECKeySigningKeysNeedingAction

DNSSECKeySigningKeyMaxNeedingActionAge

DNSSECKeySigningKeyAge

Punto de conexión de Amazon Route 53 Resolver

Información de aplicaciones de CloudWatch es compatible con las siguientes métricas:

EndpointHealthyENICount

EndpointUnHealthyENICount

InboundQueryVolume

OutboundQueryVolume

OutboundQueryAggregateVolume

Grupo de reglas de AWS Network Firewall

Información de aplicaciones de CloudWatch es compatible con las siguientes métricas:

FirewallRuleGroupQueryVolume

Asociación de grupos de reglas AWS Network Firewall

Información de aplicaciones de CloudWatch es compatible con las siguientes métricas:

FirewallRuleGroupVpcQueryVolume

Métricas con requisitos de puntos de datos

Para las métricas sin un umbral predeterminado evidente al partir del cual activar una alarma, Información de aplicaciones espera hasta que la métrica tiene puntos de datos suficientes para predecir un umbral razonable para activar una alarma. Los requisitos de los puntos de datos de la métrica que Información de aplicaciones de CloudWatch verifica antes de crear una alarma son los siguientes:

- La métrica tiene al menos 100 puntos de datos desde los últimos 15 a los últimos 2 días.
- La métrica tiene al menos 100 puntos de datos del último día.

Las métricas siguientes cumplen estos requisitos de puntos de datos. Tenga en cuenta que las métricas del agente de CloudWatch requieren hasta una hora para crear alarmas.

Métricas

- AWS/ApplicationELB
- AWS/AutoScaling
- AWS/EC2
- Elastic Block Store (EBS)
- AWS/ELB
- AWS/RDS
- AWS/Lambda

- AWS/SQS
- AWS/CWAgent
- AWS/DynamoDB
- AWS/S3
- Estados de AWS
- ApiGateway de AWS
- AWS/SNS

AWS/ApplicationELB

ActiveConnectionCount

ConsumedLCUs

HTTPCode_ELB_4XX_Count

HTTPCode_Target_2XX_Count

HTTPCode_Target_3XX_Count

HTTPCode_Target_4XX_Count

HTTPCode_Target_5XX_Count

NewConnectionCount

ProcessedBytes

TargetResponseTime

UnHealthyHostCount

AWS/AutoScaling

GroupDesiredCapacity

GroupInServiceInstances

GroupMaxSize

GroupMinSize

GroupPendingInstances

GroupStandbyInstances

GroupTerminatingInstances

GroupTotalInstances

AWS/EC2

CPUCreditBalance

CPUCreditUsage

CPUSurplusCreditBalance

CPUSurplusCreditsCharged

CPUUtilization

DiskReadBytes

DiskReadOps

DiskWriteBytes

DiskWriteOps

EBSByteBalance%

EBSIOBalance%

EBSReadBytes

EBSReadOps

EBSWriteBytes

EBSWriteOps

NetworkIn

NetworkOut

NetworkPacketsIn

NetworkPacketsOut

Elastic Block Store (EBS)

VolumeReadBytes

VolumeWriteBytes

VolumeReadOps

VolumeWriteOps

VolumeTotalReadTime

VolumeTotalWriteTime

VolumeIdleTime

VolumeQueuelength

VolumeThroughputPercentage

VolumenConsumedReadWriteOps

BurstBalance

AWS/ELB

EstimatedALBActiveConnectionCount

EstimatedALBConsumedLCUs

EstimatedALBNewConnectionCount

EstimatedProcessedBytes

HTTPCode_Backend_4XX

HTTPCode_Backend_5XX

HealthyHostCount

Latencia

RequestCount

SurgeQueueLength

UnHealthyHostCount

AWS/RDS

ActiveTransactions

AuroraBinlogReplicaLag

AuroraReplicaLag

BackupRetentionPeriodStorageUsed

BinLogDiskUsage

BlockedTransactions

CPUCreditBalance

CommitLatency

CommitThroughput

DDLLatency

DDLThroughput

DMLLatency

DMLThroughput

DatabaseConnections

Interbloqueos

DeleteLatency

DeleteThroughput

DiskQueueDepth

EngineUptime

FreeLocalStorage

FreeStorageSpace

FreeableMemory

InsertLatency

InsertThroughput

LoginFailures

NetworkReceiveThroughput

NetworkThroughput

NetworkTransmitThroughput

Consultas

ReadIOPS

ReadThroughput

SelectLatency

SelectThroughput

SnapshotStorageUsed

TotalBackupStorageBilled

UpdateLatency

UpdateThroughput

VolumeBytesUsed

VolumeReadIOPs

VolumeWriteIOPs

WriteIOPS

WriteThroughput

AWS/Lambda

Errores

DeadLetterErrors

Duración

Limitaciones

IteratorAge

ProvisionedConcurrencySpilloverInvocations

AWS/SQS

ApproximateAgeOfOldestMessage

ApproximateNumberOfMessagesDelayed

ApproximateNumberOfMessagesNotVisible

ApproximateNumberOfMessagesVisible

NumberOfEmptyReceives

NumberOfMessagesDeleted

NumberOfMessagesReceived

NumberOfMessagesSent

AWS/CWAgent

Porcentaje de espacio libre en disco lógico

Porcentaje de bytes confirmados en uso en memoria

Mbytes disponibles de memoria

Total de bytes de interfaz de red/segundo

Porcentaje de uso de archivo de paginación

Porcentaje de tiempo de disco en disco físico

Promedio de disco físico Lectura de disco/segundo

Promedio de disco físico Escritura en disco/segundo

Bytes leídos/segundo en disco físico

Lecturas/segundo en disco físico

Bytes de escritura/segundo en disco físico

Escrituras/segundo en disco físico

Porcentaje de tiempo de inactividad del procesador

Porcentaje de horas de interrupción del procesador

Porcentaje de tiempo de procesador del procesador

Porcentaje de tiempo de usuario del procesador

SQLServer: métodos de acceso: registros reenviados/segundo

SQLServer: métodos de acceso: divisiones de página/segundo

SQLServer: administrador del búfer: porcentaje de aciertos de caché del búfer

SQLServer: administrador del búfer: duración prevista de página

SQLServer: bytes recibidos por segundo del archivo de réplica de base de datos

SQLServer: bytes recibidos por segundo del registro de réplica de base de datos

SQLServer: restante para deshacer del registro de réplica de base de datos

SQLServer: cola de envío de registros de réplica de base de datos

SQLServer: transacción por segundo de escritura reflejada de réplica de base de datos

SQLServer: cola de recuperación de réplica de base de datos

SQLServer: resto de bytes de rehacer de réplica de base de datos

SQLServer: bytes por segundo rehechos de réplica de base de datos

SQLServer: registro total de réplica de base de datos que requiere deshacer

SQLServer: retraso de transacción de réplica de base de datos

SQLServer: estadísticas generales: procesos bloqueados

SQLServer: estadísticas de SQL: solicitudes por lotes/segundo

SQLServer: estadísticas de SQL: compilaciones de SQL/segundo

SQLServer: estadísticas de SQL: recopilaciones de SQL/segundo

Longitud de cola del procesador del sistema

Conexiones TCPv4 establecidas

Conexiones TCPv6 establecidas

AWS/DynamoDB

ConsumedReadCapacityUnits

ConsumedWriteCapacityUnits

ReadThrottleEvents

WriteThrottleEvents

TimeToLiveDeletedItemCount

CondicionalCheckFailedRequests

TransactionConflict

ReturnedRecordsCount

PendingReplicationCount

ReplicationLatency

AWS/S3

ReplicationLatency

BytesPendingReplication

OperationsPendingReplication

4xxErrors

5xxErrors

AllRequests

GetRequests

PutRequests

DeleteRequests

HeadRequests

PostRequests

SelectRequests

ListRequests

SelectScannedBytes

SelectReturnedBytes

FirstByteLatency

TotalRequestLatency

BytesDownloaded

BytesUploaded

Estados de AWS

ActivitiesScheduled

ActivitiesStarted

ActivitiesSucceeded

ActivityScheduleTime

ActivityRuntime

ActivityTime

LambdaFunctionsScheduled

LambdaFunctionsStarted

LambdaFunctionsSucceeded

LambdaFunctionScheduleTime

LambdaFunctionRuntime

LambdaFunctionTime

ServiceIntegrationsScheduled

ServiceIntegrationsStarted

ServiceIntegrationsSucceeded

ServiceIntegrationScheduleTime

ServiceIntegrationRuntime

ServiceIntegrationTime

ProvisionedRefillRate

ProvisionedBucketSize

ConsumedCapacity

ThrottledEvents

ApiGateway de AWS

4XXError

IntegrationLatency

Latencia

Datos procesados

CacheHitCount

CacheMissCount

AWS/SNS

NumberOfNotificationsDelivered

NumberOfMessagesPublished

NumberOfNotificationsFailed

NumberOfNotificationsFilteredOut

NumberOfNotificationsFilteredOut-InvalidAttributes

NumberOfNotificationsFilteredOut-NoMessageAttributes

NumberOfNotificationsRedrivenToDlq

NumberOfNotificationsFailedToRedriveToDlq

SMSSuccessRate

Métricas recomendadas

En la siguiente tabla se indican las métricas recomendadas para cada tipo de componente.

Tipo de componente	Tipo de carga de trabajo	Métrica que se sugiere
Instancia EC2 (servidores Windows)	Valor predeterminado/Personalizado	CPUUtilization StatusCheckFailed Porcentaje de tiempo de procesador del procesador Porcentaje de bytes confirmados en uso en memoria Porcentaje de espacio libre en disco lógico Mbytes disponibles de memoria
	Active Directory	CPUUtilization StatusCheckFailed Porcentaje de tiempo de procesador del procesador Porcentaje de bytes confirmados en uso en memoria Mbytes disponibles de memoria Base de datos ==> Porcentaje de aciertos de la caché de la base de datos de instancias Operaciones de replicación pendientes de DRA de DirectoryServices

Tipo de componente	Tipo de carga de trabajo	Métrica que se sugiere
		Sincronizaciones de replicación pendientes de DRA de DirectoryServices Error de consulta recursiva de DNS por segundo Promedio LogicalDisk Longitud de la cola de disco

Tipo de componente	Tipo de carga de trabajo	Métrica que se sugiere
	Aplicación Java	CPUUtilization StatusCheckFailed Porcentaje de tiempo de procesador del procesador Porcentaje de bytes confirmados en uso en memoria Mbytes disponibles de memoria java_lang_threading_threadcount java_lang_classloading_loadedclasscount java_lang_memory_heapmemoryusage_used java_lang_memory_heapmemoryusage_committed java_lang_operatingsystem_freephysicalmemorysize java_lang_operatingsystem_freeswapspacesize

Tipo de componente	Tipo de carga de trabajo	Métrica que se sugiere
	Servidor front-end web de Microsoft IIS/.NET	CPUUtilization StatusCheckFailed Porcentaje de tiempo de procesador del procesador Porcentaje de bytes confirmados en uso en memoria Mbytes disponibles de memoria .NET CLR Exceptions: n.º de excepciones generadas/segundo .NET CLR Memory: n.º de bytes confirmados totales Tiempo de % de memoria de CLR de .NET en GC Solicitudes de aplicaciones ASP.NET en la cola de aplicaciones Solicitudes ASP.NET en cola Reinicios de aplicaciones ASP.NET

Tipo de componente	Tipo de carga de trabajo	Métrica que se sugiere
	Capa de base de datos de Microsoft SQL Server	CPUUtilization StatusCheckFailed Porcentaje de tiempo de procesador del procesador Porcentaje de bytes confirmados en uso en memoria Mbytes disponibles de memoria Porcentaje de uso de archivo de paginación Longitud de cola del procesador del sistema Total de bytes de interfaz de red/segundo Porcentaje de tiempo de disco en disco físico SQLServer: administrador del búfer: porcentaje de aciertos de caché del búfer SQLServer: administrador del búfer: duración prevista de página SQLServer: estadísticas generales: procesos bloqueados

Tipo de componente	Tipo de carga de trabajo	Métrica que se sugiere
		SQLServer: estadísticas generales: conexiones de usuario SQLServer: bloqueos: número de interbloqueos/segundo SQLServer: estadísticas de SQL: solicitudes por lotes/segundo
	MySQL	CPUUtilization StatusCheckFailed Porcentaje de tiempo de procesador del procesador Porcentaje de bytes confirmados en uso en memoria Porcentaje de espacio libre en disco lógico Mbytes disponibles de memoria

Tipo de componente	Tipo de carga de trabajo	Métrica que se sugiere
	.NET workerpool/capa intermedia	CPUUtilization StatusCheckFailed Porcentaje de tiempo de procesador del procesador Porcentaje de bytes confirmados en uso en memoria Mbytes disponibles de memoria .NET CLR Exceptions: n.º de excepciones generadas/segundo .NET CLR Memory: n.º de bytes confirmados totales Tiempo de % de memoria de CLR de .NET en GC
	Capa básica de .NET	CPUUtilization StatusCheckFailed Porcentaje de tiempo de procesador del procesador Porcentaje de bytes confirmados en uso en memoria Mbytes disponibles de memoria

Tipo de componente	Tipo de carga de trabajo	Métrica que se sugiere
	Oracle	CPUUtilization StatusCheckFailed Porcentaje de tiempo de procesador del procesador Porcentaje de bytes confirmados en uso en memoria Porcentaje de espacio libre en disco lógico Mbytes disponibles de memoria
	Postgres	CPUUtilization StatusCheckFailed Porcentaje de tiempo de procesador del procesador Porcentaje de bytes confirmados en uso en memoria Porcentaje de espacio libre en disco lógico Mbytes disponibles de memoria

Tipo de componente	Tipo de carga de trabajo	Métrica que se sugiere
	SharePoint	CPUUtilization StatusCheckFailed Porcentaje de tiempo de procesador del procesador Porcentaje de bytes confirmados en uso en memoria Mbytes disponibles de memoria Trims de API de la caché de aplicaciones ASP.NET Solicitudes ASP.NET rechazadas Reinicios del proceso de trabajo ASP.NET Páginas de memoria/segundo Vaciados de caché de publicación de la caché de publicación de SharePoint por segundo Solicitud de página/tiempo de ejecución de SharePoint Foundation Número total de compactaciones de la caché basada en disco de SharePoint

Tipo de componente	Tipo de carga de trabajo	Métrica que se sugiere
		Relación de aciertos de caché de blob de la caché basada en disco de SharePoint Relación de llenado de caché de blob de la caché basada en disco de SharePoint Vaciados de caché de blob de la caché basada en disco de SharePoint por segundo Solicitudes ASP.NET en cola Solicitudes de aplicacio nes ASP.NET en la cola de aplicaciones Reinicios de aplicaciones ASP.NET Promedio LogicalDisk Escritura en disco/segundo Promedio LogicalDisk Lectura de disco/segundo Porcentaje de horas de interrupción del procesador
Instancia EC2 (servidores Linux)	Valor predeterminado/Per sonalizado	CPUUtilization StatusCheckFailed disk_used_percent mem_used_percent

Tipo de componente	Tipo de carga de trabajo	Métrica que se sugiere
	Aplicación Java	CPUUtilization StatusCheckFailed disk_used_percent mem_used_percent java_lang_threading_threadcount java_lang_classloading_loadedclasscount java_lang_memory_heapmemoryusage_used java_lang_memory_heapmemoryusage_committed java_lang_operatingsystem_freephysicalmemorysize java_lang_operatingsystem_freeswapspacesize
	Capa básica de.NET o capa de base de datos SQL Server	CPUUtilization StatusCheckFailed disk_used_percent mem_used_percent

Tipo de componente	Tipo de carga de trabajo	Métrica que se sugiere
	Oracle	CPUUtilization StatusCheckFailed disk_used_percent mem_used_percent
	Postgres	CPUUtilization StatusCheckFailed disk_used_percent mem_used_percent

Tipo de componente	Tipo de carga de trabajo	Métrica que se sugiere
Grupo de instancias EC2	SAP HANA de varios nodos o de un solo nodo	• hanadb_server_startup_time_variations_seconds • hanadb_level_5_alerts_count • hanadb_level_4_alerts_count • hanadb_out_of_memory_events_count • hanadb_max_trigger_read_ratio_percent • hanadb_max_trigger_write_ratio_percent • hanadb_log_switch_race_ratio_percent • hanadb_time_since_last_savepoint_seconds • hanadb_disk_usage_highlevel_percent • hanadb_current_allocation_limit_used_percent • hanadb_table_allocation_limit_used_percent • hanadb_cpu_usage_percent • hanadb_plan_cache_hit_ratio_percent • hanadb_last_data_backup_age_days

Tipo de componente	Tipo de carga de trabajo	Métrica que se sugiere
Volumen de EBS	Cualquiera	VolumeReadBytes VolumeWriteBytes VolumeReadOps VolumeWriteOps VolumeQueueLength VolumeThroughputPercentage VolumenConsumedReadWriteOps BurstBalance
Classic ELB	Cualquiera	HTTPCode_Backend_4XX HTTPCode_Backend_5XX Latencia SurgeQueueLength UnHealthyHostCount
Application ELB	Cualquiera	HTTPCode_Target_4XX_Count HTTPCode_Target_5XX_Count TargetResponseTime UnHealthyHostCount

Tipo de componente	Tipo de carga de trabajo	Métrica que se sugiere
Instancia de base de datos de RDS	Cualquiera	CPUUtilization ReadLatency WriteLatency BurstBalance FailedSQLServerAgentJobsCount
Clúster de base de datos de RDS	Cualquiera	CPUUtilization CommitLatency DatabaseConnections Interbloqueos FreeableMemory NetworkThroughput VolumeBytesUsed
Función Lambda	Cualquiera	Duración Errores IteratorAge ProvisionedConcurrencySpilloverInvocations Limitaciones

Tipo de componente	Tipo de carga de trabajo	Métrica que se sugiere
Cola de SQS	Cualquiera	ApproximateAgeOfOldestMessage ApproximateNumberOfMessagesVisible NumberOfMessagesSent
Tabla de Amazon DynamoDB.	Cualquiera	SystemErrors UserErrors ConsumedReadCapacityUnits ConsumedWriteCapacityUnits ReadThrottleEvents WriteThrottleEvents CondicionalCheckFailedRequests TransactionConflict

Tipo de componente	Tipo de carga de trabajo	Métrica que se sugiere
Bucket de Amazon S3	Cualquiera	Si la configuración de replicación con Control del Tiempo de Replicación (RTC) está habilitada: ReplicationLatency BytesPendingReplication OperationsPendingReplication Si las métricas de solicitud están activadas: 5xxErrors 4xxErrors BytesDownloaded BytesUploaded

Tipo de componente	Tipo de carga de trabajo	Métrica que se sugiere
AWS Step Functions	Cualquiera	General • ExecutionThrottled • ExecutionsAborted • ProvisionedBucketSize • ProvisionedRefillRate • ConsumedCapacity Si el tipo de máquina de estado es **EXPRESS** o el nivel de grupo de registro es **OFF** • ExecutionsFailed • ExecutionsTimedOut Si la máquina de estado tiene funciones de Lambda • LambdaFunctionsFailed • LambdaFunctionsTimedOut Si la máquina de estado tiene actividades • ActivitiesFailed • ActivitiesTimedOut • ActivitiesHeartbea tTimedOut Si la máquina de estado tiene integraciones de servicio • ServiceIntegrationsFailed

Tipo de componente	Tipo de carga de trabajo	Métrica que se sugiere
		• ServiceIntegration sTimedOut
Estado de la API REST de API Gateway	Cualquiera	• 4XXErrors • 5XXErrors • Latencia

Tipo de componente	Tipo de carga de trabajo	Métrica que se sugiere
Clúster de ECS	Cualquiera	CpuUtilized MemoryUtilized NetworkRxBytes NetworkTxBytes RunningTaskCount PendingTaskCount StorageReadBytes StorageWriteBytes CPUReservation (solo tipo de lanzamiento de EC2) CPUUtilization (solo tipo de lanzamiento de EC2) MemoryReservation (solo tipo de lanzamiento de EC2) MemoryUtilization (solo tipo de lanzamiento de EC2) GPUReservation (solo tipo de lanzamiento de EC2) instance_cpu_utilization (sólo tipo de lanzamiento de EC2) instance_filesystem_utilization (sólo tipo de lanzamiento de EC2)

Tipo de componente	Tipo de carga de trabajo	Métrica que se sugiere
		instance_memory_utilization (sólo tipo de lanzamiento de EC2) instance_network_total_bytes (sólo tipo de lanzamiento de EC2)

Tipo de componente	Tipo de carga de trabajo	Métrica que se sugiere
	Aplicación de Java	CpuUtilized MemoryUtilized NetworkRxBytes NetworkTxBytes RunningTaskCount PendingTaskCount StorageReadBytes StorageWriteBytes CPUReservation (solo tipo de lanzamiento de EC2) CPUUtilization (solo tipo de lanzamiento de EC2) MemoryReservation (solo tipo de lanzamiento de EC2) MemoryUtilization (solo tipo de lanzamiento de EC2) GPUReservation (solo tipo de lanzamiento de EC2) instance_cpu_utilization (sólo tipo de lanzamiento de EC2) instance_filesystem_utilization (sólo tipo de lanzamiento de EC2)

Tipo de componente	Tipo de carga de trabajo	Métrica que se sugiere
		instance_memory_utilization (sólo tipo de lanzamiento de EC2) instance_network_total_bytes (sólo tipo de lanzamiento de EC2) java_lang_threading_threadcount java_lang_classloading_loadedclasscount java_lang_memory_heapmemoryusage_used java_lang_memory_heapmemoryusage_committed java_lang_operatingsystem_freephysicalmemorysize java_lang_operatingsystem_freeswapspacesize

Tipo de componente	Tipo de carga de trabajo	Métrica que se sugiere
servicio de ECS	Cualquiera	CPUUtilization MemoryUtilization CpuUtilized MemoryUtilized NetworkRxBytes NetworkTxBytes RunningTaskCount PendingTaskCount StorageReadBytes StorageWriteBytes

Tipo de componente	Tipo de carga de trabajo	Métrica que se sugiere
	Aplicación de Java	CPUUtilization MemoryUtilization CpuUtilized MemoryUtilized NetworkRxBytes NetworkTxBytes RunningTaskCount PendingTaskCount StorageReadBytes StorageWriteBytes java_lang_threading_threadcount java_lang_classloading_loadedclasscount java_lang_memory_heapmemoryusage_used java_lang_memory_heapmemoryusage_committed java_lang_operatingsystem_freephysicalmemorysize java_lang_operatingsystem_freeswapspacesize

Tipo de componente	Tipo de carga de trabajo	Métrica que se sugiere
Clúster de EKS	Cualquiera	cluster_failed_node_count
		node_cpu_reserved_capacity
		node_cpu_utilization
		node_filesystem_utilization
		node_memory_reserv ed_capacity
		node_memory_utilization
		node_network_total_bytes
		pod_cpu_reserved_capacity
		pod_cpu_utilization
		pod_cpu_utilization_over_po d_limit
		pod_memory_reserve d_capacity
		pod_memory_utilization
		pod_memory_utilization_over _pod_limit
		pod_network_rx_bytes
		pod_network_tx_bytes

Tipo de componente	Tipo de carga de trabajo	Métrica que se sugiere
	Aplicación Java	cluster_failed_node_count node_cpu_reserved_capacity node_cpu_utilization node_filesystem_utilization node_memory_reserved_capacity node_memory_utilization node_network_total_bytes pod_cpu_reserved_capacity pod_cpu_utilization pod_cpu_utilization_over_pod_limit pod_memory_reserved_capacity pod_memory_utilization pod_memory_utilization_over_pod_limit pod_network_rx_bytes pod_network_tx_bytes java_lang_threading_threadcount java_lang_classloading_loadedclasscount

Tipo de componente	Tipo de carga de trabajo	Métrica que se sugiere
		java_lang_memory_heapmemoryusage_used java_lang_memory_heapmemoryusage_committed java_lang_operatingsystem_freephysicalmemorysize java_lang_operatingsystem_freeswapspacesize

Tipo de componente	Tipo de carga de trabajo	Métrica que se sugiere
Clúster de Kubernetes en EC2	Cualquiera	cluster_failed_node_count
		node_cpu_reserved_capacity
		node_cpu_utilization
		node_filesystem_utilization
		node_memory_reserv ed_capacity
		node_memory_utilization
		node_network_total_bytes
		pod_cpu_reserved_capacity
		pod_cpu_utilization
		pod_cpu_utilization_over_po d_limit
		pod_memory_reserve d_capacity
		pod_memory_utilization
		pod_memory_utilization_over _pod_limit
		pod_network_rx_bytes
		pod_network_tx_bytes

Tipo de componente	Tipo de carga de trabajo	Métrica que se sugiere
	Aplicación Java	cluster_failed_node_count node_cpu_reserved_capacity node_cpu_utilization node_filesystem_utilization node_memory_reserved_capacity node_memory_utilization node_network_total_bytes pod_cpu_reserved_capacity pod_cpu_utilization pod_cpu_utilization_over_pod_limit pod_memory_reserved_capacity pod_memory_utilization pod_memory_utilization_over_pod_limit pod_network_rx_bytes pod_network_tx_bytes java_lang_threading_threadcount java_lang_classloading_loadedclasscount

Tipo de componente	Tipo de carga de trabajo	Métrica que se sugiere
		java_lang_memory_heapmemoryusage_used java_lang_memory_heapmemoryusage_committed java_lang_operatingsystem_freephysicalmemorysize java_lang_operatingsystem_freeswapspacesize

En la siguiente tabla se enumeran los procesos recomendados y las métricas del proceso para cada tipo de componente. Información de aplicaciones de CloudWatch no recomienda la monitorización de procesos para aquellos que no se ejecutan en una instancia.

Tipo de componente	Tipo de carga de trabajo	Proceso recomendado	Métrica que se sugiere

Instancia EC2 (servidores Windows)	Servidor front-end web de Microsoft IIS/.NET	`w3wp`	`procstat cpu_usage ,`
			
			`procstat memory_rss ,`
			
			`procstat memory_vms ,`
			
			`procstat read_bytes ,`
			
			`procstat write_bytes`

	Capa de base de datos de Microsoft SQL Server	`SQLAgent`	`procstat cpu_usage ,`

Tipo de componente	Tipo de carga de trabajo	Proceso recomendado	Métrica que se sugiere
			procstat memory_rss , procstat memory_vms , procstat read_bytes , procstat write_bytes
		sqlservr	procstat cpu_usage , procstat memory_rss , procstat memory_vms , procstat read_bytes , procstat write_bytes
		sqlwriter	procstat cpu_usage , procstat memory_rss

Tipo de componente	Tipo de carga de trabajo	Proceso recomendado	Métrica que se sugiere
		`Reporting ServicesService`	`procstat cpu_usage ,` `procstat memory_rss`

		`MsDtsServr`	`procstat cpu_usage ,`
			`procstat memory_rss ,`
			`procstat memory_vms ,`
			`procstat read_bytes ,`
			`procstat write_bytes`

		`Msmdsrv`	`procstat cpu_usage ,`
			`procstat memory_rss ,`
			`procstat memory_vms ,`
			`procstat read_bytes ,`
			`procstat write_bytes`

Tipo de componente	Tipo de carga de trabajo	Proceso recomendado	Métrica que se sugiere
	.NET workerpool/capa intermedia	`w3wp`	`procstat cpu_usage` , `procstat memory_rss` , `procstat memory_vms` , `procstat read_bytes` , `procstat write_bytes`
	Capa básica de .NET	`w3wp`	`procstat cpu_usage` , `procstat memory_rss` , `procstat memory_vms` , `procstat read_bytes` , `procstat write_bytes`

Métricas de contador de rendimiento

Las métricas de contador de rendimiento solo se recomiendan para las instancias cuando los conjuntos de contadores de rendimiento correspondientes están instalados en las instancias de Windows.

Nombre de métrica del contador de rendimiento	Nombre del conjunto de contadores de rendimiento
N.º de excepciones de CLR de.NET lanzadas	Excepciones de CLR de.NET
.NET CLR Exceptions: n.º de excepciones generadas/segundo	Excepciones de CLR de.NET
.NET CLR Exceptions: n.º de filtros/segundo	Excepciones de CLR de.NET
.NET CLR Exceptions: n.º de instrucciones finally/segundo	Excepciones de CLR de.NET
.NET CLR Exceptions: generación para capturar profundidad/segundo	Excepciones de CLR de.NET
.NET CLR Interop: n.º de CCWs	.NET CLR Interop
.NET CLR Interop: n.º de stubs	.NET CLR Interop
.NET CLR Interop: n.º de exportaciones TLB/segundo	.NET CLR Interop
.NET CLR Interop: n.º de importaciones TLB/segundo	.NET CLR Interop
.NET CLR Interop: n.º de serialización	.NET CLR Interop
.NET CLR Jit: % de tiempo en Jit	.NET CLR Jit
.NET CLR Jit: errores de Jit estándar	.NET CLR Jit
.NET CLR Loading: % de tiempo de carga	.NET CLR Loading
.NET CLR Loading: errores de velocidad de carga	.NET CLR Loading
.NET CLR LocksAndThreads: tasa de contención/segundo	.NET CLR LocksAndThreads

Nombre de métrica del contador de rendimiento	Nombre del conjunto de contadores de rendimiento
.NET CLR LocksAndThreads: longitud de cola/segundo	.NET CLR LocksAndThreads
.NET CLR Memory: n.º de bytes confirmados totales	Memoria CLR de.NET
Tiempo de % de memoria de CLR de .NET en GC	Memoria CLR de.NET
.NET CLR Networking 4.0.0.0: tiempo medio de cola de HttpWebRequest	.NET CLR Networking 4.0.0.0
.NET CLR Networking 4.0.0.0: solicitudes HttpWebRequest anuladas/segundo	.NET CLR Networking 4.0.0.0
.NET CLR Networking 4.0.0.0: solicitudes HttpWebRequest con errores/segundo	.NET CLR Networking 4.0.0.0
.NET CLR Networking 4.0.0.0: solicitudes HttpWebRequest en cola/segundo	.NET CLR Networking 4.0.0.0
APP_POOL_WAS Errores totales de ping en el proceso de trabajo	APP_POOL_WAS
Reinicios de aplicaciones ASP.NET	ASP.NET
Solicitudes ASP.NET rechazadas	ASP.NET
Reinicios del proceso de trabajo ASP.NET	ASP.NET
Trims de API de la caché de aplicaciones ASP.NET	Aplicaciones ASP.NET
Porcentaje de tiempo del procesador administrado (estimado) de las aplicaciones ASP.NET	Aplicaciones ASP.NET
Errores de aplicaciones ASP.NET totales/sec	Aplicaciones ASP.NET

Nombre de métrica del contador de rendimiento	Nombre del conjunto de contadores de rendimiento
Errores de aplicaciones ASP.NET no controlados durante la ejecución/s	Aplicaciones ASP.NET
Solicitudes de aplicaciones ASP.NET en la cola de aplicaciones	Aplicaciones ASP.NET
Solicitudes de aplicaciones ASP.NET/s	Aplicaciones ASP.NET
Tiempo de espera de solicitudes ASP.NET	ASP.NET
Solicitudes ASP.NET en cola	ASP.NET
Base de datos ==> Porcentaje de aciertos de la caché de la base de datos de instancias	Base de datos ==> Instancias
Base de datos ==> Latencia media de lecturas de la base de datos de E/S de instancias	Base de datos ==> Instancias
Base de datos ==> Lecturas de la base de datos de E/S de instancias por segundo	Base de datos ==> Instancias
Base de datos ==> Latencia media de escrituras de registros de E/S de instancias	Base de datos ==> Instancias
Operaciones de replicación pendientes de DRA de DirectoryServices	DirectoryServices
Sincronizaciones de replicación pendientes de DRA de DirectoryServices	DirectoryServices
Tiempo de enlace LDAP de DirectoryServices	DirectoryServices
Consultas recursivas de DNS por segundo	DNS
Error de consulta recursiva de DNS por segundo	DNS

Nombre de métrica del contador de rendimiento	Nombre del conjunto de contadores de rendimiento
Consultas de TCP de DNS recibidas por segundo	DNS
Consultas totales de DNS recibidas por segundo	DNS
Respuestas totales de DNS enviadas por segundo	DNS
Consultas de UDP de DNS recibidas por segundo	DNS
Colas de solicitudes de servicio HTTP CurrentQueueSize	Colas de solicitudes de servicio HTTP
Porcentaje de espacio libre en disco lógico	LogicalDisk
Promedio LogicalDisk Escritura en disco/segundo	LogicalDisk
Promedio LogicalDisk Lectura de disco/segundo	LogicalDisk
Promedio LogicalDisk Longitud de la cola de disco	LogicalDisk
Porcentaje de bytes confirmados en uso en memoria	Memoria
Mbytes disponibles de memoria	Memoria
Páginas de memoria/segundo	Memoria
Duración media de la memoria caché en espera a largo plazo	Memoria
Total de bytes de interfaz de red/segundo	Interfaz de red

Nombre de métrica del contador de rendimiento	Nombre del conjunto de contadores de rendimiento
Bytes de interfaz de red recibidos por segundo	Interfaz de red
Bytes de interfaz de red enviados por segundo	Interfaz de red
Ancho de banda de interfaz de red actual	Interfaz de red
Porcentaje de uso de archivo de paginación	Archivo de paginación
Porcentaje de tiempo de disco en disco físico	PhysicalDisk
Promedio de disco físico Longitud de la cola de disco	PhysicalDisk
Promedio de disco físico Lectura de disco/segundo	PhysicalDisk
Promedio de disco físico Escritura en disco/segundo	PhysicalDisk
Bytes leídos/segundo en disco físico	PhysicalDisk
Lecturas/segundo en disco físico	PhysicalDisk
Bytes de escritura/segundo en disco físico	PhysicalDisk
Escrituras/segundo en disco físico	PhysicalDisk
Porcentaje de tiempo de inactividad del procesador	Procesador
Porcentaje de horas de interrupción del procesador	Procesador
Porcentaje de tiempo de procesador del procesador	Procesador
Porcentaje de tiempo de usuario del procesador	Procesador

Nombre de métrica del contador de rendimiento	Nombre del conjunto de contadores de rendimiento
Relación de llenado de caché de blob de la caché basada en disco de SharePoint	Caché basada en disco de SharePoint
Vaciados de caché de blob de la caché basada en disco de SharePoint por segundo	Caché basada en disco de SharePoint
Relación de aciertos de caché de blob de la caché basada en disco de SharePoint	Caché basada en disco de SharePoint
Número total de compactaciones de la caché basada en disco de SharePoint	Caché basada en disco de SharePoint
Solicitud de página/tiempo de ejecución de SharePoint Foundation	SharePoint Foundation
Vaciados de caché de publicación de la caché de publicación de SharePoint por segundo	Caché de publicación de SharePoint
Autenticaciones Kerberos de estadísticas de seguridad en todo el sistema	Estadísticas de seguridad para todo el sistema
Autenticaciones NTLM de estadísticas de seguridad en todo el sistema	Estadísticas de seguridad para todo el sistema
SQLServer: métodos de acceso: registros reenviados/segundo	SQLServer: métodos de acceso
SQLServer: métodos de acceso: exploraciones completas/segundo	SQLServer: métodos de acceso
SQLServer: métodos de acceso: divisiones de página/segundo	SQLServer: métodos de acceso
SQLServer: administrador del búfer: porcentaje de aciertos de caché del búfer	SQLServer: Buffer Manager

Nombre de métrica del contador de rendimiento	Nombre del conjunto de contadores de rendimiento
SQLServer: administrador del búfer: duración prevista de página	SQLServer: Buffer Manager
SQLServer: bytes recibidos por segundo del archivo de réplica de base de datos	SQLServer: réplica de base de datos
SQLServer: bytes recibidos por segundo del registro de réplica de base de datos	SQLServer: réplica de base de datos
SQLServer: restante para deshacer del registro de réplica de base de datos	SQLServer: réplica de base de datos
SQLServer: cola de envío de registros de réplica de base de datos	SQLServer: réplica de base de datos
SQLServer: transacción por segundo de escritura reflejada de réplica de base de datos	SQLServer: réplica de base de datos
SQLServer: cola de recuperación de réplica de base de datos	SQLServer: réplica de base de datos
SQLServer: resto de bytes de rehacer de réplica de base de datos	SQLServer: réplica de base de datos
SQLServer: bytes por segundo rehechos de réplica de base de datos	SQLServer: réplica de base de datos
SQLServer: registro total de réplica de base de datos que requiere deshacer	SQLServer: réplica de base de datos
SQLServer: retraso de transacción de réplica de base de datos	SQLServer: réplica de base de datos
SQLServer: estadísticas generales: procesos bloqueados	SQLServer: estadísticas generales

Nombre de métrica del contador de rendimiento	Nombre del conjunto de contadores de rendimiento
SQLServer: estadísticas generales: conexiones de usuario	SQLServer: estadísticas generales
SQLServer: bloqueos temporales: tiempo medio de espera de bloqueo temporal (ms)	SQLServer: bloqueos temporales
SQLServer: bloqueos: tiempo medio de espera de bloqueo (ms)	SQLServer: bloqueos
SQLServer: bloqueos: tiempos de espera de bloqueo agotados/segundo	SQLServer: bloqueos
SQLServer: bloqueos: esperas de bloqueo/s egundo	SQLServer: bloqueos
SQLServer: bloqueos: número de interbloq ueos/segundo	SQLServer: bloqueos
SQLServer: administrador de memoria: concesiones de memoria pendientes	SQLServer: administrador de memoria
SQLServer: estadísticas de SQL: solicitudes por lotes/segundo	SQLServer: SQL Statistics
SQLServer: estadísticas de SQL: compilaci ones de SQL/segundo	SQLServer: SQL Statistics
SQLServer: estadísticas de SQL: recopilac iones de SQL/segundo	SQLServer: SQL Statistics
Longitud de cola del procesador del sistema	System (Sistema)
Conexiones TCPv4 establecidas	TCPv4
Conexiones TCPv6 establecidas	TCPv6

Nombre de métrica del contador de rendimiento	Nombre del conjunto de contadores de rendimiento
W3SVC_W3WP: descargas de caché de archivos	W3SVC_W3WP
W3SVC_W3WP: errores de caché de archivos	W3SVC_W3WP
W3SVC_W3WP: solicitudes/segundo	W3SVC_W3WP
W3SVC_W3WP: descargas de caché de URI	W3SVC_W3WP
W3SVC_W3WP: errores de caché de URI	W3SVC_W3WP
Bytes de servicios web recibidos/s	Servicios Web
Bytes de servicios web enviados/s	Servicios Web
Intentos de conexión de servicio web/segundo	Servicios Web
Conexiones actuales de servicios web	Servicios Web
Solicitudes de recepción de servicio web/segundo	Servicios Web
Solicitudes de publicación de servicio web/segundo	Servicios Web

Uso de la vista del estado de recursos en la consola de CloudWatch

Puede utilizar la vista de estado de recursos para descubrir, administrar y visualizar automáticamente el estado y el rendimiento de los anfitriones en las aplicaciones en una sola vista. Puede visualizar el estado de sus anfitriones por una dimensión de rendimiento, como CPU o memoria, y cortar cientos de anfitriones en una sola vista mediante el uso de filtros. Puede filtrar por etiquetas o por casos de uso, como anfitriones del mismo grupo de Auto Scaling o anfitriones que utilizan el mismo balanceador de carga,

Requisitos previos

Para asegurarse de obtener el máximo beneficio de la vista de estado de recursos, verifique que cumple los siguientes prerequisitos.

- Para ver la utilización de la memoria de sus anfitriones y utilizarla como filtro, debe instalar el agente de CloudWatch en sus anfitriones y configurarlo para enviar una métrica de memoria a CloudWatch en el espacio de nombres del `CWAgent` predeterminado. En las instancias de Linux y macOS, el agente de CloudWatch debe enviar la métrica de `mem_used_percent`. En las instancias de Windows, el agente debe enviar la métrica de `Memory % Committed Bytes In Use`. Estas métricas se incluyen si utiliza el asistente para crear el archivo de configuración del agente de CloudWatch y selecciona cualquiera de los conjuntos de métricas predefinidos. Las métricas recopiladas por el agente de CloudWatch se facturan como métricas personalizadas. Para obtener más información, consulte Instalación del agente de CloudWatch.

  Cuando utilice el agente de CloudWatch para recopilar estas métricas de memoria para utilizarlas con la vista de estado de recursos, debe incluir la siguiente sección en el archivo de configuración del agente de CloudWatch. Esta sección contiene la configuración de dimensiones predeterminada y se crea de forma predeterminada, así que no cambie ninguna parte de esta sección por algo diferente de lo que se muestra en el siguiente ejemplo.

```
"append_dimensions": {
  "ImageId": "${aws:ImageId}",
  "InstanceId": "${aws:InstanceId}",
  "InstanceType": "${aws:InstanceType}",
  "AutoScalingGroupName": "${aws:AutoScalingGroupName}"
},
```

- Para ver toda la información disponible en la vista de estado de recursos, debe haber iniciado sesión en una cuenta que tenga los siguientes permisos. Si ha iniciado sesión con menos permisos, puede seguir utilizando la vista de estado de recursos, pero algunos datos de rendimiento no estarán visibles.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "autoscaling:Describe*",
                "cloudwatch:Describe*",
```

```
                "cloudwatch:Describe*",
                "cloudwatch:Get*",
                "cloudwatch:List*",
                "logs:Get*",
                "logs:Describe*",
                "sns:Get*",
                "sns:List*",
                "ec2:DescribeInstances",
                "ec2:DescribeInstanceStatus",
                "ec2:DescribeRegions"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

Para ver el estado de recursos en su cuenta

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Monitoreo de infraestructuras, Estado del recurso.

   Aparecerá la página de estado de recursos que muestra un cuadrado para cada host de su cuenta. Cada cuadrado se colorea en función del estado actual de ese host, según la configuración de Color by (Colorear por). Los cuadrados del host con un símbolo de alarma tienen una o más alarmas actualmente en estado de ALARM (ALARMA).

   Puede ver hasta 500 anfitriones en una sola vista. Si tiene más anfitriones en la cuenta, utilice la configuración de filtro en el paso 6 de este procedimiento.
3. Para cambiar los criterios que se utilizan para mostrar el estado de cada anfitrión, elija una configuración para Color by (Colorear por). Puede elegir CPU Utilization (Utilización de la CPU),Memory Utilization (Utilización de la memoria) o Status check (Verificación de estado). Las métricas de utilización de la memoria sólo están disponibles para los anfitriones que ejecutan el agente de CloudWatch y lo tienen configurado para recopilar métricas de memoria y enviarlas al espacio de nombres del `CWAgent` determinado . Para obtener más información, consulte Recopile las métricas, registros y seguimientos con el agente de CloudWatch.
4. Para cambiar los umbrales y los colores que se utilizan para los indicadores de estado en la cuadrícula, elija el icono con el engranaje situado por encima de la cuadrícula.

5. Para alternar si desea mostrar alarmas en la cuadrícula del host, seleccione o desactive Show alarms across all metrics (Mostrar alarmas en todas las métricas).
6. Para dividir en grupos los anfitriones del mapa, elija un criterio de agrupación para Group by (Agrupar por).
7. Para reducir la vista a menos anfitriones, elija un criterio de filtro para Filter by (Filtrar por). Puede filtrar por etiquetas y por agrupaciones de recursos, como grupo de Auto Scaling, tipo de instancias, grupo de seguridad, etc.
8. Para ordenar los anfitriones, elija un criterio de clasificación para Sort by (Ordenar por). Puede ordenar por resultados de verificación de estado, por estado de instancia, utilización de la CPU o la memoria y por el número de alarmas que están en estado de ALARM (ALARMA).
9. Para ver más información sobre un host, elija el cuadrado que representa a ese anfitrión. Aparecerá un panel emergente. Para ver más detalles acerca de ese host, elija View dashboard (Ver panel) oView on list (Ver en lista).

Observabilidad entre cuentas de CloudWatch

Gracias a la observabilidad entre cuentas de Amazon CloudWatch, puede supervisar y solucionar problemas en las aplicaciones que abarcan varias cuentas de una región. Puede buscar, visualizar y analizar sin problemas sus métricas, registros, seguimientos, aplicaciones de Información de aplicaciones y monitores de Internet Monitor en cualquiera de las cuentas vinculadas, sin límites de cuenta.

Configure una o más cuentas de AWS como cuentas de monitoreo y vincúlelas con varias cuentas de origen. Una cuenta de monitoreo es una cuenta central de AWS que puede ver los datos de observabilidad generados a partir de las cuentas de origen e interactuar con ellos. Una cuenta de origen es una cuenta individual de AWS que genera datos de observabilidad para los recursos que residen en ella. Las cuentas de origen comparten sus datos de observabilidad con la cuenta de monitoreo. Los datos de observabilidad compartidos pueden incluir los siguientes tipos de telemetría:

- Métricas en Amazon CloudWatch. Puede elegir compartir las métricas de todos los espacios de nombres con la cuenta de monitoreo o filtrarlas a un subconjunto de espacios de nombres.
- Grupos de registro de Registros de Amazon CloudWatch. Puede elegir compartir todos los grupos de registros con la cuenta de supervisión o filtrarlos a un subconjunto de grupos de registros.
- Trazas en AWS X-Ray
- Aplicaciones en Información de aplicaciones de Amazon CloudWatch
- Monitores en CloudWatch Internet Monitor

Para crear enlaces entre las cuentas de monitoreo y las de origen, puede usar la consola de CloudWatch. Como alternativa, utilice los comandos del administrador de acceso de observabilidad en AWS CLI y la API. Para más información, consulte Observability Access Manager API Reference (Referencia de la API del administrador de acceso de observabilidad).

Un receptor es un recurso que representa un punto de enlace en una cuenta de monitoreo. Las cuentas de origen se pueden vincular al receptor para compartir datos de observabilidad. Cada cuenta puede tener un receptor por región. La cuenta de monitoreo administra cada receptor en el lugar donde se encuentre. Un enlace de observabilidad es un recurso que representa el enlace establecido entre una cuenta de origen y una cuenta de monitoreo. Los enlaces los administra la cuenta de origen.

Para ver una demostración en vídeo sobre cómo configurar la observabilidad entre cuentas de CloudWatch, consulte el siguiente vídeo.

En el siguiente tema se explica cómo configurar la observabilidad entre cuentas de CloudWatch tanto en las cuentas de monitoreo como en las de origen. Para obtener más información sobre el panel de CloudWatch entre cuentas y regiones, consulte Consola de CloudWatch para cuentas y Regiones cruzadas.

Use Organizations para las cuentas de origen

Hay dos opciones para vincular las cuentas de origen a la cuenta de monitoreo. Puede usar una opción o ambas.

- Se usa AWS Organizations para vincular las cuentas de una organización o unidad organizativa a la cuenta de monitoreo.
- Conecte cuentas de AWS individuales a la cuenta de monitoreo.

Le recomendamos que use Organizations para que las cuentas nuevas de AWS que cree más adelante en la organización se incorporen automáticamente como cuentas de origen en la opción de observabilidad entre cuentas.

Detalles sobre la vinculación de cuentas de monitoreo y cuentas de origen

- Cada cuenta de monitoreo se puede vincular a un máximo de 100 000 cuentas de origen.
- Cada cuenta de origen puede compartir datos con hasta cinco cuentas de monitoreo.
- Puede configurar una sola cuenta como cuenta de monitoreo y como cuenta de origen. Si lo hace, recuerde que esta cuenta únicamente envía sus propios datos de observabilidad a la cuenta de monitoreo vinculada. No transmite los datos de sus cuentas de origen.
- Una cuenta de monitoreo especifica qué tipos de telemetría se pueden compartir con ella. En cambio, una cuenta de origen especifica los tipos de telemetría que quiera compartir.
  - Si hay más tipos de telemetría seleccionados en la cuenta de monitoreo que en la cuenta de origen, quiere decir que las cuentas están vinculadas. Solo se comparten los tipos de datos seleccionados en ambas cuentas.
  - Si hay más tipos de telemetría seleccionados en la cuenta de origen que en la cuenta de monitoreo, se produce un error en la creación del enlace y no se comparte nada.

  - El nombre de una métrica no aparece en la consola de la cuenta de supervisión hasta que esa métrica emita nuevos puntos de datos una vez creado el enlace.
- Si quiere eliminar un enlace entre cuentas, hágalo desde la cuenta de origen.
- Para eliminar un receptor en una cuenta de monitoreo, primero debe eliminar todos los enlaces a ese receptor de la cuenta de monitoreo.

Precios

La observabilidad entre cuentas en CloudWatch no implica ningún costo adicional para los registros y las métricas, y la primera copia del registro de seguimiento es gratuita. Para obtener más información sobre precios, consulte Precios de Amazon CloudWatch.

Contenido

Vinculación de cuentas de monitoreo con cuentas de origen

En los temas de esta sección se explica cómo configurar enlaces entre cuentas de monitoreo y cuentas de origen.

Le recomendamos que cree una cuenta nueva de AWS que sirva como cuenta de monitoreo para su organización.

Contenido

Permisos necesarios

Para crear enlaces entre una cuenta de monitoreo y una cuenta de origen, debe iniciar sesión con ciertos permisos.

- Para configurar una cuenta de monitoreo, debe tener acceso completo de administrador a la cuenta de monitoreo o debe iniciar sesión en esa cuenta con los siguientes permisos:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowSinkModification",
            "Effect": "Allow",
            "Action": [
                "oam:CreateSink",
                "oam:DeleteSink",
                "oam:PutSinkPolicy",
                "oam:TagResource"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AllowReadOnly",
```

```
            "Effect": "Allow",
            "Action": ["oam:Get*", "oam:List*"],
            "Resource": "*"
        }
    ]
}
```

- Cuenta de origen en el ámbito de una cuenta de monitoreo específica: para crear, actualizar y administrar enlaces para una sola cuenta de monitoreo específica, debe iniciar sesión en la cuenta con, al menos, los siguientes permisos. En este ejemplo, la cuenta de monitoreo es 999999999999.

  Si el enlace no va a compartir los cinco tipos de recursos (métricas, registros, seguimientos, aplicaciones de Información de aplicaciones y monitores de Internet Monitor), puede omitir `cloudwatch:Link`, `logs:Link`, `xray:Link`, `applicationinsights:Link` o `internetmonitor:Link` según sea necesario.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "oam:CreateLink",
                "oam:UpdateLink",
                "oam:DeleteLink",
                "oam:GetLink",
                "oam:TagResource"
            ],
            "Effect": "Allow",
            "Resource": "arn:*:oam:*:*:link/*"
        },
        {
            "Action": [
                "oam:CreateLink",
                "oam:UpdateLink"
            ],
            "Effect": "Allow",
            "Resource": "arn:*:oam:*:*:sink/*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceAccount": [
                        "999999999999"
```

```
                    ]
                }
            }
        },
        {
            "Action": "oam:ListLinks",
            "Effect": "Allow",
            "Resource": "*"
        },
        {
            "Action": "cloudwatch:Link",
            "Effect": "Allow",
            "Resource": "*"
        },
        {
            "Action": "logs:Link",
            "Effect": "Allow",
            "Resource": "*"
        },
        {
            "Action": "xray:Link",
            "Effect": "Allow",
            "Resource": "*"
        },
        {
             "Action": "applicationinsights:Link",
             "Effect": "Allow",
             "Resource": "*"
         },
        {
             "Action": "internetmonitor:Link",
             "Effect": "Allow",
             "Resource": "*"
        }
    ]
}
```

- Cuenta de origen, con permisos para vincularse a cualquier cuenta de monitoreo: para crear un enlace a cualquier receptor de cuenta de monitoreo existente y compartir métricas, grupos de registro, seguimientos, aplicaciones de Información de aplicaciones y monitores de Internet Monitor, debe iniciar sesión en la cuenta de origen con permisos de administrador completos o iniciar sesión en esta con los siguientes permisos

Si el enlace no va a compartir los cinco tipos de recursos (métricas, registros, seguimientos, aplicaciones de Información de aplicaciones y monitores de Internet Monitor), puede omitir `cloudwatch:Link`, `logs:Link`, `xray:Link`, `applicationinsights:Link` o `internetmonitor:Link` según sea necesario.

```
{
    "Version": "2012-10-17",
    "Statement": [{
            "Effect": "Allow",
            "Action": [
                "oam:CreateLink",
                "oam:UpdateLink"
            ],
            "Resource": [
                "arn:aws:oam:*:*:link/*",
                "arn:aws:oam:*:*:sink/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "oam:List*",
                "oam:Get*"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "oam:DeleteLink",
                "oam:GetLink",
                "oam:TagResource"
            ],
            "Resource": "arn:aws:oam:*:*:link/*"
        },
        {
            "Action": "cloudwatch:Link",
            "Effect": "Allow",
            "Resource": "*"
        },
        {
            "Action": "xray:Link",
```

```
            "Effect": "Allow",
            "Resource": "*"
        },
        {
            "Action": "logs:Link",
            "Effect": "Allow",
            "Resource": "*"
        },
        {
             "Action": "applicationinsights:Link",
             "Effect": "Allow",
             "Resource": "*"
        },
        {
             "Action": "internetmonitor:Link",
             "Effect": "Allow",
             "Resource": "*"
        }
    ]
}
```

Descripción general de la configuración

Los siguientes pasos de alto nivel le muestran cómo configurar la observabilidad entre cuentas de CloudWatch.

> **Note**
>
> Le recomendamos crear una nueva cuenta de AWS para usarla como cuenta de monitoreo de la organización.

1. Configure una cuenta de monitoreo dedicada.
2. (Opcional) Descargue una plantilla de AWS CloudFormation o copie una URL para vincular las cuentas de origen.
3. Vincule las cuentas de origen a la cuenta de monitoreo.

Después de completar estos pasos, puede usar la cuenta de monitoreo para ver los datos de observabilidad de las cuentas de origen.

Paso 1: configuración de una cuenta de monitoreo

Siga los pasos de esta sección para configurar una cuenta de AWS como cuenta de monitoreo para la observabilidad entre cuentas de CloudWatch.

Requisitos previos

- Si quiere configurar las cuentas de una organización AWS Organizations como cuentas de origen: obtenga la ruta de acceso de la organización o el ID de esta.
- Si no usa Organizations en las cuentas de origen, puede obtener los ID de las cuentas de origen.

Para configurar una cuenta como cuenta de monitoreo, debe contar con determinados permisos. Para obtener más información, consulte Permisos necesarios.

Configuración de una cuenta de monitoreo

1. Inicie sesión en la cuenta de que quiera usar como cuenta de monitoreo.
2. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
3. En el panel de navegación izquierdo, elija Configuración.
4. Al supervisar la configuración de la cuenta, seleccione Configure (Configurar).
5. En Seleccionar datos, decida si la cuenta de monitoreo podrá ver los datos de Registros, Métricas, Seguimientos, Información de aplicaciones: aplicaciones e Internet Monitor: monitores de las cuentas de origen a las que esté vinculada.
6. Para enumerar las cuentas de origen, ingrese las cuentas de origen que verá la cuenta de monitoreo. Para identificar las cuentas de origen, introduzca los ID de las cuentas individuales, las rutas de acceso de la organización o los ID de la organización. Si ingresa una ruta de acceso de la organización o un ID de organización, tenga presente que la cuenta de monitoreo podrá ver los datos de observabilidad de todas las cuentas vinculadas a la organización.

   Separe las entradas de la lista con comas.

   > **Important**
   >
   > Cuando introduzca la ruta de una organización, siga el formato exacto. El ou-id debe terminar con una / (barra oblicua). Por ejemplo: `o-a1b2c3d4e5/r-f6g7h8i9j0example/ou-def0-awsbbbb/`.

7. En Define a label to identify your source account (Definir una etiqueta para identificar la cuenta de origen), especifique si quiere usar nombres de cuenta o direcciones de correo electrónico para identificar las cuentas de origen cuando use la cuenta de monitoreo para verlas.
8. Elija Configurar.

> **Important**
>
> El enlace entre las cuentas de monitoreo y de origen no estará completo hasta que configure las cuentas de origen. Para obtener más información, consulte las siguientes secciones.

Paso 2: (opcional) descarga de una plantilla o URL de AWS CloudFormation

Para vincular las cuentas de origen a una cuenta de monitoreo, es recomendable usar una plantilla o una URL de AWS CloudFormation.

- Si va a vincular una organización completa, CloudWatch le proporciona una plantilla AWS CloudFormation.
- Si va a vincular cuentas individuales, use la plantilla o URL de AWS CloudFormation que le proporcione CloudWatch.

Para usar una plantilla AWS CloudFormation, debe descargarla durante estos pasos. Después de vincular la cuenta de monitoreo con al menos una cuenta de origen, la plantilla AWS CloudFormation ya no estará disponible para descargar.

Descarga de una plantilla AWS CloudFormation o copiar una URL para vincular las cuentas de origen a la cuenta de monitoreo

1. Inicie sesión en la cuenta de que quiera usar como cuenta de monitoreo.
2. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
3. En el panel de navegación izquierdo, elija Configuración.
4. En la opción Monitoring account configuration (Configuración de cuentas de supervisión), elija Resources to link accounts (Recursos para vincular cuentas).
5. Realice una de las siguientes acciones siguientes:

- Elija AWS organización para obtener la plantilla que quiera usar para vincular las cuentas de una organización a esta cuenta de monitoreo.
- Elija Any account (Cualquier cuenta) para obtener una plantilla o URL para configurar cuentas individuales como cuentas de origen.

6. Realice una de las siguientes acciones siguientes:

   - Si elige AWS organization, seleccione Download CloudFormation template (Descargar plantilla de CloudFormation).
   - Si elige Any account, seleccione Download CloudFormation template (Descargar la plantilla de CloudFormation) o Copy URL (Copiar URL).

7. (Opcional) Repita los pasos 5 y 6 para descargar tanto la plantilla AWS CloudFormation como la URL.

Paso 3: vinculación de las cuentas de origen

Siga los pasos de estas secciones para vincular cuentas de origen a una cuenta de monitoreo.

Para vincular cuentas de monitoreo con cuentas de origen, debe contar con determinados permisos. Para obtener más información, consulte Permisos necesarios.

Use una plantilla de AWS CloudFormation para configurar todas las cuentas de una organización o unidad organizativa como cuentas de origen

En estos pasos se supone que ya ha descargado la plantilla de AWS CloudFormation necesaria al realizar los pasos de la sección Paso 2: (opcional) descarga de una plantilla o URL de AWS CloudFormation.

Uso de una plantilla de AWS CloudFormation para vincular las cuentas de una organización o unidad organizativa a la cuenta de monitoreo

1. Inicie sesión en la cuenta administrativa de la organización.
2. Abra la consola de AWS CloudFormation en https://console.aws.amazon.com/cloudformation.
3. En el panel de navegación de la izquierda, seleccione StackSets.
4. Compruebe que ha iniciado sesión en la región que quiera usar y, a continuación, seleccione Create StackSet (Crear StackSet).
5. Elija Siguiente.

6. Seleccione Template is ready (La plantilla está lista) y Upload a template file (Cargar un archivo de plantilla).
7. Seleccione Choose file (Elegir archivo), elija la plantilla que descargó de la cuenta de monitoreo y haga clic en Open (Abrir).
8. Elija Siguiente.
9. En la página Specify StackSet details (Especificar detalles de StackSet), escriba el nombre de la instancia de StackSet, y haga clic en Next (Siguiente).
10. En Add stacks to stack set (Agregar pilas al conjunto de pilas), seleccione Deploy new stacks (Implementar pilas nuevas).
11. En Deployment targets (Objetivos de implementación), decida si quiere realizar la implementación en toda la organización o en unidades organizativas especificadas.
12. En Specify regions (Especificar regiones), seleccione las regiones en las que quiera implementar la observabilidad entre cuentas de CloudWatch.
13. Elija Siguiente.
14. En la página Review (Revisar), confirme las opciones seleccionadas y haga clic en Submit (Enviar).
15. En la pestaña Stack instances (Instancias de pila), actualice la pantalla hasta que vea que las instancias de pila tienen el estado CREATE_COMPLETE (CREACIÓN COMPLETA).

Uso de una plantilla AWS CloudFormation para configurar cuentas de origen individuales

En estos pasos se supone que ya ha descargado la plantilla de AWS CloudFormation necesaria al realizar los pasos de la sección Paso 2: (opcional) descarga de una plantilla o URL de AWS CloudFormation.

Uso de una plantilla de AWS CloudFormation para configurar cuentas de origen individuales para la observabilidad entre cuentas de CloudWatch

1. Inicie sesión en la cuenta de origen.
2. Abra la consola de AWS CloudFormation en https://console.aws.amazon.com/cloudformation.
3. En el panel de navegación de la izquierda, seleccione Stacks (Pilas).
4. Compruebe que ha iniciado sesión en la región que quiera usar y, a continuación, seleccione Create stack (Crear pila) y la opción With new resources (standard) (Con recursos nuevos [estándar]).

5. Elija Siguiente.
6. Elija Upload a template file (Cargar un archivo de plantilla).
7. Seleccione Choose file (Elegir archivo), elija la plantilla que descargó de la cuenta de monitoreo y haga clic en Open (Abrir).
8. Elija Siguiente.
9. En la página Specify stack details (Especificar los detalles de la pila), ingrese un nombre para la pila, y haga clic en Next (Siguiente).
10. En la página Configurar opciones de pila, elija Siguiente.
11. En la página Review (Revisar), elija Submit (Enviar).
12. En la página de estado de la pila, actualice la pantalla hasta que vea que la pila tiene el estado CREATE_COMPLETE.
13. Para usar esta misma plantilla para vincular más cuentas de origen a la cuenta de monitoreo, cierre la sesión de esta cuenta e iníciela en la siguiente cuenta de origen. A continuación, repita los pasos 2 a 12.

Uso de una URL para configurar cuentas de origen individuales

En estos pasos se supone que ya copió la URL necesaria al realizar los pasos de la sección Paso 2: (opcional) descarga de una plantilla o URL de AWS CloudFormation.

Uso de una URL para vincular cuentas de origen individuales a la cuenta de monitoreo

1. Inicie sesión en la cuenta que quiera usar como cuenta de monitoreo.
2. Ingrese la URL que copió de la cuenta de monitoreo.

   Verá la página de configuración de CloudWatch, con algunos datos rellenados.
3. En Seleccionar datos, decida si la cuenta de monitoreo podrá compartir los datos de Registros, Métricas, Seguimientos, Información de aplicaciones: aplicaciones e Internet Monitor: monitores de las cuentas de origen a las que esté vinculada.

   Tanto para los Registros como para las Métricas, puede elegir si desea compartir todos los recursos o un subconjunto con la cuenta de monitoreo.

   a. (Opcional) Para compartir un subconjunto de los grupos de registro de esta cuenta con la cuenta de supervisión, seleccione Registros y elija Filtrar registros. A continuación, utilice el cuadro Filtrar registros para crear una consulta que busque los grupos de registro

que desee compartir. La consulta utilizará el término `LogGroupName` y uno o más de los siguientes operandos.

- `=` y `!=`
- `AND`
- `OR`
- `^` indica IGUAL A y `!^` indica NO ES IGUAL A. Solo se pueden usar como búsquedas de prefijos. Incluya un `%` al final de la cadena que desee buscar e incluir.
- `IN` y `NOT IN`, usando paréntesis `(  )`

La consulta completa no debe tener más de 2000 caracteres y está limitada a cinco operandos condicionales. Los operandos condicionales son `AND` y `OR`. No existe límite en el número de otros operandos.

> **Tip**
>
> Seleccione Ver las consultas de muestra para ver la sintaxis correcta de los formatos de consulta más comunes.

b. (Opcional) Para compartir un subconjunto de los espacios de nombres de métricas de esta cuenta con la cuenta de monitoreo, seleccione Métricas y elija Filtrar métricas. A continuación, utilice el recuadro Filtrar registros para construir una consulta para encontrar los espacios de nombres de métricas que desea compartir. Utilice el término `Namespace` y uno o más de los siguientes operandos.

- `=` y `!=`
- `AND`
- `OR`
- `LIKE` y `NOT LIKE`. Solo se pueden usar como búsquedas de prefijos. Incluya un `%` al final de la cadena que desee buscar e incluir.
- `IN` y `NOT IN`, usando paréntesis `(  )`

La consulta completa no debe tener más de 2000 caracteres y está limitada a cinco operandos condicionales. Los operandos condicionales son `AND` y `OR`. No existe límite en el número de otros operandos.

> **Tip**
>
> Seleccione Ver las consultas de muestra para ver la sintaxis correcta de los formatos de consulta más comunes.

4. No cambie el ARN de la opción Enter monitoring account configuration ARN (Introduzca el ARN de configuración de cuentas de supervisión).
5. La sección Define a label to identify your source account (Definir una etiqueta para identificar la cuenta de origen) ya está rellenada con la opción de etiqueta de la cuenta de monitoreo. Para cambiar la opción, puede seleccionar Edit (Editar).
6. Elija Vincular.
7. Ingrese **`Confirm`** en el recuadro y elija Confirm (Confirmar).
8. Si quiere usar esta misma URL para vincular más cuentas de origen a la cuenta de monitoreo, cierre la sesión de esta cuenta e inicie sesión en la siguiente cuenta de origen. A continuación, repita los pasos 2 a 7.

Administración de las cuentas de monitoreo y las cuentas de origen

Después de configurar las cuentas de monitoreo y las de origen, puede seguir los pasos de estas secciones para administrarlas.

Contenido

Vinculación de más cuentas de origen a una cuenta de monitoreo existente

Siga los pasos de esta sección para agregar enlaces desde cuentas de origen adicionales a una cuenta de monitoreo existente.

Cada cuenta de origen se puede vincular a un máximo de cinco cuentas de monitoreo. Cada cuenta de monitoreo se puede vincular a un máximo de 100 000 cuentas de origen.

Para administrar una cuenta de origen, debe contar con determinados permisos. Para obtener más información, consulte Permisos necesarios.

Incorporación de más cuentas de origen a una cuenta de monitoreo

1. Inicie sesión en la cuenta de monitoreo.
2. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
3. En el panel de navegación izquierdo, elija Configuración.
4. En la opción Monitoring account configuration (Configuración de cuentas de supervisión), elija Manage source accounts (Administrar cuentas de origen).
5. Elija la pestaña Configuration policy (Política de configuración).
6. En el cuadro Configuration policy (Política de configuración), agregue el nuevo ID de la cuenta de origen en la línea Principal (Entidad principal).

   Por ejemplo, suponga que la línea Principal (Principal) es actualmente la siguiente:

   ```
   "Principal": {"AWS": ["111111111111", "222222222222"]}
   ```

   Para agregar 999999999999 como tercera cuenta de origen, edite la línea de la siguiente manera:

   ```
   "Principal": {"AWS": ["111111111111", "222222222222", "999999999999"]}
   ```
7. Elija Actualizar.
8. Elija la pestaña Configuration details (Detalles de configuración).
9. Seleccione el icono de copia que se encuentra junto al ARN del receptor de la cuenta de monitoreo.
10. Inicie sesión en la cuenta que quiera usar como nueva cuenta de monitoreo.
11. Pegue el ARN del receptor de la cuenta de monitoreo que copió en el paso 9.

    Verá la página de configuración de CloudWatch, con algunos datos rellenados.
12. En Seleccionar datos, decida si la cuenta de monitoreo podrá enviar los datos de Registros, Métricas, Seguimientos e Información de aplicaciones: aplicaciones de las cuentas de origen a las que esté vinculada.
13. No cambie el ARN de la opción Enter monitoring account configuration ARN (Introduzca el ARN de configuración de cuentas de supervisión).

14. La sección Define a label to identify your source account (Definir una etiqueta para identificar la cuenta de origen) ya está rellenada con la opción de etiqueta de la cuenta de monitoreo. Para cambiar la opción, puede seleccionar Edit (Editar).
15. Elija Vincular.
16. Ingrese **`Confirm`** en el recuadro y elija Confirm (Confirmar).

Eliminación del enlace entre una cuenta de monitoreo y una cuenta de origen

Siga los pasos de esta sección para detener el envío de datos de una cuenta de origen a una cuenta de monitoreo.

Debe tener los permisos necesarios para administrar una cuenta de origen para completar esta tarea. Para obtener más información, consulte Permisos necesarios.

Eliminación del enlace entre una cuenta de origen y una cuenta de monitoreo

1. Inicie sesión en la cuenta de origen.
2. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
3. En el panel de navegación izquierdo, elija Configuración.
4. En Source account information (Información de la cuenta de origen), seleccione View monitoring accounts (Ver cuentas de monitoreo).
5. Seleccione la casilla de verificación situada junto a la cuenta de monitoreo con la que quiera dejar de compartir datos.
6. Seleccione Stop sharing data (Dejar de compartir datos) y Confirm (Confirmar).
7. Inicie sesión en la cuenta de monitoreo.
8. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
9. Elija Configuración.
10. En Monitoring account information (Información de la cuenta de monitoreo), seleccione View configuration (Ver configuración).
11. En el cuadro Policy (Política), elimine el ID de la cuenta de origen de la línea Principal (Entidad principal) y seleccione Update (Actualizar).

Visualización de la información de una cuenta de monitoreo

Siga los pasos de esta sección para ver la configuración entre cuentas de una cuenta de monitoreo.

Para administrar una cuenta de monitoreo, debe contar con determinados permisos. Para obtener más información, consulte Permisos necesarios.

Administración de una cuenta de monitoreo

1. Inicie sesión en la cuenta de monitoreo.
2. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
3. En el panel de navegación izquierdo, elija Configuración.
4. En la opción Monitoring account configuration (Configuración de cuentas de supervisión), elija Manage source accounts (Administrar cuentas de origen).
5. Para ver la política del administrador de acceso de observabilidad que permite que esta cuenta sea una cuenta de monitoreo, seleccione la pestaña Configuration policy (Política de configuración).
6. Para ver las cuentas de origen que estén vinculadas a esta cuenta de monitoreo, seleccione la pestaña Linked source accounts (Cuentas de origen vinculadas).
7. Para ver el ARN del receptor de la cuenta de monitoreo y los tipos de datos que esta cuenta de monitoreo puede ver en las cuentas de origen vinculadas, seleccione la pestaña Linked source accounts.

Consulta de métricas de otros orígenes de datos

Puede usar CloudWatch para consultar, visualizar y crear alarmas para métricas de otros orígenes de datos. Para ello, debe conectar CloudWatch a los demás orígenes de datos. Esto le proporciona una experiencia de supervisión única y consolidada dentro de la consola de CloudWatch. Puede tener una vista unificada de las métricas de la infraestructura y las aplicaciones, independientemente del lugar donde se almacenen los datos, lo que ayuda a identificar y resolver los problemas con mayor rapidez.

Tras conectarse a un origen de datos mediante un asistente de CloudWatch, CloudWatch crea una pila de AWS CloudFormation que implementa y configura una función AWS Lambda. Esta función de Lambda se ejecuta bajo demanda cada vez que se consulta el origen de datos. El generador de consultas de CloudWatch le muestra en tiempo real una lista de elementos que se pueden consultar, como métricas, tablas, campos o etiquetas. A medida que vaya tomando decisiones, el generador de consultas rellena previamente una consulta en el idioma nativo del origen seleccionado.

CloudWatch proporciona asistentes guiados para que se conecte a los siguientes orígenes de datos. Para estos orígenes de datos, debe proporcionar información básica para identificar el origen de datos y las credenciales. También puede crear conectores a otros orígenes de datos de forma manual mediante la creación de sus propias funciones de Lambda.

- Amazon OpenSearch Service: obtenga métricas de los registros y seguimientos del servicio OpenSearch.
- Amazon Managed Service para Prometheus: consulte estas métricas mediante PromQL.
- Amazon RDS para MySQL: utilice SQL para convertir los datos almacenados en las tablas de Amazon RDS en métricas.
- Amazon RDS para PostgreSQL: utilice SQL para convertir los datos almacenados en las tablas de Amazon RDS en métricas.
- Archivos CSV de Amazon S3: muestran datos de métricas de un archivo CSV almacenado en un bucket de Amazon S3.
- Microsoft Azure Monitor: consulte métricas de la cuenta de Microsoft Azure Monitor.
- Prometheus: consulte estas métricas mediante PromQL.

Después de crear los conectores a los orígenes de datos, consulte Creación de un gráfico de métricas a partir de otro origen de datos para obtener información sobre cómo graficar una métrica

a partir de un origen de datos. Para obtener información sobre cómo configurar una alarma en una métrica de un origen de datos, consulte Creación de una alarma basada en un origen de datos conectado.

Temas

Administración del acceso a orígenes de datos

CloudWatch utiliza AWS CloudFormation para crear los recursos necesarios en la cuenta. Le recomendamos que utilice la condición `cloudformation:TemplateUrl` para controlar el acceso a las plantillas AWS CloudFormation cuando conceda permisos `CreateStack` a los usuarios de IAM.

> **Warning**
>
> Cualquier usuario al que conceda permiso de invocación a un origen de datos puede consultar las métricas de ese origen de datos, incluso si ese usuario no tiene permisos de IAM directos para acceder al origen de datos. Por ejemplo, si concede permisos `lambda:InvokeFunction` en una función de Lambda de origen de datos de Amazon Managed Service para Prometheus a un usuario, ese usuario podrá consultar las métricas del espacio de trabajo correspondiente de Amazon Managed Service para Prometheus aunque no le haya concedido acceso de IAM directo a ese espacio de trabajo.

Puede encontrar las URL de plantillas para los orígenes de datos en la página Crear pila de la consola de configuración de CloudWatch.

```
{
  "Version":"2012-10-17",
  "Statement":[
    {
      "Effect" : "Allow",
```

```
      "Action" : [ "cloudformation:CreateStack" ],
      "Resource" : "*",
      "Condition" : {
        "StringEquals" : {
          "cloudformation:TemplateUrl" : [ data-source-template-url ]
        }
      }
    }
  ]
}
```

Para obtener más información sobre cómo controlar el acceso AWS CloudFormation, consulte Controlar el acceso con Administración de identidad y acceso de AWS.

Conéctese a un origen de datos prediseñado con un asistente

En este tema se proporcionan instrucciones para usar el asistente para conectar CloudWatch a los siguientes orígenes de datos.

- Amazon OpenSearch Service
- Servicio administrado por Amazon para Prometheus
- Amazon RDS para MySQL
- Amazon RDS para PostgreSQL
- Archivos CSV de Amazon S3
- Microsoft Azure Monitor
- Prometheus

Más adelante en esta sección hay subsecciones con notas sobre la administración y la consulta de cada uno de estos orígenes de datos.

Para crear un conector de un origen de datos

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, seleccione Configuración.
3. Seleccione la pestaña Orígenes de datos de métricas.
4. Elija Crear origen de datos.

5. Seleccione el origen que desee y, a continuación, elija Siguiente.
6. Escriba un nombre para el origen de datos.
7. Introduzca el resto de la información necesaria, en función del origen de datos que haya elegido. Esto puede incluir credenciales para acceder al origen de datos y a la información de identificación del origen de datos, como el nombre del espacio de trabajo de Prometheus, el nombre de la base de datos o el nombre del bucket de Amazon S3. En el caso de los servicios de AWS, el asistente detecta los recursos y los rellena en el menú desplegable de selección.

   Para obtener más información sobre el origen de datos que está utilizando, consulte las secciones posteriores a este procedimiento.
8. Para que CloudWatch se conecte al origen de datos de una VPC, elija Utilizar una VPC y seleccione la VPC que desee utilizar. A continuación, seleccione la subred y el grupo de seguridad.
9. Elija Reconozco que AWS CloudFormation creará recursos de IAM. Este recurso es el rol de ejecución de la función de Lambda.
10. Elija Crear origen de datos.

    El nuevo origen que acaba de añadir no aparecerá hasta que la pila AWS CloudFormation haya terminado de crearlo. Para comprobar el progreso, puede elegir Ver el estado de mi pila de CloudFormation. O puede seleccionar el icono de actualización para actualizar esta lista.

    Cuando el nuevo origen de datos aparezca en esta lista, estará listo para usar. Puede elegir Consultar desde las métricas de CloudWatch para empezar a realizar consultas con ella. Para obtener más información, consulte Creación de un gráfico de métricas a partir de otro origen de datos.

Servicio administrado por Amazon para Prometheus

Actualización de la configuración del origen de datos

- Puede actualizar el origen de datos de forma manual de la siguiente manera:
  - Para actualizar el ID del espacio de trabajo de Amazon Managed Service para Prometheus, actualice la variable de entorno `AMAZON_PROMETHEUS_WORKSPACE_ID` de la función de Lambda del conector del origen de datos.
  - Para actualizar la configuración de la VPC, consulte Configuración del acceso a la VPC (consola) para obtener más información.

Consulta del origen de datos

- Al consultar Amazon Managed Service para Prometheus, después de seleccionar el origen de datos en la pestaña Consulta de orígenes múltiples y seleccionar un conector de Amazon Managed Service para Prometheus, puede utilizar el asistente de consultas para descubrir métricas y etiquetas y realizar consultas PromQL sencillas. También puede utilizar el editor de consultas de PromQL para crear una consulta de PromQL.
- Los conectores de orígenes de datos de CloudWatch no admiten consultas multilínea. Cada fuente de línea se reemplaza por un espacio cuando se ejecuta la consulta o cuando se crea una alarma o un widget de panel con la consulta. En algunos casos, esto puede hacer que la consulta no sea válida. Por ejemplo, si la consulta contiene un comentario de una sola línea, no será válida. Si intenta crear un panel o una alarma con una consulta multilínea desde la línea de comandos o desde Infraestructura como código, la API rechazará la acción y generará un error de análisis.

Amazon OpenSearch Service

Creación del origen de datos

Si el dominio de OpenSearch está permitido para FGAC, debe asignar el rol de ejecución del conector de la función de Lambda a un usuario en OpenSearch Service. Para obtener más información, consulte la sección Asignación de usuarios a roles en Administración de permisos en la documentación de OpenSearch Service.

Si solo se puede acceder a su dominio de OpenSearch desde una Nube Privada Virtual (VPC), debe incluir manualmente una nueva variable de entorno en la función de Lambda llamada `AMAZON_OPENSEARCH_ENDPOINT`. El valor de esta variable debe ser el dominio raíz del punto de conexión de OpenSearch. Puede obtener este dominio raíz quitando `https://` y `<region>.es.amazonaws.com` desde el punto de conexión del dominio que aparece en la consola de OpenSearch Service. Por ejemplo, si punto de conexión de dominio es `https://sample-domain.us-east-1.es.amazonaws.com`, el dominio raíz debería ser `sample-domain`.

Actualización del origen de datos

- Puede actualizar el origen de datos de forma manual de la siguiente manera:
  - Para actualizar el dominio de OpenSearch Service, actualice la variable de entorno `AMAZON_OPENSEARCH_DOMAIN_NAME` de la función de Lambda del conector del origen de datos.

- Para actualizar la configuración de la VPC, consulte Configuración del acceso a la VPC (consola) para obtener más información.

Consulta del origen de datos

- Al consultar OpenSearch Service, después de seleccionar el origen de datos en la pestaña Consulta de orígenes múltiples, haga lo siguiente:
  - Seleccione el índice que desee consultar.
  - Seleccione el nombre de la métrica (cualquier campo numérico del documento) y la estadística.
  - Seleccione el eje de tiempo (cualquier campo de fecha del documento).
  - Seleccione los filtros que desee aplicar (cualquier campo de cadena del documento).
  - Seleccione Consulta gráfica.

Amazon RDS para PostgreSQL y Amazon RDS para MySQL

Creación del origen de datos

- Si solo se puede acceder al origen de datos en una VPC, debe incluir la configuración de VPC para el conector, tal y como se describe en Conéctese a un origen de datos prediseñado con un asistente. Si el origen de datos se va a conectar a la VPC para obtener credenciales, el punto de conexión debe configurarse en la VPC. Para obtener más información, consulte Uso de un punto de conexión de VPC de AWS Secrets Manager.

  Además, debe crear un punto de conexión de VPC para el servicio de Amazon RDS. Para obtener más información, consulte Amazon RDS API e interfaz de los puntos de conexión de VPC (AWS PrivateLink).

Actualización del origen de datos

- Puede actualizar el origen de datos de forma manual de la siguiente manera:
  - Para actualizar la instancia de la base de datos, actualice la variable del entorno `RDS_INSTANCE` de la función de Lambda del conector del origen de datos.
  - Para actualizar el nombre de usuario y la contraseña utilizados para conectarse a Amazon RDS, utilice AWS Secrets Manager. Puede encontrar el ARN del secreto utilizado para el origen de datos en la variable de entorno `RDS_SECRET` de la función de Lambda del origen de datos. Para

obtener más información sobre cómo actualizar el secreto en AWS Secrets Manager, consulte Modificar un secreto en AWS Secrets Manager.

- Para actualizar la configuración de la VPC, consulte Configuración del acceso a la VPC (consola) para obtener más información.

Consulta del origen de datos

- Al realizar consultas en Amazon RDS, después de seleccionar el origen de datos en la pestaña Consulta de orígenes múltiples y seleccionar un conector de Amazon RDS, puede utilizar el buscador de bases de datos para ver las bases de datos, tablas y columnas disponibles. También puede utilizar el editor de SQL para crear una consulta de SQL.

  Puede utilizar las siguientes variables en la consulta:

  - `$start.iso`: la hora de inicio en formato de fecha ISO
  - `$end.iso`: la hora de finalización en formato de fecha ISO
  - `$period`: el período seleccionado en segundos

  Por ejemplo, puede consultar `SELECT value, timestamp FROM table WHERE timestamp BETWEEN $start.iso and $end.iso`
- Los conectores de orígenes de datos de CloudWatch no admiten consultas multilínea. Cada fuente de línea se reemplaza por un espacio cuando se ejecuta la consulta o cuando se crea una alarma o un widget de panel con la consulta. En algunos casos, esto puede hacer que la consulta no sea válida. Por ejemplo, si la consulta contiene un comentario de una sola línea, no será válida. Si intenta crear un panel o una alarma con una consulta multilínea desde la línea de comandos o desde Infraestructura como código, la API rechazará la acción y generará un error de análisis.

> **Note**
>
> Si no se encuentra ningún campo de fecha en los resultados, los valores de cada campo numérico se suman en valores únicos y se representan gráficamente en el intervalo de tiempo proporcionado. Si las marcas temporales no se alinean con el período seleccionado en CloudWatch, los datos se añaden automáticamente mediante `SUM` y se alinean con el período de CloudWatch.

Archivos CSV de Amazon S3

Consulta del origen de datos

- Al consultar los archivos CSV de Amazon S3, después de seleccionar el origen de datos en la pestaña Consulta de orígenes múltiples y seleccionar un conector de Amazon S3, seleccione el bucket y la clave de Amazon S3.

  El archivo CSV debe tener el siguiente formato:

  - La marca de tiempo debe estar en la primera columna.
  - La tabla debe tener una fila de encabezados. Los encabezados se usan para asignar un nombre a las métricas. Se ignorará el título de la columna de fecha y hora; solo se utilizarán los títulos de las columnas de métricas.
  - Las marcas de la hora deben estar en formato de la ISO.
  - Las métricas deben ser campos numéricos.

```
Timestamp, Metric-1, Metric-2, ...
```

A continuación, se muestra un ejemplo:

Marca de tiempo	CPU (%)	Memory (%) (Porcentaje de memoria)	Almacenamiento (%)
2023-11-23T17:09:41+00:00	1	2	3
2023-11-23T17:04:41+00:00	4	5	6
2023-11-23T16:59:41+00:00	7	8	9
2023-11-23T16:54:41+00:00	10	11	12

> **Note**
>
> Si no se proporciona una marca de tiempo, los valores de cada métrica se suman en valores únicos y se representan gráficamente en el intervalo de tiempo proporcionado. Si las marcas temporales no se alinean con el período seleccionado en CloudWatch, los datos se añaden automáticamente mediante `SUM` y se alinean con el período de CloudWatch.

Microsoft Azure Monitor

Creación del origen de datos

- Debe proporcionar la ID de inquilino, ID de cliente y secreto de cliente para conectarse a Microsoft Azure Monitor. Las credenciales se almacenarán en AWS Secrets Manager. Para obtener más información, consulte Crear una aplicación y una entidad principal de servicio de Microsoft Entra que pueda obtener acceso a los recursos en la documentación de Microsoft.

Actualización del origen de datos

- Puede actualizar el origen de datos de forma manual de la siguiente manera:
  - Para actualizar el ID de inquilino, el ID de cliente y el secreto de cliente utilizados para conectarse a Azure Monitor, puede buscar el ARN del secreto utilizado para el origen de datos como variable de entorno `AZURE_CLIENT_SECRET` en la función de Lambda del origen de datos. Para obtener más información sobre cómo actualizar el secreto en AWS Secrets Manager, consulte Modificar un secreto en AWS Secrets Manager.

Consulta del origen de datos

- Al consultar Azure Monitor, después de seleccionar el origen de datos en la pestaña Consulta de orígenes múltiples y seleccionar un conector de Azure Monitor, debe especificar la suscripción de Azure y el grupo de recursos y el recurso. A continuación, puede seleccionar el espacio de nombres, la métrica y la agregación de la métrica y filtrar por dimensiones.

Prometheus

Creación del origen de datos

- Debe proporcionar el punto de conexión de Prometheus y el usuario y la contraseña necesarios para consultar Prometheus. Las credenciales se almacenarán en AWS Secrets Manager.
- Si solo se puede acceder al origen de datos en una VPC, debe incluir la configuración de VPC para el conector, tal y como se describe en Conéctese a un origen de datos prediseñado con un asistente. Si se va a conectar el origen de datos para obtener credenciales, el punto de conexión debe estar configurado en la VPC. Para obtener más información, consulte Uso de un punto de conexión de VPC de AWS Secrets Manager.

Actualización de la configuración del origen de datos

- Puede actualizar el origen de datos de forma manual de la siguiente manera:
  - Para actualizar el punto de conexión de Prometheus, especifique el nuevo punto final como variable de entorno `PROMETHEUS_API_ENDPOINT` en la función de Lambda del origen de datos.
  - Para actualizar el nombre de usuario y la contraseña utilizados para conectarse a Prometheus, puede encontrar el ARN del secreto utilizado para el origen de datos como variable de entorno `PROMETHEUS_API_SECRET` en la función de Lambda del origen de datos. Para obtener más información sobre cómo actualizar el secreto en AWS Secrets Manager, consulte Modificar un secreto en AWS Secrets Manager.
  - Para actualizar la configuración de la VPC, consulte Configuración del acceso a la VPC (consola) para obtener más información.

Consulta del origen de datos

> **Important**
>
> Los tipos de métricas de Prometheus son diferentes de las métricas de CloudWatch y muchas de las métricas disponibles a través de Prometheus son acumulativas por diseño. Cuando consulta las métricas de Prometheus, CloudWatch no aplica ninguna transformación adicional a los datos: si especifica solo el nombre o la etiqueta de la métrica, el valor mostrado será acumulativo. Para obtener más información, consulte Tipos de métricas en la documentación de Prometheus.
>
> Para ver los datos de las métricas de Prometheus como valores discretos, como métricas de CloudWatch, debe editar la consulta antes de ejecutarla. Por ejemplo, puede necesitar añadir una llamada a la función de tasa sobre el nombre de la métrica de Prometheus. Para obtener

documentación sobre la función de velocidad y otras funciones de Prometheus, consulte tasa() en la documentación de Prometheus.

Los conectores de orígenes de datos de CloudWatch no admiten consultas multilínea. Cada fuente de línea se reemplaza por un espacio cuando se ejecuta la consulta o cuando se crea una alarma o un widget de panel con la consulta. En algunos casos, esto puede hacer que la consulta no sea válida. Por ejemplo, si la consulta contiene un comentario de una sola línea, no será válida. Si intenta crear un panel o una alarma con una consulta multilínea desde la línea de comandos o desde Infraestructura como código, la API rechazará la acción y generará un error de análisis.

Notificación de actualizaciones disponibles

De vez en cuando, Amazon puede enviarle una notificación en la que le recomendamos que actualice los conectores con una versión más reciente disponible y le dará instrucciones sobre cómo hacerlo.

Creación de un conector personalizado a un origen de datos

Tiene dos opciones para conectar un origen de datos personalizado a CloudWatch:

- Comience con una plantilla de ejemplo que proporciona CloudWatch. Puede usar JavaScript o Python con esta plantilla. Estas plantillas incluyen un ejemplo de código de Lambda que le resultará útil a la hora de crear la función de Lambda. A continuación, puede modificar la función de Lambda de la plantilla para conectarla al origen de datos personalizado.
- Cree una función AWS Lambda desde cero que implemente el conector del origen de datos, la consulta de datos y la preparación de las series temporales para que CloudWatch las utilice. Esta función debe añadir previamente o combinar puntos de datos si es necesario y también alinear el período y las marcas de tiempo para que sea compatible con CloudWatch.

Contenido

    - Evento DescribeGetMetricData
    - Consideraciones importantes sobre las alarmas de CloudWatch
    - (Opcional) Uso de AWS Secrets Manager para almacenar credenciales
    - (Opcional) Conexión a un origen de datos en una VPC
- Paso 2: crear una política de permisos de Lambda
- Paso 3: asociar la etiqueta de recurso a la función de Lambda

Uso de una plantilla

El uso de una plantilla crea una función de Lambda de muestra y puede ayudar a crear el conector personalizado más rápido. Estas funciones de ejemplo proporcionan códigos de ejemplo para muchos escenarios comunes relacionados con la creación de un conector personalizado. Puede examinar el código de Lambda después de crear un conector con una plantilla y, a continuación, modificarlo para utilizarlo en la conexión al origen de datos.

Además, si usa la plantilla, CloudWatch se encarga de crear la política de permisos de Lambda y de adjuntar etiquetas de recursos a la función de Lambda.

Cómo usar la plantilla para crear un conector a un origen de datos personalizado

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, seleccione Configuración.
3. Seleccione la pestaña Orígenes de datos de métricas.
4. Elija Crear origen de datos.
5. Seleccione el botón de radio de Personalización: plantilla de inicio y, a continuación, seleccione Siguiente.
6. Escriba un nombre para el origen de datos.
7. Seleccione una de las plantillas de la lista.
8. Seleccione Node.js o Python.
9. Elija Crear origen de datos.

   El nuevo origen personalizado que acaba de añadir no aparecerá hasta que la pila AWS CloudFormation haya terminado de crearlo. Para comprobar el progreso, puede elegir Ver el estado de mi pila de CloudFormation. O puede seleccionar el icono de actualización para actualizar esta lista.

Cuando el nuevo origen de datos aparezca en esta lista, estará listo para que lo pruebe en la consola y lo modifique.

10. (Opcional) Para consultar los datos de prueba de este origen en la consola, siga las instrucciones en Creación de un gráfico de métricas a partir de otro origen de datos.
11. Modifique la función de Lambda según sus necesidades.

    a. En el panel de navegación, seleccione Configuración.

    b. Seleccione la pestaña Orígenes de datos de métricas.

    c. Seleccione Ver en la consola de Lambda para el origen que desea modificar.

    Ahora puede modificar la función para obtener acceso al origen de datos. Para obtener más información, consulte Paso 1: crear la función.

    > **Note**
    >
    > Al utilizar la plantilla, al escribir la función de Lambda no es necesario seguir las instrucciones en Paso 2: crear una política de permisos de Lambda o Paso 3: asociar la etiqueta de recurso a la función de Lambda. CloudWatch realizó estos pasos porque utilizó la plantilla.

Creación de un origen de datos personalizado desde cero

Siga los pasos de esta sección para crear una función de Lambda que conecte CloudWatch con un origen de datos.

Paso 1: crear la función

Un conector de origen de datos personalizado debe admitir eventos `GetMetricData` de CloudWatch. Si lo desea, también puede implementar un evento `DescribeGetMetricData` para proporcionar documentación a los usuarios en la consola de CloudWatch sobre cómo usar el conector. La respuesta `DescribeGetMetricData` también se puede usar para establecer los valores predeterminados que se utilizan en el generador de consultas personalizadas de CloudWatch.

CloudWatch proporciona fragmentos de código como ejemplos para ayudarlo a comenzar. Para obtener más información, consulte el repositorio de muestras en https://github.com/aws-samples/cloudwatch-data-source-samples.

Restricciones

- La respuesta de Lambda debe ser inferior a 6 Mb. Si la respuesta supera los 6 Mb, la respuesta `GetMetricData` marca la función de Lambda como `InternalError` y no devuelve ningún dato.
- La función de Lambda debe completar la ejecución en 10 segundos para fines de visualización y creación de cuadros de mando, o en 4,5 segundos para el uso de alarmas. Si el tiempo de ejecución supera ese tiempo, la respuesta `GetMetricData` marca la función de Lambda como `InternalError` y no se devuelve ningún dato.
- La función de Lambda debe enviar su salida mediante marcas de tiempo de época en segundos.
- Si la función de Lambda no vuelve a muestrear los datos y, en cambio, devuelve datos que no corresponden a la hora de inicio ni a la duración del período que solicitó el usuario de CloudWatch, CloudWatch ignora esos datos. Los datos adicionales se descartan de cualquier visualización o alarma. También se descartan todos los datos que no estén entre la hora de inicio y la hora de finalización.

  Por ejemplo, si un usuario solicita datos entre las 10:00 y las 11:00 con un período de 5 minutos, los intervalos de tiempo válidos para devolver los datos son de “10:00:00 a 10:04:59” y “10:05:00 a 10:09:59”. Debe devolver una serie temporal que incluya `10:00 value1`, `10:05 value2`, etc. Si la función devuelve `10:03 valueX`, por ejemplo, se descarta porque las 10:03 no corresponden a la hora ni al período de inicio solicitados.
- Los conectores de orígenes de datos de CloudWatch no admiten consultas multilínea. Cada fuente de línea se reemplaza por un espacio cuando se ejecuta la consulta o cuando se crea una alarma o un widget de panel con la consulta. En algunos casos, esto puede hacer que la consulta no sea válida.

Evento GetMetricData

Carga de solicitud

El siguiente es un ejemplo de una carga de solicitud `GetMetricData` enviada como entrada a la función de Lambda.

```
{
```

```
  "EventType": "GetMetricData",
  "GetMetricDataRequest": {
    "StartTime": 1697060700,
    "EndTime": 1697061600,
    "Period": 300,
    "Arguments": ["serviceregistry_external_http_requests{host_cluster!=\"prod\"}"]
  }
}
```

- StartTime: la marca de tiempo que especifica los primeros datos que se devolverán. El tipo es marca de tiempo, época y segundos.
- EndTime: la marca de tiempo que especifica los últimos datos que se van a devolver. El tipo es marca de tiempo, época y segundos.
- Período: el número de segundos que representa cada agregación de los datos de las métricas. El mínimo es de 60 segundos. El tipo es segundos.
- Argumentos: una matriz de argumentos para pasar a la expresión matemática métrica de Lambda. Para obtener más información sobre cómo pasar argumentos, consulte Cómo pasar argumentos a la función de Lambda.

Carga de respuesta

A continuación, se muestra un ejemplo de una carga de respuesta `GetMetricData` que devuelve la función de Lambda.

```
{
   "MetricDataResults": [
      {
         "StatusCode": "Complete",
         "Label": "CPUUtilization",
         "Timestamps": [ 1697060700, 1697061000, 1697061300 ],
         "Values": [ 15000, 14000, 16000 ]
      }
   ]
}
```

La carga de respuesta contendrá un campo `MetricDataResults` o un campo `Error`, pero no ambos.

Un campo `MetricDataResults` es una lista de campos de series temporales de tipo `MetricDataResult`. Cada uno de esos campos de series temporales puede incluir los siguientes campos.

- StatusCode: (opcional) `Complete` indica que se devolvieron todos los puntos de datos del intervalo de tiempo solicitado. `PartialData` significa que se devolvió un conjunto de puntos de datos incompleto. Si esto se omite, el valor predeterminado es `Complete`.

  Valores válidos: `Complete` | `InternalError` | `PartialData` | `Forbidden`
- Mensajes: lista opcional de mensajes con información adicional sobre los datos devueltos.

  Tipo: matriz de objetos MessageData con cadenas `Code` y `Value`.
- Etiqueta: la etiqueta legible por humanos asociada a los datos.

  Tipo: cadena
- Marcas temporales: las marcas de tiempo de los puntos de datos, formateadas por épocas. El número de marcas de tiempo siempre coincide con el número de valores y el valor de `Timestamps[x]` es `Values[x]`.

  Tipo: matriz de marcas temporales
- Valores: los valores de los puntos de datos de la métrica, correspondientes a `Timestamps`. El número de valores siempre coincide con el número de marcas temporales y el valor de `Timestamps[x]` es `Values[x]`.

  Tipo: matriz de dobles

Para obtener más información acerca de objetos `Error`, consulte las siguientes secciones.

Formatos de respuesta a errores

Si lo desea, puede utilizar la respuesta de error para proporcionar más información sobre los errores. Le recomendamos que devuelva un error de validación del código cuando se produzca un error de validación, por ejemplo, cuando falte un parámetro o sea del tipo incorrecto.

El siguiente es un ejemplo de la respuesta cuando la función de Lambda quiere generar una excepción de validación `GetMetricData`.

```
{
```

```
   "Error": {
      "Code": "Validation",
      "Value": "Invalid Prometheus cluster"
   }
}
```

El siguiente ejemplo muestra la respuesta cuando la función de Lambda indica que no puede devolver datos debido a un problema de acceso. La respuesta se traduce en una serie temporal única con un código de estado de `Forbidden`.

```
{
   "Error": {
      "Code": "Forbidden",
      "Value": "Unable to access ..."
   }
}
```

A continuación se muestra un ejemplo de cuando la función de Lambda genera una excepción `InternalError` general, que se traduce en una única serie temporal con un código de estado `InternalError` y un mensaje. Siempre que un código de error tenga un valor distinto de `Validation` o `Forbidden`, CloudWatch asume que se trata de un error interno genérico.

```
{
   "Error": {
      "Code": "PrometheusClusterUnreachable",
      "Value": "Unable to communicate with the cluster"
   }
}
```

Evento DescribeGetMetricData

Carga de solicitud

El siguiente es un ejemplo de una carga de solicitud de `DescribeGetMetricData`.

```
{
  "EventType": "DescribeGetMetricData"
}
```

Carga de respuesta

El siguiente es un ejemplo de una carga de respuesta de `DescribeGetMetricData`.

```
{
    "Description": "Data source connector",
    "ArgumentDefaults": [{
        Value: "default value"
     }]
}
```

- Descripción: descripción de cómo utilizar el conector del origen de datos. Esta descripción aparecerá en la consola de CloudWatch. Se admite Markdown.

  Tipo: cadena

- ArgumentDefaults: matriz opcional de valores predeterminados de argumentos que se utiliza para rellenar previamente el generador de orígenes de datos personalizados.

  Si se devuelve `[{ Value: "default value 1"}, { Value: 10}]`, el generador de consultas de la consola de CloudWatch muestra dos entradas: la primera con el "valor predeterminado 1" y la segunda con 10.

  Si no se proporciona `ArgumentDefaults`, se muestra una única entrada con el tipo predeterminado establecido en `String`.

  Tipo: matriz de objetos que contiene el valor y el tipo.

- Error: (opcional) se puede incluir un campo de error en cualquier respuesta. Puede ver algunos ejemplos en Evento GetMetricData.

Consideraciones importantes sobre las alarmas de CloudWatch

Si va a utilizar el origen de datos para configurar las alarmas de CloudWatch, debe configurarlo para que notifique los datos con marcas temporales a cada minuto a CloudWatch. Para obtener más información y otras consideraciones sobre la creación de alarmas en las métricas de los orígenes de datos conectados, consulte Creación de una alarma basada en un origen de datos conectado.

(Opcional) Uso de AWS Secrets Manager para almacenar credenciales

Si la función de Lambda necesita usar credenciales para acceder al origen de datos, le recomendamos que utilice AWS Secrets Manager para almacenar estas credenciales en lugar de codificarlas en la función de Lambda. Para obtener más información sobre el uso de AWS Secrets

Manager con Lambda, consulte Utilizar secretos de AWS Secrets Manager en funciones de AWS Lambda.

(Opcional) Conexión a un origen de datos en una VPC

Si el origen de datos se encuentra en una VPC administrada por Amazon Virtual Private Cloud, debe configurar la función de Lambda para tener acceso. Para obtener más información, consulte Conexión de redes salientes a los recursos de una VPC.

Es posible que también necesite configurar los puntos de conexión del servicio de VPC para acceder a servicios como AWS Secrets Manager. Para obtener más información, consulte Acceso a un servicio de AWS a través de un punto de conexión de VPC de interfaz.

Paso 2: crear una política de permisos de Lambda

Debe crear una declaración de política que conceda permiso a CloudWatch para usar la función de Lambda que ha creado. Puede utilizar la AWS CLI o la consola de Lambda para crear la declaración de política.

Cómo usar la AWS CLI para crear la declaración de política

- Escriba el siguiente comando. Sustituya *123456789012* por el ID de la cuenta, sustituya *my-data-source-function* por el nombre de la función de Lambda y sustituya *MyDataSource-DataSourcePermission1234* por un valor único arbitrario.

```
aws lambda add-permission --function-name my-data-source-function --statement-
id MyDataSource-DataSourcePermission1234 --action lambda:InvokeFunction --principal
 lambda.datasource.cloudwatch.amazonaws.com --source-account 123456789012
```

Paso 3: asociar la etiqueta de recurso a la función de Lambda

La consola CloudWatch determina qué funciones de Lambda son conectores de orígenes de datos mediante una etiqueta. Al crear un origen de datos mediante uno de los asistentes, la pila AWS CloudFormation que la configura aplica automáticamente la etiqueta. Al crear un origen de datos usted mismo, puede usar la siguiente etiqueta para la función de Lambda. Esto hace que el conector aparezca en el menú desplegable Orígenes de datos de la consola de CloudWatch al consultar las métricas.

- Una etiqueta con `cloudwatch:datasource` como la clave y `custom` como el valor.

Uso del origen de datos personalizado

Después de crear un origen de datos, puede utilizarlo para consultar los datos de ese origen, visualizarlos y configurar alarmas. Si usó la plantilla para crear el conector de origen de datos personalizado o añadió la etiqueta que aparece en Paso 3: asociar la etiqueta de recurso a la función de Lambda, puede seguir los pasos que se indican en Creación de un gráfico de métricas a partir de otro origen de datos para consultarla.

También puede utilizar la función matemática métrica `LAMBDA` para consultarla, como se explica en la siguiente sección.

Para obtener información sobre cómo crear alarmas en métricas del origen de datos, consulte Creación de una alarma basada en un origen de datos conectado.

Cómo pasar argumentos a la función de Lambda

La forma recomendada de pasar argumentos al origen de datos personalizado es utilizar el generador de consultas de la consola de CloudWatch cuando se consulta el origen de datos.

También puede usar la función de Lambda para recuperar datos del origen de datos mediante la nueva expresión `LAMBDA` en las matemáticas métricas de CloudWatch.

```
LAMBDA("LambdaFunctionName" [, optional-arg]*)
```

`optional-arg` es de hasta 20 cadenas, números o valores booleanos. Por ejemplo, `param`, `3.14` o `true`.

> **Note**
>
> Los conectores de orígenes de datos de CloudWatch no admiten cadenas multilínea. Cada fuente de línea se reemplaza por un espacio cuando se ejecuta la consulta o cuando se crea una alarma o un widget de panel con la consulta. En algunos casos, esto puede hacer que la consulta no sea válida.

Al utilizar la función matemática de métrica `LAMBDA`, puede proporcionar el nombre de la función (`"MyFunction"`). Si la política de recursos lo permite, también puede usar una versión específica de la función (`"MyFunction:22"`) o un alias de función de Lambda (`"MyFunction:MyAlias"`). Imposibilidad de utilizar *

A continuación se muestran algunos ejemplos de llamamiento de la función de LAMBDA.

```
LAMBDA("AmazonOpenSearchDataSource", "MyDomain", "some-query")
```

```
LAMBDA("MyCustomDataSource", true, "fuzzy", 99.9)
```

La función matemática de la métrica LAMBDA devuelve una lista de series temporales que puede devolverse al solicitante o combinarse con otras funciones matemáticas métricas. El siguiente es un ejemplo de combinación de LAMBDA con otras funciones matemáticas métricas.

```
FILL(LAMBDA("AmazonOpenSearchDataSource", "MyDomain", "some-query"), 0)
```

Eliminación de un conector de un origen de datos

Para eliminar un conector de un origen de datos, siga las instrucciones de esta sección.

Cómo eliminar un conector de un origen de datos

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, seleccione Configuración.
3. Seleccione la pestaña Orígenes de datos de métricas.
4. Elija Administrar en CloudFormation en la fila del origen de datos que desea eliminar.

   Accede a la consola de AWS CloudFormation.
5. En la sección con el nombre del origen de datos, elija Eliminar.
6. En el mensaje de confirmación, elija Eliminar.

Recopile las métricas, registros y seguimientos con el agente de CloudWatch

El agente unificado de CloudWatch le permite hacer lo siguiente:

- Recopile métricas internas de nivel de sistema de instancias de Amazon EC2 en distintos sistemas operativos. Las métricas pueden incluir métricas de invitados, además de las métricas de instancias EC2. Las métricas adicionales que se pueden recopilar se indican en Métricas que el agente de CloudWatch ha recopilado.
- Recopilar métricas de nivel de sistema de servidores locales. Pueden incluir servidores en un entorno híbrido, así como servidores no administrados por AWS.
- Recupere métricas personalizadas de sus aplicaciones o servicios mediante los protocolos `collectd` y `StatsD`. `StatsD` se admite tanto en los servidores Linux como en los servidores Windows Server. `collectd` solo se admite en servidores Linux.
- Recopile registros de instancias Amazon EC2 y servidores en las instalaciones con Linux o Windows Server.

  > **Note**
  >
  > El agente de CloudWatch no admite la recopilación de registros de canales FIFO.

- Se puede usar la versión 1.300031.0 y las versiones posteriores para habilitar CloudWatch Application Signals. Para obtener más información, consulte Application Signals.
- Las versiones 1.300025.0 y posteriores pueden recopilar registros de seguimiento de SDK de cliente de OpenTelemetry o de X-Ray y enviarlos a X-Ray.

  El uso del agente de CloudWatch le permite recopilar seguimientos sin necesidad de ejecutar un daemon de recopilación de seguimientos independiente, lo que ayuda a reducir la cantidad de agentes que ejecuta y administra.

Puede almacenar y ver las métricas que ha recopilado con el agente de CloudWatch en CloudWatch del mismo modo que con otras métricas de CloudWatch. El espacio de nombres predeterminado para las métricas que el agente de CloudWatch ha recolectado es `CWAgent`, aunque se puede especificar otro espacio de nombres al configurar el agente.

Los registros que ha recopilado el agente unificado de CloudWatch se procesan y almacenan en Amazon CloudWatch Logs, de la misma manera que los registros que el antiguo agente de CloudWatch Logs ha recopilado. Para obtener más información sobre los precios de CloudWatch Logs, consulte Precios de Amazon CloudWatch.

Las métricas que el agente de CloudWatch ha recopilado se facturan como métricas personalizadas. Para obtener más información sobre los precios de las métricas de CloudWatch, consulte Precios de Amazon CloudWatch.

El agente de CloudWatch es de código abierto bajo la licencia MIT y se hosted on GitHub (aloja en GitHub). Si desea crear, personalizar el agente de CloudWatch o contribuir con él, consulte el repositorio de GitHub para obtener las instrucciones más recientes. Si cree que ha encontrado un problema de seguridad potencial, no lo publique en GitHub ni en ningún foro público. En su lugar, siga las instrucciones de Vulnerability Reporting (Informes de vulnerabilidad) o envíele un correo electrónico a seguridad de AWS directamente.

En los pasos de esta sección se explica cómo instalar el agente unificado de CloudWatch en instancias de Amazon EC2 y servidores en las instalaciones. Para obtener más información acerca de las métricas que puede recopilar el agente de CloudWatch, consulte Métricas que el agente de CloudWatch ha recopilado.

Sistemas operativos compatibles

El agente de CloudWatch es compatible con la arquitectura x86-64 en los siguientes sistemas operativos. También es compatible con todas las actualizaciones de las versiones secundarias de cada una de las versiones principales que se indican aquí.

- Amazon Linux 2023
- Amazon Linux 2
- Ubuntu Server, versiones 23.10, 22.04, 20.04, 18.04, 16.04 y 14.04
- CentOS versiones 9, 8 y 7
- Red Hat Enterprise Linux (RHEL) versiones 9, 8 y 7
- Debian, versiones 12, 11 y 10
- SUSE Linux Enterprise Server (SLES) versión 15 y versión 12
- Oracle Linux, versiones 9, 8 y 7
- Versiones 9 y 8 de AlmaLinux

- Versiones 9 y 8 de Rocky Linux
- Las siguientes computadoras macOS: instancias de EC2 M1 Mac1 y computadoras con macOS 14 (Sonoma), macOS 13 (Ventura) y macOS 12 (Monterey)
- Versiones de 64 bits de Windows Server 2022, Windows Server 2019 y Windows Server 2016
- Windows 10 de 64 bits

El agente es compatible con la arquitectura ARM64 en los siguientes sistemas operativos. También es compatible con todas las actualizaciones de las versiones secundarias de cada una de las versiones principales que se indican aquí.

- Amazon Linux 2023
- Amazon Linux 2
- Ubuntu Server, versiones 23.10, 22.04, 20.04, 18.04 y 16.04
- CentOS versiones 9 y 8
- Red Hat Enterprise Linux (RHEL) versiones 9, 8 y 7
- Debian, versiones 12, 11 y 10
- SUSE Linux Enterprise Server 15
- Las siguientes computadoras macOS: macOS 14 (Sonoma), macOS 13 (Ventura) y macOS 12 (Monterey)

Información general del proceso de instalación

Puede descargar e instalar el agente de CloudWatch manualmente con la línea de comandos o puede integrarlo con SSM. El flujo general de la instalación del agente de CloudWatch con cualquier método es el siguiente:

1. Cree roles o usuarios de IAM o usuarios que permitan que el agente recopile métricas del servidor y opte por integrarse con AWS Systems Manager.
2. Descargue del paquete del agente.
3. Modifique el archivo de configuración del agente de CloudWatch y especifique las métricas que desea recopilar.
4. Instale e inicie el agente en sus servidores. Cuando instale el agente en una instancia EC2, asocie el rol de IAM que ha creado en el paso 1. Cuando instale el agente en un servidor en las

instalaciones, especifique un perfil con nombre que contenga las credenciales del usuario de IAM que ha creado en el paso 1.

Contenido

Instalación del agente de CloudWatch

El agente de CloudWatch está disponible como paquete en Amazon Linux 2023 y Amazon Linux 2. Si está utilizando este sistema operativo, puede instalar el paquete al introducir el siguiente comando. Asimismo, debe asegurarse de que el rol de IAM asociado a la instancia tenga adjunto CloudWatchAgentServerPolicy. Para obtener más información, consulte Cree roles de IAM para utilizarlos con el agente de CloudWatch en instancias de Amazon EC2.

```
sudo yum install amazon-cloudwatch-agent
```

En todos los sistemas operativos compatibles, incluidos Linux y Windows Server, puede descargar e instalar el agente de CloudWatch mediante la línea de comandos con un enlace de descarga de Amazon S3, con Amazon EC2 Systems Manager o mediante una plantilla de AWS CloudFormation. Consulte las secciones siguientes para obtener más detalles.

Contenido

Instalación del agente de CloudWatch con la línea de comandos

Utilice los siguientes temas para descargar, configurar e instalar el paquete del agente de CloudWatch.

Temas

Descargue y configure el agente de CloudWatch con la línea de comandos

Siga los pasos que se describen a continuación para descargar el paquete del agente de CloudWatch, crear roles de IAM o usuarios y, de forma opcional, modificar el archivo de configuración común.

Descargue del paquete de del agente de CloudWatch

> **Note**
>
> Para descargar el agente de CloudWatch, la conexión deben usar la TLS 1.2 o una versión posterior.

El agente de CloudWatch está disponible como paquete en Amazon Linux 2023 y Amazon Linux 2. Si está utilizando este sistema operativo, puede instalar el paquete al introducir el siguiente comando. Asimismo, debe asegurarse de que el rol de IAM asociado a la instancia tenga adjunto CloudWatchAgentServerPolicy. Para obtener más información, consulte Cree roles de IAM y usuarios para utilizarlos con el agente de CloudWatch .

```
sudo yum install amazon-cloudwatch-agent
```

En todos los sistemas operativos compatibles, puede descargar e instalar el agente de CloudWatch mediante la línea de comandos.

Para cada enlace de descarga, hay un enlace general y enlaces para cada región. Por ejemplo, para Amazon Linux 2023, Amazon Linux 2 y la arquitectura x86-64, tres de los enlaces de descarga válidos son:

- `https://amazoncloudwatch-agent.s3.amazonaws.com/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm`
- `https://amazoncloudwatch-agent-us-east-1.s3.us-east-1.amazonaws.com/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm`
- `https://amazoncloudwatch-agent-eu-central-1.s3.eu-central-1.amazonaws.com/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm`

También puede descargar un archivo README sobre los últimos cambios realizados en el agente y un archivo que indique el número de versión que está disponible para su descarga. Estos archivos se encuentran en las siguientes ubicaciones:

- `https://amazoncloudwatch-agent.s3.amazonaws.com/info/latest/RELEASE_NOTES` o `https://amazoncloudwatch-agent-`*`region`*`.s3.`*`region`*`.amazonaws.com/info/latest/RELEASE_NOTES`
- `https://amazoncloudwatch-agent.s3.amazonaws.com/info/latest/CWAGENT_VERSION` o `https://amazoncloudwatch-agent-`*`region`*`.s3.`*`region`*`.amazonaws.com/info/latest/CWAGENT_VERSION`

Arquitectura	Plataforma	Enlace de descarga	Enlace de archivo de firma
x86-64	Amazon Linux 2023 y Amazon Linux 2	https://amazoncloudwatch-agent.s3.amazonaws.com/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm	https://amazoncloudwatch-agent.s3.amazonaws.com/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm.sig
		https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm	https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm.sig
x86-64	Centos	https://amazoncloudwatch-agent.s3.amazonaws.com/	https://amazoncloudwatch-agent.s3.amazonaws.com/

Arquitectura	Plataforma	Enlace de descarga	Enlace de archivo de firma
		centos/amd64/latest/amazon-cloudwatch-agent.rpm https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/centos/amd64/latest/amazon-cloudwatch-agent.rpm	centos/amd64/latest/amazon-cloudwatch-agent.rpm.sig https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/centos/amd64/latest/amazon-cloudwatch-agent.rpm.sig
x86-64	Redhat	https://amazoncloudwatch-agent.s3.amazonaws.com/redhat/amd64/latest/amazon-cloudwatch-agent.rpm https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/redhat/amd64/latest/amazon-cloudwatch-agent.rpm	https://amazoncloudwatch-agent.s3.amazonaws.com/redhat/amd64/latest/amazon-cloudwatch-agent.rpm.sig https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/redhat/amd64/latest/amazon-cloudwatch-agent.rpm.sig
x86-64	SUSE	https://amazoncloudwatch-agent.s3.amazonaws.com/suse/amd64/latest/amazon-cloudwatch-agent.rpm https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/suse/amd64/latest/amazon-cloudwatch-agent.rpm	https://amazoncloudwatch-agent.s3.amazonaws.com/suse/amd64/latest/amazon-cloudwatch-agent.rpm.sig https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/suse/amd64/latest/amazon-cloudwatch-agent.rpm.sig

Arquitectura	Plataforma	Enlace de descarga	Enlace de archivo de firma
x86-64	Debian	https://amazoncloudwatch-agent.s3.amazonaws.com/debian/amd64/latest/amazon-cloudwatch-agent.deb https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/debian/amd64/latest/amazon-cloudwatch-agent.deb	https://amazoncloudwatch-agent.s3.amazonaws.com/debian/amd64/latest/amazon-cloudwatch-agent.deb.sig https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/debian/amd64/latest/amazon-cloudwatch-agent.deb.sig
x86-64	Ubuntu	https://amazoncloudwatch-agent.s3.amazonaws.com/ubuntu/amd64/latest/amazon-cloudwatch-agent.deb https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/ubuntu/amd64/latest/amazon-cloudwatch-agent.deb	https://amazoncloudwatch-agent.s3.amazonaws.com/ubuntu/amd64/latest/amazon-cloudwatch-agent.deb.sig https://s3.*region*.amazonaws.com/amazoncloudwatch-agent-*region*/ubuntu/amd64/latest/amazon-cloudwatch-agent.deb.sig
x86-64	Oracle	https://amazoncloudwatch-agent.s3.amazonaws.com/oracle_linux/amd64/latest/amazon-cloudwatch-agent.rpm https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/oracle_linux/amd64/latest/amazon-cloudwatch-agent.rpm	https://amazoncloudwatch-agent.s3.amazonaws.com/oracle_linux/amd64/latest/amazon-cloudwatch-agent.rpm.sig https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/oracle_linux/amd64/latest/amazon-cloudwatch-agent.rpm.sig

Arquitectura	Plataforma	Enlace de descarga	Enlace de archivo de firma
x86-64	macOS	https://amazoncloudwatch-agent.s3.amazonaws.com/darwin/amd64/latest/amazon-cloudwatch-agent.pkg https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/darwin/amd64/latest/amazon-cloudwatch-agent.pkg	https://amazoncloudwatch-agent.s3.amazonaws.com/darwin/amd64/latest/amazon-cloudwatch-agent.pkg.sig https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/darwin/amd64/latest/amazon-cloudwatch-agent.pkg.sig
x86-64	Windows	https://amazoncloudwatch-agent.s3.amazonaws.com/windows/amd64/latest/amazon-cloudwatch-agent.msi https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/windows/amd64/latest/amazon-cloudwatch-agent.msi	https://amazoncloudwatch-agent.s3.amazonaws.com/windows/amd64/latest/amazon-cloudwatch-agent.msi.sig https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/windows/amd64/latest/amazon-cloudwatch-agent.msi.sig
ARM64	Amazon Linux 2023 y Amazon Linux 2	https://amazoncloudwatch-agent.s3.amazonaws.com/amazon_linux/arm64/latest/amazon-cloudwatch-agent.rpm https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/amazon_linux/arm64/latest/amazon-cloudwatch-agent.rpm	https://amazoncloudwatch-agent.s3.amazonaws.com/amazon_linux/arm64/latest/amazon-cloudwatch-agent.rpm.sig https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/amazon_linux/arm64/latest/amazon-cloudwatch-agent.rpm.sig

Arquitectura	Plataforma	Enlace de descarga	Enlace de archivo de firma
ARM64	Redhat	https://amazoncloudwatch-agent.s3.amazonaws.com/redhat/arm64/latest/amazon-cloudwatch-agent.rpm https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/redhat/arm64/latest/amazon-cloudwatch-agent.rpm	https://amazoncloudwatch-agent.s3.amazonaws.com/redhat/arm64/latest/amazon-cloudwatch-agent.rpm.sig https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/redhat/arm64/latest/amazon-cloudwatch-agent.rpm.sig
ARM64	Ubuntu	https://amazoncloudwatch-agent.s3.amazonaws.com/ubuntu/arm64/latest/amazon-cloudwatch-agent.deb https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/ubuntu/arm64/latest/amazon-cloudwatch-agent.deb	https://amazoncloudwatch-agent.s3.amazonaws.com/ubuntu/arm64/latest/amazon-cloudwatch-agent.deb.sig https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/ubuntu/arm64/latest/amazon-cloudwatch-agent.deb.sig
ARM64	SUSE	https://amazoncloudwatch-agent.s3.amazonaws.com/suse/arm64/latest/amazon-cloudwatch-agent.rpm https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/suse/arm64/latest/amazon-cloudwatch-agent.rpm	https://amazoncloudwatch-agent.s3.amazonaws.com/suse/arm64/latest/amazon-cloudwatch-agent.rpm.sig https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/suse/arm64/latest/amazon-cloudwatch-agent.rpm.sig

Arquitectura	Plataforma	Enlace de descarga	Enlace de archivo de firma
ARM64	MacOS	https://amazoncloudwatch-agent.s3.amazonaws.com/darwin/arm64/latest/amazon-cloudwatch-agent.pkg https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/darwin/arm64/latest/amazon-cloudwatch-agent.pkg	https://amazoncloudwatch-agent.s3.amazonaws.com/darwin/arm64/latest/amazon-cloudwatch-agent.pkg.sig https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/darwin/arm64/latest/amazon-cloudwatch-agent.pkg.sig

Para utilizar la línea de comandos para descargar e instalar el paquete del agente de CloudWatch

1. Descargue el agente de CloudWatch.

   En un servidor Linux, escriba lo siguiente. Para *download-link*, utilice el enlace de descarga adecuado de la tabla anterior.

   ```
   wget download-link
   ```

   En un servidor con Windows Server, descargue el siguiente archivo:

   ```
   https://amazoncloudwatch-agent.s3.amazonaws.com/windows/amd64/latest/amazon-cloudwatch-agent.msi
   ```

2. Después de haber descargado el paquete, puede verificar la firma del paquete. Para obtener más información, consulte Verificación de la firma del paquete del agente de CloudWatch.
3. Instale el paquete. Si ha descargado un paquete RPM en un servidor Linux, cambie al directorio que contiene el paquete y escriba lo siguiente:

   ```
   sudo rpm -U ./amazon-cloudwatch-agent.rpm
   ```

   Si ha descargado un paquete DEB en un servidor Linux, cambie al directorio que contiene el paquete y escriba lo siguiente:

```
sudo dpkg -i -E ./amazon-cloudwatch-agent.deb
```

Si ha descargado un paquete MSI en un servidor con Windows Server, cambie al directorio que contiene el paquete y escriba lo siguiente:

```
msiexec /i amazon-cloudwatch-agent.msi
```

Este comando también funciona desde PowerShell. Para obtener más información acerca de opciones de comandos MSI, consulte Opciones de línea de comandos en la documentación de Microsoft Windows.

Si ha descargado un paquete PKG en un servidor Linux, cambie al directorio que contiene el paquete y escriba lo siguiente:

```
sudo installer -pkg ./amazon-cloudwatch-agent.pkg -target /
```

Cree y modifique el archivo de configuración del agente

Después de haber descargar el agente de CloudWatch, debe crear el archivo de configuración antes de comenzar el agente en algún servidor. Para obtener más información, consulte Cree el archivo de configuración del agente de CloudWatch.

Cree roles de IAM y usuarios para utilizarlos con el agente de CloudWatch

Para obtener acceso a los recursos de AWS se necesitan permisos. Puede crear un rol de IAM, un usuario de IAM o ambos para conceder permisos que el agente de CloudWatch necesita para registrar métricas en CloudWatch. Si va a utilizar el agente en instancias de Amazon EC2, debe crear un rol de IAM. Si va a utilizar el agente servidores en las instalaciones, debe crear un usuario de IAM.

> **Note**
>
> Recientemente, modificamos los siguientes procedimientos con las nuevas políticas `CloudWatchAgentServerPolicy` y `CloudWatchAgentAdminPolicy` creadas por Amazon en lugar de pedir a los clientes que creen estas políticas ellos mismos. Para registrar archivos y descargar archivos desde el almacén de parámetros, las políticas que Amazon ha creado solo admiten archivos con nombres que comiencen con `AmazonCloudWatch-`. Si tiene un archivo de configuración del agente de CloudWatch con un nombre de archivo que

no empieza con `AmazonCloudWatch-`, estas políticas no se pueden utilizar para registrar el archivo en el almacén de parámetros o descargarlo de allí.

Si va a ejecutar el agente de CloudWatch en instancias de Amazon EC2, siga estos pasos para crear el rol de IAM necesario. Este rol proporciona permisos para leer información de la instancia y registrarla en CloudWatch.

Para crear el rol de IAM necesario para ejecutar el agente de CloudWatch en las instancias EC2

1. Inicie sesión en la AWS Management Console y abra la consola de IAM en https://console.aws.amazon.com/iam/.
2. En el panel de navegación de la izquierda, seleccione Roles y, a continuación, seleccione Create Role (Crear roles).
3. Asegúrese de que el servicio de AWS esté seleccionado en Trusted entity type (Tipo de entidad de confianza).
4. Para Use case (Caso de uso), elija EC2 bajo el título Common use cases (Casos de uso comunes).
5. Elija Siguiente.
6. En la lista de políticas, seleccione la casilla junto a CloudWatchAgentServerPolicy. Si es necesario, utilice el cuadro de búsqueda para encontrar la política.
7. (Opcional) Si el agente envía registros de seguimiento a X-Ray, también debe asignar el rol a la política AWSXRayDaemonWriteAccess. Para ello, busque la política en la lista y active la casilla de verificación que hay al lado.
8. Elija Siguiente.
9. En Role name (Nombre del rol), escriba el nombre del rol, como *`CloudWatchAgentServerRole`*. Si lo desea, proporcione una descripción. A continuación, elija Crear rol.

   Se crea el rol.
10. (Opcional) Si el agente va a enviar registros a CloudWatch Logs y desea que el agente pueda establecer políticas de retención para estos grupos de registros, debe agregar el permiso `logs:PutRetentionPolicy` al rol. Para obtener más información, consulte Permitir que el agente de CloudWatch establezca la política de retención de registros.

Si va a ejecutar el agente de CloudWatch en servidores en las instalaciones, siga estos pasos para crear el usuario de IAM necesario.

> **Warning**
>
> En este escenario, se requieren usuarios de IAM con acceso programático y credenciales de larga duración, lo que supone un riesgo de seguridad. Para ayudar a mitigar este riesgo, le recomendamos que brinde a estos usuarios únicamente los permisos que necesitan para realizar la tarea y que los elimine cuando ya no los necesiten. Las claves de acceso se pueden actualizar si es necesario. Para obtener más información, consulte Actualización de claves de acceso en la Guía de usuario de IAM.

Para crear el usuario de IAM necesario para que el agente de CloudWatch se ejecute en servidores en las instalaciones

1. Inicie sesión en la AWS Management Console y abra la consola de IAM en https://console.aws.amazon.com/iam/.
2. En el panel de navegación de la izquierda, elija Users (Usuarios) y, luego, Add users (Agregar usuarios).
3. Escriba el nombre de usuario del nuevo usuario.
4. Seleccione Answer key - Programmatic access (Clave de acceso: acceso mediante programación) y elija Next: Permissions (Siguiente: permisos).
5. Elija Attach existing policies directly (Adjuntar políticas existentes directamente).
6. En la lista de políticas, seleccione la casilla junto a CloudWatchAgentServerPolicy. Si es necesario, utilice el cuadro de búsqueda para encontrar la política.
7. (Opcional) Si el agente va a rastrear a X-Ray, también debe asignar a la función la política AWSXRayDaemonWriteAccess. Para ello, busque la política en la lista y active la casilla de verificación que hay al lado.
8. Elija Siguiente: etiquetas.
9. Si así lo desea, cree etiquetas para el nuevo usuario de IAM y, a continuación, elija Next: Review (Siguiente: revisar).
10. Confirme que se muestra la política correcta y elija Create user (Crear usuario).
11. Junto al nombre del usuario nuevo, elija Show. Copie la clave de acceso y la clave secreta en un archivo para que pueda usarlas al instalar el agente. Elija Close.

Permitir que el agente de CloudWatch establezca la política de retención de registros

Puede configurar el agente de CloudWatch de manera que establezca la política de retención de los grupos de registros a los cuales envía eventos de registro. Si hace esto, debe conceder la `logs:PutRetentionPolicy` al rol o el usuario de IAM que utiliza el agente. El agente utiliza un rol de IAM para ejecutarlo en las instancias de Amazon EC2 y un usuario de IAM para los servidores en las instalaciones.

Para conceder al rol de IAM del agente de CloudWatch permiso para establecer políticas de retención de registros

1. Inicie sesión en AWS Management Console y abra la consola IAM en https://console.aws.amazon.com/iam/.
2. En el panel de navegación izquierdo, seleccione Roles.
3. En el cuadro de búsqueda, escriba el principio del nombre del rol de IAM del agente de CloudWatch. Eligió este nombre cuando creó el rol. Podría llamarse `CloudWatchAgentServerRole`.

   Cuando vea el rol, elija su nombre.
4. En la pestaña Permissions (Permisos), elija Add permissions (Agregar permisos) y, luego, Create inline policy (Crear política insertada).
5. Elija la pestaña JSON y copie la siguiente política en el cuadro, de manera que se reemplaza el JSON predeterminado del cuadro:

   ```
   {
     "Version": "2012-10-17",
     "Statement": [
       {
         "Effect": "Allow",
         "Action": "logs:PutRetentionPolicy",
         "Resource": "*"
       }
     ]
   }
   ```
6. Elija Revisar política.
7. Para Name (Nombre), ingrese **`CloudWatchAgentPutLogsRetention`** o algo similar, y elija Create policy (Crear política).

Para conceder al usuario de IAM del agente de CloudWatch permiso para establecer políticas de retención de registros

1. Inicie sesión en la AWS Management Console y abra la consola de IAM en https://console.aws.amazon.com/iam/.
2. En el panel de navegación izquierdo, elija Users (Usuarios).
3. En el cuadro de búsqueda, escriba el principio del nombre del usuario de IAM del agente de CloudWatch. Eligió este nombre cuando creó el usuario.

   Cuando vea al usuario, elija su nombre.
4. En la pestaña Permissions (Permisos), elija Add inline policy (Añadir política insertada).
5. Elija la pestaña JSON y copie la siguiente política en el cuadro, de manera que se reemplaza el JSON predeterminado del cuadro:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "logs:PutRetentionPolicy",
      "Resource": "*"
    }
  ]
}
```

6. Elija Revisar política.
7. Para Name (Nombre), ingrese **`CloudWatchAgentPutLogsRetention`** o algo similar, y elija Create policy (Crear política).

Instalación y ejecución del agente de CloudWatch en los servidores

Una vez que haya creado el archivo de configuración del agente que desee y un rol o un usuario de IAM, siga estos pasos para instalar y ejecutar el agente en sus servidores con dicha configuración. En primer lugar, asocie un rol o usuario de IAM al servidor que va a ejecutar el agente. A continuación, en ese servidor, descargue el paquete del agente e inícielo utilizando la configuración del agente que ha creado.

Descargue el paquete del agente de CloudWatch mediante un enlace de descarga de S3

> **Note**
>
> Para descargar el agente de CloudWatch, la conexión deben usar la TLS 1.2 o una versión posterior.

Debe instalar el agente en cada servidor en el que vaya a ejecutar el agente.

AMI de Amazon Linux

El agente de CloudWatch está disponible como paquete en Amazon Linux 2023 y Amazon Linux 2. Si está utilizando este sistema operativo, puede instalar el paquete al introducir el siguiente comando. Asimismo, debe asegurarse de que el rol de IAM asociado a la instancia tenga adjunto CloudWatchAgentServerPolicy. Para obtener más información, consulte Cree roles de IAM para utilizarlos con el agente de CloudWatch en instancias de Amazon EC2.

```
sudo yum install amazon-cloudwatch-agent
```

Todos los sistemas operativos

En todos los sistemas operativos compatibles, puede descargar e instalar el agente de CloudWatch mediante la línea de comandos con un enlace de descarga de Amazon S3 tal y como se describe en los siguientes pasos.

Para cada enlace de descarga, hay un enlace general y enlaces para cada región. Por ejemplo, para Amazon Linux 2023, Amazon Linux 2 y la arquitectura x86-64, tres de los enlaces de descarga válidos son:

- `https://amazoncloudwatch-agent.s3.amazonaws.com/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm`
- `https://amazoncloudwatch-agent-us-east-1.s3.us-east-1.amazonaws.com/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm`
- `https://amazoncloudwatch-agent-eu-central-1.s3.eu-central-1.amazonaws.com/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm`

Arquitectura	Plataforma	Enlace de descarga	Enlace de archivo de firma
x86-64	Amazon Linux 2023 y Amazon Linux 2	https://amazoncloudwatch-agent.s3.amazonaws.com/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm	https://amazoncloudwatch-agent.s3.amazonaws.com/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm.sig https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm.sig
x86-64	Centos	https://amazoncloudwatch-agent.s3.amazonaws.com/centos/amd64/latest/amazon-cloudwatch-agent.rpm https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/centos/amd64/latest/amazon-cloudwatch-agent.rpm	https://amazoncloudwatch-agent.s3.amazonaws.com/centos/amd64/latest/amazon-cloudwatch-agent.rpm.sig https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/centos/amd64/latest/amazon-cloudwatch-agent.rpm.sig
x86-64	Redhat	https://amazoncloudwatch-agent.s3.amazonaws.com/redhat/amd64/latest/amazon-cloudwatch-agent.rpm https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/redhat/amd64/latest/amazon-cloudwatch-agent.rpm	https://amazoncloudwatch-agent.s3.amazonaws.com/redhat/amd64/latest/amazon-cloudwatch-agent.rpm.sig https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/redhat/amd64/latest/amazon-cloudwatch-agent.rpm.sig
x86-64	SUSE	https://amazoncloudwatch-agent.s3.amazonaws.com/	https://amazoncloudwatch-agent.s3.amazonaws.com/

Arquitectura	Plataforma	Enlace de descarga	Enlace de archivo de firma
		suse/amd64/latest/amazon-cloudwatch-agent.rpm https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/suse/amd64/latest/amazon-cloudwatch-agent.rpm	suse/amd64/latest/amazon-cloudwatch-agent.rpm.sig https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/suse/amd64/latest/amazon-cloudwatch-agent.rpm.sig
x86-64	Debian	https://amazoncloudwatch-agent.s3.amazonaws.com/debian/amd64/latest/amazon-cloudwatch-agent.deb https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/debian/amd64/latest/amazon-cloudwatch-agent.deb	https://amazoncloudwatch-agent.s3.amazonaws.com/debian/amd64/latest/amazon-cloudwatch-agent.deb.sig https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/debian/amd64/latest/amazon-cloudwatch-agent.deb.sig
x86-64	Ubuntu	https://amazoncloudwatch-agent.s3.amazonaws.com/ubuntu/amd64/latest/amazon-cloudwatch-agent.deb https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/ubuntu/amd64/latest/amazon-cloudwatch-agent.deb	https://amazoncloudwatch-agent.s3.amazonaws.com/ubuntu/amd64/latest/amazon-cloudwatch-agent.deb.sig https://s3.*region*.amazonaws.com/amazoncloudwatch-agent-*region*/ubuntu/amd64/latest/amazon-cloudwatch-agent.deb.sig

Arquitectura	Plataforma	Enlace de descarga	Enlace de archivo de firma
x86-64	Oracle	https://amazoncloudwatch-agent.s3.amazonaws.com/oracle_linux/amd64/latest/amazon-cloudwatch-agent.rpm https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/oracle_linux/amd64/latest/amazon-cloudwatch-agent.rpm	https://amazoncloudwatch-agent.s3.amazonaws.com/oracle_linux/amd64/latest/amazon-cloudwatch-agent.rpm.sig https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/oracle_linux/amd64/latest/amazon-cloudwatch-agent.rpm.sig
x86-64	macOS	https://amazoncloudwatch-agent.s3.amazonaws.com/darwin/amd64/latest/amazon-cloudwatch-agent.pkg https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/darwin/amd64/latest/amazon-cloudwatch-agent.pkg	https://amazoncloudwatch-agent.s3.amazonaws.com/darwin/amd64/latest/amazon-cloudwatch-agent.pkg.sig https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/darwin/amd64/latest/amazon-cloudwatch-agent.pkg.sig
x86-64	Windows	https://amazoncloudwatch-agent.s3.amazonaws.com/windows/amd64/latest/amazon-cloudwatch-agent.msi https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/windows/amd64/latest/amazon-cloudwatch-agent.msi	https://amazoncloudwatch-agent.s3.amazonaws.com/windows/amd64/latest/amazon-cloudwatch-agent.msi.sig https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/windows/amd64/latest/amazon-cloudwatch-agent.msi.sig

Arquitectura	Plataforma	Enlace de descarga	Enlace de archivo de firma
ARM64	Amazon Linux 2023 y Amazon Linux 2	https://amazoncloudwatch-agent.s3.amazonaws.com/amazon_linux/arm64/latest/amazon-cloudwatch-agent.rpm https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/amazon_linux/arm64/latest/amazon-cloudwatch-agent.rpm	https://amazoncloudwatch-agent.s3.amazonaws.com/amazon_linux/arm64/latest/amazon-cloudwatch-agent.rpm.sig https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/amazon_linux/arm64/latest/amazon-cloudwatch-agent.rpm.sig
ARM64	Redhat	https://amazoncloudwatch-agent.s3.amazonaws.com/redhat/arm64/latest/amazon-cloudwatch-agent.rpm https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/redhat/arm64/latest/amazon-cloudwatch-agent.rpm	https://amazoncloudwatch-agent.s3.amazonaws.com/redhat/arm64/latest/amazon-cloudwatch-agent.rpm.sig https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/redhat/arm64/latest/amazon-cloudwatch-agent.rpm.sig
ARM64	Ubuntu	https://amazoncloudwatch-agent.s3.amazonaws.com/ubuntu/arm64/latest/amazon-cloudwatch-agent.deb https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/ubuntu/arm64/latest/amazon-cloudwatch-agent.deb	https://amazoncloudwatch-agent.s3.amazonaws.com/ubuntu/arm64/latest/amazon-cloudwatch-agent.deb.sig https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/ubuntu/arm64/latest/amazon-cloudwatch-agent.deb.sig

Arquitectura	Plataforma	Enlace de descarga	Enlace de archivo de firma
ARM64	SUSE	https://amazoncloudwatch-agent.s3.amazonaws.com/suse/arm64/latest/amazon-cloudwatch-agent.rpm	https://amazoncloudwatch-agent.s3.amazonaws.com/suse/arm64/latest/amazon-cloudwatch-agent.rpm.sig
		https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/suse/arm64/latest/amazon-cloudwatch-agent.rpm	https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/suse/arm64/latest/amazon-cloudwatch-agent.rpm.sig

Para utilizar la línea de comandos para instalar el agente de CloudWatch en una instancia de Amazon EC2

1. Descargue el agente de CloudWatch. En un servidor Linux, escriba lo siguiente. Para *download-link*, utilice el enlace de descarga adecuado de la tabla anterior.

   ```
   wget download-link
   ```

   En el caso de un servidor con Windows Server, descargue el siguiente archivo:

   ```
   https://amazoncloudwatch-agent.s3.amazonaws.com/windows/amd64/latest/amazon-cloudwatch-agent.msi
   ```

2. Después de haber descargado el paquete, puede verificar la firma del paquete. Para obtener más información, consulte Verificación de la firma del paquete del agente de CloudWatch.

3. Instale el paquete. Si ha descargado un paquete RPM en un servidor Linux, cambie al directorio que contiene el paquete y escriba lo siguiente:

   ```
   sudo rpm -U ./amazon-cloudwatch-agent.rpm
   ```

   Si ha descargado un paquete DEB en un servidor Linux, cambie al directorio que contiene el paquete y escriba lo siguiente:

   ```
   sudo dpkg -i -E ./amazon-cloudwatch-agent.deb
   ```

Si ha descargado un paquete MSI en un servidor con Windows Server, cambie al directorio que contiene el paquete y escriba lo siguiente:

```
msiexec /i amazon-cloudwatch-agent.msi
```

Este comando también funciona desde PowerShell. Para obtener más información acerca de opciones de comandos MSI, consulte Opciones de línea de comandos en la documentación de Microsoft Windows.

(Instalación en una instancia EC2) Asociación de un rol de IAM

Para que el agente de CloudWatch pueda enviar datos desde la instancia, debe adjuntar un rol de IAM a la instancia. El rol que se va a asociar es CloudWatchAgentServerRole. Deberías haber creado este rol antes. Para obtener más información, consulte Cree roles de IAM y usuarios para utilizarlos con el agente de CloudWatch.

Para obtener más información sobre cómo adjuntar un rol de IAM a una instancia, consulte Attaching an IAM Role to an Instance (Adjuntar un rol de IAM a una instancia) en la Guía del usuario de Amazon EC2 para instancias de Windows.

(Instalación en un servidor en las instalaciones) Especifique las credenciales de IAM y la Región de AWS

Para que el agente de CloudWatch pueda enviar datos desde un servidor en las instalaciones, debe especificarse la clave de acceso y la clave secreta del usuario de IAM que creó anteriormente. Para obtener más información sobre la creación de este usuario, consulte Cree roles de IAM y usuarios para utilizarlos con el agente de CloudWatch.

También debe especificar la región de AWS a la que se envían las métricas, utilizando el campo `region` de la sección `[AmazonCloudWatchAgent]` del archivo de configuración de AWS, tal y como se muestra en el siguiente ejemplo.

```
[profile AmazonCloudWatchAgent]
region = us-west-1
```

A continuación se muestra un ejemplo del uso del comando `aws configure` para crear un perfil con nombre para el agente de CloudWatch. En este ejemplo se supone que usa el nombre de perfil predeterminado de `AmazonCloudWatchAgent`.

Para crear el perfil AmazonCloudWatchAgent para el agente de CloudWatch

1. Si aún no lo ha hecho, instale AWS Command Line Interface en el servidor. Para obtener más información, consulte Instalación de AWS CLI.
2. En los servidores Linux, escriba el siguiente comando y siga las instrucciones:

   ```
   sudo aws configure --profile AmazonCloudWatchAgent
   ```

   En Windows Server, abra PowerShell como administrador, escriba el siguiente comando y siga las instrucciones.

   ```
   aws configure --profile AmazonCloudWatchAgent
   ```

Verificación del acceso a Internet

Las instancias de Amazon EC2 deben tener acceso a Internet de subida para poder enviar datos a CloudWatch o a CloudWatch Logs. Para obtener más información acerca de cómo configurar el acceso a Internet, consulte Internet Gateways (Gateways de Internet) en la Guía del usuario de Amazon VPC.

Los puntos de enlace y los puertos para configurar en su proxy son los siguientes:

- Si va a utilizar el agente para recopilar métricas, debe añadir a la lista de permitidos los puntos de enlace de CloudWatch para las regiones apropiadas. Estos puntos de conexión se enumeran en los puntos de conexión y las cuotas de Amazon CloudWatch.
- Si va a utilizar el agente para recopilar métricas, debe añadir a la lista de permitidos los puntos de enlace de CloudWatch para las regiones apropiadas. Estos puntos de conexión se enumeran en los puntos de conexión y las cuotas de Registros de Amazon CloudWatch.
- Si va a utilizar el Systems Manager para instalar el agente o el almacén de parámetros para almacenar el archivo de configuración, debe añadir a la lista de permitidos los puntos de enlace del Systems Manager para las regiones apropiadas. Estos puntos de conexión se enumeran en los puntos de conexión y cuotas de AWS Systems Manager.

(Opcional) Modifique la configuración común para la información del proxy o de la región

El agente de CloudWatch incluye un archivo de configuración llamado `common-config.toml`. Si lo desea, puede utilizar este archivo para especificar la información de proxy y región.

En un servidor con Linux, este archivo se encuentra en el directorio `/opt/aws/amazon-cloudwatch-agent/etc`. En un servidor con Windows Server, este archivo se encuentra en el directorio `C:\ProgramData\Amazon\AmazonCloudWatchAgent`.

> **Note**
>
> Le recomendamos que utilice el archivo `common-config.toml` para proporcionar configuraciones y credenciales compartidas cuando ejecute el agente de CloudWatch en modo local, y también puede resultar útil cuando ejecuta Amazon EC2 y desea reutilizar los archivos y perfiles de credenciales compartidas existentes. Habilitarlo mediante el `common-config.toml` tiene la ventaja adicional de que, si el archivo de credenciales compartidas se rota con las credenciales renovadas una vez caducadas, el agente las recogerá automáticamente sin necesidad de reiniciarlas.

El `common-config.toml` predeterminado es el siguiente.

```
# This common-config is used to configure items used for both ssm and cloudwatch access


## Configuration for shared credential.
## Default credential strategy will be used if it is absent here:
##            Instance role is used for EC2 case by default.
##            AmazonCloudWatchAgent profile is used for the on-premises case by
 default.
# [credentials]
#    shared_credential_profile = "{profile_name}"
#    shared_credential_file= "{file_name}"

## Configuration for proxy.
## System-wide environment-variable will be read if it is absent here.
## i.e. HTTP_PROXY/http_proxy; HTTPS_PROXY/https_proxy; NO_PROXY/no_proxy
## Note: system-wide environment-variable is not accessible when using ssm run-command.
## Absent in both here and environment-variable means no proxy will be used.
# [proxy]
#    http_proxy = "{http_url}"
#    https_proxy = "{https_url}"
#    no_proxy = "{domain}"
```

Inicialmente, todas las líneas están convertidas en comentarios. Para establecer la configuración del perfil de credenciales o del proxy, quite # de esa línea y especifique un valor. Puede editar este archivo manualmente o mediante Run Command `RunShellScript` en Systems Manager:

- `shared_credential_profile`: para los servidores locales, esta línea especifica el perfil de credenciales del usuario de IAM que se va a utilizar para enviar datos a CloudWatch. Si mantiene esta línea dentro de los comentarios, se utiliza `AmazonCloudWatchAgent`. Para obtener más información sobre la creación de este perfil, consulte (Instalación en un servidor en las instalaciones) Especifique las credenciales de IAM y la Región de AWS.

  En una instancia EC2, puede utilizar esta línea para que el agente de CloudWatch envíe datos desde esta instancia a CloudWatch en una Región de AWS diferente. Para ello, especifique un perfil con nombre que incluya un campo `region` especificando el nombre de la región a la que se van a enviar los datos.

  Si especifica un `shared_credential_profile`, también debe eliminar el # desde el principio de la línea `[credentials]`.
- `shared_credential_file`: para que el agente busque credenciales en un archivo ubicado en una ruta distinta de la ruta predeterminada, especifique esa ruta completa y el nombre de archivo aquí. La ruta predeterminada es `/root/.aws` en Linux y `C:\\Users\\Administrator\\.aws` en Windows Server.

  El primer ejemplo incluido a continuación muestra la sintaxis de una línea `shared_credential_file` válida para los servidores Linux y el segundo ejemplo es válido para Windows Server. En Windows Server, debe aplicar escape a los caracteres \.

  ```
  shared_credential_file= "/usr/username/credentials"
  ```

  ```
  shared_credential_file= "C:\\Documents and Settings\\username\\.aws\\credentials"
  ```

  Si especifica un `shared_credential_file`, también debe eliminar el # desde el principio de la línea `[credentials]`.
- Configuración del proxy: si los servidores usan proxies HTTP o HTTPS para contactarse con los servicios de AWS, especifíquelos en los campos `http_proxy` y `https_proxy`. Si hay URL que se deben excluir del proxy, especifíquelas en el campo `no_proxy`, separadas por comas.

Inicie el agente de CloudWatch con la línea de comandos

Siga estos pasos para utilizar la línea de comandos para iniciar el agente de CloudWatch en un servidor.

Para utilizar la línea de comandos para iniciar el agente de CloudWatch en un servidor

1. Copie el archivo de configuración del agente que desea utilizar en el servidor donde va a ejecutar el agente. Anote el nombre de la ruta donde lo va a copiar.
2. En este comando, `-a fetch-config` provoca que el agente cargue la última versión del archivo de configuración del agente de CloudWatch y `-s` inicia el agente.

   Especifique uno de los siguientes comandos. Reemplace *configuration-file-path* por la ruta al archivo de configuración del agente. Este archivo se denomina `config.json` si se creó con el asistente y podría llamarse `amazon-cloudwatch-agent.json` si se creó manualmente.

   En una instancia EC2 que ejecute Linux, escriba el siguiente comando.

   ```
   sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-
   config -m ec2 -s -c file:configuration-file-path
   ```

   En un servidor local que ejecute Linux, escriba lo siguiente:

   ```
   sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-
   config -m onPremise -s -c file:configuration-file-path
   ```

   En una instancia EC2 que ejecute Windows Server, escriba lo siguiente desde la consola de PowerShell:

   ```
   & "C:\Program Files\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent-ctl.ps1" -
   a fetch-config -m ec2 -s -c file:configuration-file-path
   ```

   En un servidor local que ejecute Windows Server, escriba lo siguiente desde la consola de PowerShell:

   ```
   & "C:\Program Files\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent-ctl.ps1" -
   a fetch-config -m onPremise -s -c file:configuration-file-path
   ```

Instalación del agente de CloudWatch mediante AWS Systems Manager

Utilice los siguientes temas para instalar y ejecutar el agente de CloudWatch mediante AWS Systems Manager.

Temas

Cree roles y usuarios de IAM para usarlos con el agente de CloudWatch

Para obtener acceso a los recursos de AWS se necesitan permisos. Puede crear roles y usuarios de IAM que incluyan los permisos necesarios para que el agente de CloudWatch registre métricas en CloudWatch y para que se comunique con Amazon EC2 y AWS Systems Manager. Los roles de IAM se utilizan en las instancias de Amazon EC2 y los usuarios de IAM en los servidores en las instalaciones.

Un rol o un usuario permite que el agente de CloudWatch se instale en un servidor y envíe métricas a CloudWatch. El otro rol o usuario es necesario para almacenar la configuración del agente de CloudWatch en el almacén de parámetros de Systems Manager. El almacén de parámetros permite que varios servidores usen una configuración del agente de CloudWatch.

La posibilidad de hacer registros en el almacén de parámetros es un permiso amplio y poderoso. Solo debe utilizarse cuando sea necesario y no debería asociarse a varias instancias de la implementación. Si almacena la configuración del agente de CloudWatch en el almacén de parámetros, se recomienda lo siguiente:

- Configure la instancia en la que realizará esta configuración.
- Utilice el rol de IAM con permisos para hacer registros en el almacén de parámetros únicamente en esta instancia.
- Utilice el rol de IAM con permisos para hacer registros en el almacén de parámetros únicamente mientras esté trabajando y guardando el archivo de configuración del agente de CloudWatch.

> **Note**
>
> Recientemente, modificamos los siguientes procedimientos con las nuevas políticas `CloudWatchAgentServerPolicy` y `CloudWatchAgentAdminPolicy` creadas por Amazon en lugar de pedir a los clientes que creen estas políticas ellos mismos. Para utilizar estas políticas para registrar el archivo de configuración del agente en el almacén de parámetros y luego descargarlo desde allí, el archivo de configuración del agente debe tener un nombre que empiece con `AmazonCloudWatch-`. Si tiene un archivo de configuración del agente de CloudWatch cuyo nombre no comience con `AmazonCloudWatch-`, estas políticas no se pueden utilizar para registrar el archivo en el almacén de parámetros ni para descargarlo de allí.

Cree roles de IAM para utilizarlos con el agente de CloudWatch en instancias de Amazon EC2

El primer procedimiento crea el rol de IAM que debe adjuntar a cada instancia de Amazon EC2 que ejecuta el agente de CloudWatch. Este rol proporciona permisos para leer la información de la instancia y registrarla en CloudWatch.

El segundo procedimiento crea el rol de IAM que debe adjuntar a la instancia de Amazon EC2 que se ha usado para crear el archivo de configuración del agente de CloudWatch. Este paso es necesario si va a almacenar este archivo en el almacén de parámetros de Systems Manager para que otros servidores puedan utilizarlo. Este rol proporciona permisos para registrarlos en el almacén de parámetros, además de los permisos para leer la información de la instancia y registrarla en CloudWatch. Este rol incluye los permisos suficientes para ejecutar el agente de CloudWatch, así como para realizar escrituras en el almacén de parámetros.

> **Note**
>
> El almacén de parámetros admite parámetros en los niveles estándar y avanzado. Estos niveles de parámetros no están relacionados con los niveles de detalles básico, estándar y avanzado disponibles con los conjuntos de métricas predefinidos del agente de CloudWatch.

Para crear el rol de IAM necesario para que cada servidor ejecute el agente de CloudWatch

1. Inicie sesión en la AWS Management Console y abra la consola de IAM en https://console.aws.amazon.com/iam/.

2. En el panel de navegación, seleccione Roles y, a continuación, seleccione Crear rol.
3. En Seleccionar tipo de entidad de confianza, seleccione Servicio de AWS.
4. Seguido a Common use cases (Casos de uso comunes), elija EC2, y, a continuación, elija Next: Permissions (Siguiente: Permisos).
5. En la lista de políticas, seleccione la casilla junto a CloudWatchAgentServerPolicy. Si es necesario, utilice el cuadro de búsqueda para encontrar la política.
6. Para utilizar Systems Manager para instalar o configurar el agente de CloudWatch, seleccione la casilla situada junto a AmazonSSMManagedInstanceCore. Esta política administrada por AWS permite que una instancia utilice la funcionalidad básica del servicio Systems Manager. Si es necesario, utilice el cuadro de búsqueda para encontrar la política. Esta política no es necesaria si inicia y configura el agente solo a través de la línea de comandos.
7. Elija Next: Tags (Siguiente: Etiquetas).
8. (Opcional) Añada uno o varios pares de clave de etiqueta-valor para organizar, realizar un seguimiento o controlar el acceso a este rol y, a continuación, elija Next: Review (Siguiente: Revisar).
9. En Role name (Nombre del rol), escriba un nombre para el rol nuevo (por ejemplo, **`CloudWatchAgentServerRole`** o el nombre que prefiera).
10. (Opcional) En Role description (Descripción del rol), escriba una descripción.
11. Compruebe que CloudWatchAgentServerPolicy y, de forma opcional, AmazonSSMManagedInstanceCore aparecen junto a Policies (Políticas).
12. Elija Crear rol.

    Se crea el rol.

El siguiente procedimiento crea el rol de IAM que también puede realizar escrituras en el almacén de parámetros. Se puede utilizar este rol para almacenar el archivo de configuración del agente en el almacén de parámetros para que otros servidores pueden recuperarlo.

Los permisos para realizar escrituras en el almacén de parámetros proporcionan un acceso amplio. Este rol no debería asociarse a todos los servidores y solo los administradores deben utilizarlo. Cuando termine de crear el archivo de configuración del agente y lo copie en el almacén de parámetros, debe desconectar este rol de la instancia y utilizar `CloudWatchAgentServerRole` en su lugar.

Para crear el rol de IAM para que un administrador realice escrituras en el almacén de parámetros

1. Inicie sesión en la AWS Management Console y abra la consola de IAM en https://console.aws.amazon.com/iam/.
2. En el panel de navegación, seleccione Roles y, a continuación, seleccione Crear rol.
3. En Seleccionar tipo de entidad de confianza, seleccione Servicio de AWS.
4. Justo debajo de Choose the service that will use this role (Elegir el servicio que utilizará este rol), elija EC2 y, a continuación, elija Next: Permissions (Siguiente: Permisos).
5. En la lista de políticas, seleccione la casilla junto a CloudWatchAgentAdminPolicy. Si es necesario, utilice el cuadro de búsqueda para encontrar la política.
6. Para utilizar Systems Manager para instalar o configurar el agente de CloudWatch, seleccione la casilla situada junto a AmazonSSMManagedInstanceCore. Esta política administrada por AWS permite que una instancia utilice la funcionalidad básica del servicio Systems Manager. Si es necesario, utilice el cuadro de búsqueda para encontrar la política. Esta política no es necesaria si inicia y configura el agente solo a través de la línea de comandos.
7. Elija Next: Tags (Siguiente: Etiquetas).
8. (Opcional) Añada uno o varios pares de clave de etiqueta-valor para organizar, realizar un seguimiento o controlar el acceso a este rol y, a continuación, elija Next: Review (Siguiente: Revisar).
9. En Role name (Nombre del rol), escriba un nombre para el rol nuevo (por ejemplo, **`CloudWatchAgentAdminRole`** o el nombre que prefiera).
10. (Opcional) En Role description (Descripción del rol), escriba una descripción.
11. Compruebe que CloudWatchAgentAdminPolicy y, de forma opcional, AmazonSSMManagedInstanceCore aparecen junto a Policies (Políticas).
12. Elija Crear rol.

    Se crea el rol.

Cree usuarios de IAM para utilizarlos con el agente de CloudWatch en servidores en las instalaciones

El primer procedimiento crea el usuario de IAM que se necesita para ejecutar el agente de CloudWatch. Este usuario proporciona los permisos para enviar datos a CloudWatch.

El segundo procedimiento crea el usuario de IAM que se puede utilizar al crear el archivo de configuración del agente de CloudWatch. Utilice este procedimiento para almacenar este archivo en

el almacén de parámetros de Systems Manager para que otros servidores pueden utilizarlo. Este usuario proporciona los permisos para realizar escrituras en el almacén de parámetros, además de permisos para registrar datos en CloudWatch.

> **Note**
>
> El almacén de parámetros admite parámetros en los niveles estándar y avanzado. Estos niveles de parámetros no están relacionados con los niveles de detalles básico, estándar y avanzado que están disponibles con los conjuntos de métricas predefinidos del agente de CloudWatch .

Para crear el usuario de IAM necesario para que el agente de CloudWatch registre datos en CloudWatch

1. Inicie sesión en la AWS Management Console y abra la consola de IAM en https://console.aws.amazon.com/iam/.
2. En el panel de navegación, elija Users y luego elija la opción Add user.
3. Escriba el nombre de usuario del nuevo usuario.
4. En Access type (Tipo de acceso), elija Programmatic access (Acceso mediante programación) y, a continuación, elija Next: Permissions (Siguiente: Permisos).
5. En la página Set permissions (Establecer permisos), elija Attach existing policies directly (Asociar directamente las políticas existentes).
6. En la lista de políticas, seleccione la casilla junto a CloudWatchAgentServerPolicy. Si es necesario, utilice el cuadro de búsqueda para encontrar la política.
7. Para utilizar Systems Manager para instalar o configurar el agente de CloudWatch, seleccione la casilla situada junto a AmazonSSMManagedInstanceCore. Esta política administrada por AWS permite que una instancia utilice la funcionalidad básica del servicio Systems Manager. (De ser necesario, utilice el cuadro de búsqueda para hallar la política. La política no es necesaria si inicia y configura el agente solo a través de la línea de comandos).
8. Elija Next: Tags (Siguiente: Etiquetas).
9. (Opcional) Añada uno o varios pares de clave de etiqueta-valor para organizar, realizar un seguimiento o controlar el acceso a este rol y, a continuación, elija Next: Review (Siguiente: Revisar).
10. Compruebe que se muestran las políticas correctas y, a continuación, elija Create user (Crear usuario).

11. En la fila del usuario nuevo, elija Show (Mostrar). Copie la clave de acceso y la clave secreta en un archivo para que pueda usarlas al instalar el agente. Elija Close.

El siguiente procedimiento crea el usuario de IAM que también puede realizar escrituras en el almacén de parámetros. Si va a almacenar el archivo de configuración del agente en el almacén de parámetros para que otros servidores pueden utilizarlo, necesita utilizar este usuario de IAM. Este usuario de IAM proporciona permisos para realizar escrituras en el almacén de parámetros. Este usuario también proporciona permisos para leer la información de la instancia y registrarla en CloudWatch. Los permisos para realizar escrituras en el almacén de parámetros de Systems Manager proporcionan un acceso amplio. Este usuario de IAM no debería adjuntarse a todos los servidores y solo los administradores deberían utilizarlo. Debería utilizar este usuario de IAM solo cuando vaya a almacenar el archivo de configuración del agente en el almacén de parámetros.

Para crear el usuario de IAM necesario para almacenar el archivo de configuración en el almacén de parámetros y enviar información a CloudWatch

1. Inicie sesión en la AWS Management Console y abra la consola de IAM en https://console.aws.amazon.com/iam/.
2. En el panel de navegación, elija Users y luego elija la opción Add user.
3. Escriba el nombre de usuario del nuevo usuario.
4. En Access type (Tipo de acceso), elija Programmatic access (Acceso mediante programación) y, a continuación, elija Next: Permissions (Siguiente: Permisos).
5. En la página Set permissions (Establecer permisos), elija Attach existing policies directly (Asociar directamente las políticas existentes).
6. En la lista de políticas, seleccione la casilla junto a CloudWatchAgentAdminPolicy. Si es necesario, utilice el cuadro de búsqueda para encontrar la política.
7. Para utilizar Systems Manager con el fin de instalar o configurar el agente de CloudWatch, seleccione la casilla de verificación situada junto a AmazonSSMManagedInstanceCore. Esta política administrada por AWS permite que una instancia utilice la funcionalidad básica del servicio Systems Manager. (De ser necesario, utilice el cuadro de búsqueda para hallar la política. La política no es necesaria si inicia y configura el agente solo a través de la línea de comandos).
8. Elija Next: Tags (Siguiente: Etiquetas).

9. (Opcional) Añada uno o varios pares de clave de etiqueta-valor para organizar, realizar un seguimiento o controlar el acceso a este rol y, a continuación, elija Next: Review (Siguiente: Revisar).
10. Compruebe que se muestran las políticas correctas y, a continuación, elija Create user (Crear usuario).
11. En la fila del usuario nuevo, elija Show (Mostrar). Copie la clave de acceso y la clave secreta en un archivo para que pueda usarlas al instalar el agente. Elija Close.

Descargue y configure el agente de CloudWatch

En esta sección se explica cómo utilizar Systems Manager para descargar el agente y luego crear el archivo de configuración del agente. Antes de utilizar Systems Manager para descargar el agente, debe asegurarse de que la instancia esté configurada correctamente para Systems Manager.

Instalación o actualización de SSM Agent

En una instancia de Amazon EC2, el agente de CloudWatch requiere que la instancia ejecute la versión 2.2.93.0 o una posterior. Antes de instalar el agente de CloudWatch, actualice o instale el SSM Agent en la instancia si aún no lo ha hecho.

Para obtener información acerca de la instalación o la actualización de SSM Agent en una instancia con Linux, consulte Installing and Configuring SSM Agent on Linux Instances (Instalación y configuración de SSM Agent en instancias de Linux) en la Guía del usuario de AWS Systems Manager.

Para obtener información sobre la instalación o la actualización de SSM Agent, consulte Working with SSM Agent (Ejecución con SSM Agent) en la Guía del usuario AWS Systems Manager.

(Opcional) Verifique los prerrequisitos de Systems Manager

Verificación del acceso a Internet

Las instancias de Amazon EC2 deben tener acceso a Internet de subida para poder enviar datos a CloudWatch o a CloudWatch Logs. Para obtener más información acerca de cómo configurar el acceso a Internet, consulte Internet Gateways (Gateways de Internet) en la Guía del usuario de Amazon VPC.

Los puntos de enlace y los puertos para configurar en su proxy son los siguientes:

- Si va a utilizar el agente para recopilar métricas, debe agregar a la lista de puntos de conexión de CloudWatch para las regiones apropiadas. Estos puntos de enlace se enumeran en Amazon CloudWatch en la Referencia general de Amazon Web Services.
- Si va a utilizar el agente para recopilar métricas, debe agregar a la lista de puntos de conexión de registro CloudWatch para las regiones apropiadas. Estos puntos de enlace se enumeran en Registros de Amazon CloudWatch en la Referencia general de Amazon Web Services.
- Si va a utilizar Systems Manager para instalar el agente o el almacén de parámetros para almacenar el archivo de configuración, debe agregar a la lista de permitidos los puntos de conexión de Systems Manager para las regiones apropiadas. Estos puntos de enlace se enumeran en AWS Systems Manager en la Referencia general de Amazon Web Services.

Siga estos pasos para descargar el paquete del agente de CloudWatch mediante Systems Manager.

Para descargar el agente de CloudWatch mediante el Systems Manager

1. Abra la consola de Administrador de sistemas en https://console.aws.amazon.com/systems-manager/.
2. En el panel de navegación, elija Ejecutar comando.

   -o bien-

   Si la página de inicio de AWS Systems Manager se abre, desplácese hacia abajo y elija Explore Run Command (Explorar Run Command).
3. Elija Run command (Ejecutar comando).
4. En la lista Command document, elija AWS-ConfigureAWSPackage.
5. En el área Destinos, elija la instancia en la que va a instalar el agente de CloudWatch. Si no ve una instancia específica, puede que no esté configurada como instancia administrada para su uso con Systems Manager. Para obtener más información, consulte Configuración de AWS Systems Manager para entornos híbridos en la Guía del usuario de AWS Systems Manager.
6. En la lista Action (Acción), elija Install (Instalar).
7. En el campo Name (Nombre), escriba *AmazonCloudWatchAgent*.
8. Deje la opción Version (Versión) establecida en latest (más reciente) para instalar la última versión del agente.
9. Elija Ejecutar.

10. Opcionalmente, en las áreas de Targets and outputs (Destinos y salidas), seleccione el botón situado junto a un nombre de instancia y elija View output (Ver salidas). Systems Manager debería mostrar que el agente se ha instalado correctamente.

Cree y modifique el archivo de configuración del agente

Después de haber descargado el agente de CloudWatch, debe crear el archivo de configuración antes de iniciar el agente en algún servidor.

Si va a guardar el archivo de configuración del agente en el almacén de parámetros de Systems Manager, debe utilizar una instancia EC2 para guardarlo en dicho almacén. Además, primero debe adjuntar el rol de IAM `CloudWatchAgentAdminRole` a dicha instancia. Para obtener más información acerca de cómo se adjuntan los roles de IAM, consulte Attaching an IAM Role to an Instance (Adjuntar un rol de IAM a una instancia) en la Guía del usuario de Amazon EC2 para instancias de Windows.

Para obtener más información sobre la creación del archivo de configuración del agente de CloudWatch, consulte Cree el archivo de configuración del agente de CloudWatch.

Instalación del agente de CloudWatch en instancias EC2 mediante la configuración del agente

Luego de haber guardado una configuración del agente de CloudWatch en el almacén de parámetros, se podrá utilizar cuando se instale el agente en otros servidores.

Temas

- Cómo se adjunta un rol de IAM a la instancia
- Descargue del paquete del agente de CloudWatch en una instancia de Amazon EC2
- (Opcional) Modifique la configuración común y el perfil con nombre para el agente de CloudWatch
- Inicie el agente de CloudWatch

Cómo se adjunta un rol de IAM a la instancia

Debe adjuntar el rol de IAM CloudWatchAgentServerRole en la instancia EC2 para que el agente de CloudWatch se pueda ejecutar en la instancia. Este rol permite que el agente de CloudWatch realice acciones en la instancia. Deberías haber creado este rol antes. Para obtener más información, consulte Cree roles de IAM y usuarios para utilizarlos con el agente de CloudWatch.

Para obtener más información, consulte Attaching an IAM Role to an Instance (Adjuntar un rol de IAM a una instancia) en la Guía del usuario de Amazon EC2 para instancias de Windows.

Descargue del paquete del agente de CloudWatch en una instancia de Amazon EC2

Debe instalar el agente en cada servidor en el que vaya a ejecutar el agente. El agente de CloudWatch está disponible como paquete en Amazon Linux 2023 y Amazon Linux 2. Si está utilizando este sistema operativo, puede instalar el paquete al introducir el siguiente comando. Asimismo, debe asegurarse de que el rol de IAM asociado a la instancia tenga adjunto CloudWatchAgentServerPolicy. Para obtener más información, consulte Cree roles de IAM para utilizarlos con el agente de CloudWatch en instancias de Amazon EC2.

```
sudo yum install amazon-cloudwatch-agent
```

En todos los sistemas operativos compatibles, puede descargar el paquete del agente de CloudWatch mediante Run Command de Systems Manager o un enlace de descarga de Amazon S3. Para obtener más información acerca del uso del enlace de descarga de Amazon S3, consulte Descargue del paquete de del agente de CloudWatch.

> **Note**
>
> Cuando instala o actualiza el agente de CloudWatch, solo se admite la opción Uninstall and reinstall (Desinstalar e instalar). No puede usar la opción In-place update (Actualización in situ).

Descargue el agente de CloudWatch en una instancia de Amazon EC2 mediante Systems Manager

Antes de utilizar Systems Manager para instalar el agente de CloudWatch, debe asegurarse de que la instancia esté configurada correctamente para Systems Manager.

Instalación o actualización de SSM Agent

En una instancia de Amazon EC2, el agente de CloudWatch requiere que la instancia ejecute la versión 2.2.93.0 o una posterior. Antes de instalar el agente de CloudWatch, actualice o instale el SSM Agent en la instancia si aún no lo ha hecho.

Para obtener información sobre la instalación o actualización de SSM Agent en una instancia con Linux, consulte Installing and Configuring the SSM Agent on Linux Instances (Instalación y

configuración de SSM Agent en instancias de Linux) en la Guía del usuario de AWS Systems Manager.

Para obtener información acerca de la instalación o la actualización de SSM Agent en una instancia con Windows Server, consulte Installing and Configuring SSM Agent on Windows Instances (Instalación y configuración de SSM Agent en instancias de Windows) en la Guía del usuario de AWS Systems Manager.

(Opcional) Verifique los prerrequisitos de Systems Manager

Antes de utilizar Run Command de Systems Manager para instalar y configurar el agente de CloudWatch, verifique que las instancias cumplen los requisitos mínimos de Systems Manager. Para obtener más información, consulte Configuración de AWS Systems Manager en la Guía del usuario de AWS Systems Manager.

Verificación del acceso a Internet

Las instancias de Amazon EC2 deben tener acceso a Internet de subida con el fin de enviar datos a CloudWatch o a CloudWatch Logs. Para obtener más información acerca de cómo configurar el acceso a Internet, consulte Internet Gateways (Gateways de Internet) en la Guía del usuario de Amazon VPC.

Descargue el paquete del agente de CloudWatch

Run Command de Systems Manager le permite administrar la configuración de las instancias. Puede especificar un documento de Systems Manager, especificar parámetros y ejecutar el comando en una o varias instancias. SSM Agent procesa el comando en la instancia y configura la instancia tal y como se especifica.

Para descargar el agente de CloudWatch mediante Run Command

1. Abra la consola de Administrador de sistemas en https://console.aws.amazon.com/systems-manager/.
2. En el panel de navegación, elija Ejecutar comando.

   -o bien-

   Si la página de inicio de AWS Systems Manager se abre, desplácese hacia abajo y elija Explore Run Command (Explorar Run Command).
3. Elija Run command (Ejecutar comando).

4. En la lista Command document, elija AWS-ConfigureAWSPackage.
5. En el área Targets (Destinos), elija la instancia en la que instalar el agente de CloudWatch. Si no se ve una instancia específica, es posible que no esté configurada para Run Command. Para obtener más información, consulte Configuración de AWS Systems Manager para entornos híbridos en la Guía del usuario de AWS Systems Manager.
6. En la lista Action (Acción), elija Install (Instalar).
7. En el cuadro Name (Nombre), escriba *AmazonCloudWatchAgent*.
8. Deje la opción Version (Versión) establecida en latest (más reciente) para instalar la última versión del agente.
9. Elija Ejecutar.
10. Opcionalmente, en las áreas de Targets and outputs (Destinos y salidas), seleccione el botón situado junto a un nombre de instancia y elija View output (Ver salidas). Systems Manager debería mostrar que el agente se ha instalado correctamente.

(Opcional) Modifique la configuración común y el perfil con nombre para el agente de CloudWatch

El agente de CloudWatch incluye un archivo de configuración llamado `common-config.toml`. Si lo desea, puede utilizar este archivo para especificar la información de proxy y región.

En un servidor con Linux, este archivo se encuentra en el directorio `/opt/aws/amazon-cloudwatch-agent/etc`. En un servidor con Windows Server, este archivo se encuentra en el directorio `C:\ProgramData\Amazon\AmazonCloudWatchAgent`.

El `common-config.toml` predeterminado es el siguiente:

```
# This common-config is used to configure items used for both ssm and cloudwatch access


## Configuration for shared credential.
## Default credential strategy will be used if it is absent here:
##            Instance role is used for EC2 case by default.
##            AmazonCloudWatchAgent profile is used for onPremise case by default.
# [credentials]
#    shared_credential_profile = "{profile_name}"
#    shared_credential_file= "{file_name}"

## Configuration for proxy.
## System-wide environment-variable will be read if it is absent here.
```

```
## i.e. HTTP_PROXY/http_proxy; HTTPS_PROXY/https_proxy; NO_PROXY/no_proxy
## Note: system-wide environment-variable is not accessible when using ssm run-command.
## Absent in both here and environment-variable means no proxy will be used.
# [proxy]
#    http_proxy = "{http_url}"
#    https_proxy = "{https_url}"
#    no_proxy = "{domain}"
```

Inicialmente, todas las líneas están convertidas en comentarios. Para establecer la configuración del perfil de credenciales o del proxy, quite # de esa línea y especifique un valor. Puede editar este archivo manualmente o mediante Run Command `RunShellScript` en Systems Manager:

- `shared_credential_profile`: para los servidores en las instalaciones, esta línea especifica el perfil de credenciales del usuario de IAM que se va a utilizar para enviar datos a CloudWatch. Si mantiene esta línea dentro de los comentarios, se utiliza `AmazonCloudWatchAgent`.

  En una instancia EC2, puede utilizar esta línea para que el agente de CloudWatch envíe datos desde esta instancia a CloudWatch en una Región de AWS distinta. Para ello, especifique un perfil con nombre que incluya un campo `region` especificando el nombre de la región a la que se van a enviar los datos.

  Si especifica un `shared_credential_profile`, también debe eliminar el # desde el principio de la línea `[credentials]`.
- `shared_credential_file`: para que el agente busque credenciales en un archivo ubicado en una ruta distinta de la ruta predeterminada, especifique esa ruta completa y el nombre de archivo aquí. La ruta predeterminada es `/root/.aws` en Linux y `C:\\Users\\Administrator\\.aws` en Windows Server.

  El primer ejemplo incluido a continuación muestra la sintaxis de una línea `shared_credential_file` válida para los servidores Linux y el segundo ejemplo es válido para Windows Server. En Windows Server, debe aplicar escape a los caracteres \.

  ```
  shared_credential_file= "/usr/username/credentials"
  ```

  ```
  shared_credential_file= "C:\\Documents and Settings\\username\\.aws\\credentials"
  ```

  Si especifica un `shared_credential_file`, también debe eliminar el # desde el principio de la línea `[credentials]`.

- Configuración del proxy: si los servidores usan proxies HTTP o HTTPS para contactarse con los servicios de AWS, especifíquelos en los campos `http_proxy` y `https_proxy`. Si hay URL que se deben excluir del proxy, especifíquelas en el campo `no_proxy`, separadas por comas.

Inicie el agente de CloudWatch

Se puede iniciar el agente con Run Command de Systems Manager o con la línea de comando.

Inicie el agente de CloudWatch con Run Command de Systems Manager

Siga estos pasos para iniciar el agente mediante Run Command de Systems Manager.

Para iniciar el agente de CloudWatch mediante Run Command

1. Abra la consola de Administrador de sistemas en https://console.aws.amazon.com/systems-manager/.
2. En el panel de navegación, elija Ejecutar comando.

   -o bien-

   Si la página de inicio de AWS Systems Manager se abre, desplácese hacia abajo y elija Explore Run Command (Explorar Run Command).
3. Elija Run command (Ejecutar comando).
4. En la lista Command document, elija AmazonCloudWatch-ManageAgent.
5. En el área Targets (Destinos), elija la instancia donde ha instalado el agente de CloudWatch.
6. En la lista Action, elija configure.
7. En la lista Optional Configuration Source, elija ssm.
8. En el cuadro Ubicación de configuración opcional, escriba el nombre del parámetro de Systems Manager nombre del archivo de configuración del agente que ha creado y guardado en el almacén de parámetros de Systems Manager, tal como se explica en Cree el archivo de configuración del agente de CloudWatch.
9. En la lista Optional Restart, elija yes para iniciar el agente después de haber finalizado estos pasos.
10. Elija Ejecutar.
11. Opcionalmente, en las áreas de Targets and outputs (Destinos y salidas), seleccione el botón situado junto a un nombre de instancia y elija View output (Ver salida). Systems Manager debería mostrar que el agente se ha iniciado correctamente.

Inicie el agente de CloudWatch en una instancia de Amazon EC2 mediante la línea de comandos

Siga estos pasos para utilizar la línea de comandos con el fin de instalar el agente de CloudWatch en una instancia de Amazon EC2.

Para utilizar la línea de comandos para iniciar el agente de CloudWatch en una instancia de Amazon EC2

- En este comando, `-a fetch-config` provoca que el agente cargue la última versión del archivo de configuración del agente de CloudWatch y `-s` inicia el agente.

  Linux y macOS: si ha guardado el archivo de configuración en el almacén de parámetros de Systems Manager, escriba lo siguiente:

  ```
  sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-
  config -m ec2 -s -c ssm:configuration-parameter-store-name
  ```

  Linux y macOS: si ha guardado el archivo de configuración en el equipo local, escriba el siguiente comando: Reemplace *`configuration-file-path`* por la ruta al archivo de configuración del agente. Este archivo se denomina `config.json` si lo creó con el asistente y podría llamarse `amazon-cloudwatch-agent.json` si lo creó manualmente.

  ```
  sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-
  config -m ec2 -s -c file:configuration-file-path
  ```

  Windows Server: si ha guardado el archivo de configuración del agente en el almacén de parámetros de Systems Manager, escriba lo siguiente en la consola de PowerShell:

  ```
  & "C:\Program Files\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent-ctl.ps1" -
  a fetch-config -m ec2 -s -c ssm:configuration-parameter-store-name
  ```

  Windows Server: si ha guardado el archivo de configuración del agente en el equipo local, escriba lo siguiente en la consola de PowerShell:

  ```
  & "C:\Program Files\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent-ctl.ps1"
   -a fetch-config -m ec2 -s -c file:"C:\Program Files\Amazon\AmazonCloudWatchAgent
  \config.json"
  ```

Instalación del agente de CloudWatch en servidores en las instalaciones

Si ha descargado el agente de CloudWatch en un equipo y ha creado el archivo de configuración del agente que desea utilizar, puede usar ese archivo de configuración para instalar el agente en servidores en las instalaciones.

Descargue del agente de CloudWatch en un servidor en las instalaciones

Puede descargar el paquete del agente de CloudWatch mediante el Run Command de Systems Manager o con un enlace de descarga de Amazon S3. Para obtener más información acerca del uso del enlace de descarga de Amazon S3, consulte Descargue del paquete de del agente de CloudWatch.

Descargue mediante Systems Manager

Para utilizar Run Command de Systems Manager, debe registrar el servidor en las instalaciones en Amazon EC2 Systems Manager. Para obtener más información, consulte Setting Up Systems Manager in Hybrid Environments (Configuración de Systems Manager en entornos híbridos) en la Guía del usuario de AWS Systems Manager.

Si ya ha registrado el servidor, actualice SSM Agent a la versión más reciente.

Para obtener más información acerca de la actualización de SSM Agent en un servidor con Linux, consulte Install SSM Agent for a Hybrid Environment (Linux) (Instalar SSM Agent para un entorno híbrido [Linux]) en la Guía del usuario de AWS Systems Manager.

Para obtener información acerca de la actualización de SSM Agent en un servidor con Windows Server, consulte Install SSM Agent for a Hybrid Environment (Windows) (Instalar SSM Agent para un entorno híbrido [Windows]) en la Guía del usuario de AWS Systems Manager.

Para utilizar SSM Agent para descargar el paquete de agente de CloudWatch en un servidor en las instalaciones

1. Abra la consola de Administrador de sistemas en https://console.aws.amazon.com/systems-manager/.
2. En el panel de navegación, elija Ejecutar comando.

   -o bien-

   Si la página de inicio de AWS Systems Manager se abre, desplácese hacia abajo y elija Explore Run Command (Explorar Run Command).

3. Elija Run command (Ejecutar comando).
4. En la lista Command document, seleccione el botón situado junto a AWS-ConfigureAWSPackage.
5. En el área Targets (Destinos), seleccione el servidor en el que va a instalar el agente de CloudWatch. Si no se ve un servidor específico, es posible que no esté configurado para Run Command. Para obtener más información, consulte Configuración de AWS Systems Manager para entornos híbridos en la Guía del usuario de AWS Systems Manager.
6. En la lista Action (Acción), elija Install (Instalar).
7. En el cuadro Name (Nombre), escriba *AmazonCloudWatchAgent*.
8. Deje Version (Versión) en blanco para instalar la última versión del agente.
9. Elija Ejecutar.

   El paquete de agente está descargado y los próximos pasos son configurarlo e iniciarlo.

(Instalación en un servidor en las instalaciones) Especifique las credenciales de IAM y la Región de AWS

Para que el agente de CloudWatch pueda enviar datos desde un servidor en las instalaciones, debe especificarse la clave de acceso y la clave secreta del usuario de IAM que creó anteriormente. Para obtener más información sobre la creación de este usuario, consulte Cree roles y usuarios de IAM para usarlos con el agente de CloudWatch.

También debe especificar la región de AWS a la que se envían las métricas mediante el campo `region`.

A continuación se muestra un ejemplo de este archivo.

```
[AmazonCloudWatchAgent]
aws_access_key_id=my_access_key
aws_secret_access_key=my_secret_key
region = us-west-1
```

Para *my_access_key* y *my_secret_key*, utilice las claves del usuario de IAM que no tienen los permisos para realizar escrituras en el almacén de parámetros de Systems Manager. Para obtener más información sobre los usuarios de IAM necesarios para el agente de CloudWatch, consulte Cree usuarios de IAM para utilizarlos con el agente de CloudWatch en servidores en las instalaciones.

Si asigna a este perfil el nombre `AmazonCloudWatchAgent`, no tiene que hacer nada más. Si lo prefiere, puede asignarle un nombre diferente y especificar ese nombre como valor de `shared_credential_profile` en el archivo `common-config.toml`, que se explica en la siguiente sección.

A continuación se muestra un ejemplo del uso del comando aws configure para crear un perfil con nombre para el agente de CloudWatch. En este ejemplo se presupone que usa el nombre de perfil predeterminado de `AmazonCloudWatchAgent`.

Para crear el perfil AmazonCloudWatchAgent para el agente de CloudWatch

1. Si aún no lo ha hecho, instale AWS Command Line Interface en el servidor. Para obtener más información, consulte Instalación de AWS CLI.
2. En los servidores Linux, escriba el siguiente comando y siga las instrucciones:

   ```
   sudo aws configure --profile AmazonCloudWatchAgent
   ```

   En Windows Server, abra PowerShell como administrador, escriba el siguiente comando y siga las instrucciones.

   ```
   aws configure --profile AmazonCloudWatchAgent
   ```

(Opcional) Modificación de la configuración común y el perfil con nombre para el agente de CloudWatch

El agente de CloudWatch incluye un archivo de configuración llamado `common-config.toml`. Si lo desea, puede utilizar este archivo para especificar la información de proxy y región.

En un servidor con Linux, este archivo se encuentra en el directorio `/opt/aws/amazon-cloudwatch-agent/etc`. En un servidor con Windows Server, este archivo se encuentra en el directorio `C:\ProgramData\Amazon\AmazonCloudWatchAgent`.

El `common-config.toml` predeterminado es el siguiente:

```
# This common-config is used to configure items used for both ssm and cloudwatch access


## Configuration for shared credential.
```

```
## Default credential strategy will be used if it is absent here:
##            Instance role is used for EC2 case by default.
##            AmazonCloudWatchAgent profile is used for onPremise case by default.
# [credentials]
#    shared_credential_profile = "{profile_name}"
#    shared_credential_file= "{file_name}"

## Configuration for proxy.
## System-wide environment-variable will be read if it is absent here.
## i.e. HTTP_PROXY/http_proxy; HTTPS_PROXY/https_proxy; NO_PROXY/no_proxy
## Note: system-wide environment-variable is not accessible when using ssm run-command.
## Absent in both here and environment-variable means no proxy will be used.
# [proxy]
#    http_proxy = "{http_url}"
#    https_proxy = "{https_url}"
#    no_proxy = "{domain}"
```

Inicialmente, todas las líneas están convertidas en comentarios. Para establecer la configuración del perfil de credenciales o del proxy, quite # de esa línea y especifique un valor. Puede editar este archivo manualmente o mediante Run Command `RunShellScript` en Systems Manager:

- `shared_credential_profile`: para los servidores en las instalaciones, esta línea especifica el perfil de credenciales del usuario de IAM que se va a utilizar para enviar datos a CloudWatch. Si mantiene esta línea dentro de los comentarios, se utiliza `AmazonCloudWatchAgent`. Para obtener más información sobre la creación de este perfil, consulte (Instalación en un servidor en las instalaciones) Especifique las credenciales de IAM y la Región de AWS.

  En una instancia EC2, puede utilizar esta línea para que el agente de CloudWatch envíe datos desde esta instancia a CloudWatch en una Región de AWS diferente. Para ello, especifique un perfil con nombre que incluya un campo `region` especificando el nombre de la región a la que se van a enviar los datos.

  Si especifica un `shared_credential_profile`, también debe eliminar el # desde el principio de la línea `[credentials]`.
- `shared_credential_file`: para que el agente busque credenciales en un archivo ubicado en una ruta distinta de la ruta predeterminada, especifique esa ruta completa y el nombre de archivo aquí. La ruta predeterminada es `/root/.aws` en Linux y `C:\\Users\\Administrator\\.aws` en Windows Server.

El primer ejemplo incluido a continuación muestra la sintaxis de una línea `shared_credential_file` válida para los servidores Linux y el segundo ejemplo es válido para Windows Server. En Windows Server, debe aplicar escape a los caracteres \.

```
shared_credential_file= "/usr/username/credentials"
```

```
shared_credential_file= "C:\\Documents and Settings\\username\\.aws\\credentials"
```

Si especifica un `shared_credential_file`, también debe eliminar el # desde el principio de la línea `[credentials]`.

- Configuración del proxy: si los servidores usan proxies HTTP o HTTPS para contactarse con los servicios de AWS, especifíquelos en los campos `http_proxy` y `https_proxy`. Si hay URL que se deben excluir del proxy, especifíquelas en el campo `no_proxy`, separadas por comas.

Inicio del agente de CloudWatch

Se puede iniciar el agente de CloudWatch mediante Run Command de Systems Manager o con la línea de comandos.

Para utilizar SSM Agent con el fin de iniciar el agente de CloudWatch en un servidor en las instalaciones

1. Abra la consola de Administrador de sistemas en https://console.aws.amazon.com/systems-manager/.
2. En el panel de navegación, elija Ejecutar comando.

   -o bien-

   Si la página de inicio de AWS Systems Manager se abre, desplácese hacia abajo y elija Explore Run Command (Explorar Run Command).
3. Elija Run command (Ejecutar comando).
4. En la lista Command document, seleccione el botón situado junto a AmazonCloudWatch-ManageAgent.
5. En el área Targets, seleccione la instancia donde ha instalado el agente.
6. En la lista Action, elija configure.
7. En la lista Mode, elija onPremise.

8. En el cuadro Ubicación de configuración opcional, ingrese el nombre del archivo de configuración del agente que ha creado con el asistente y que ha guardado en el almacén de parámetros.
9. Elija Ejecutar.

   El agente se inicia con la configuración que ha especificado en el archivo de configuración.

Para utilizar la línea de comandos con el fin de iniciar el agente de CloudWatch en un servidor en las instalaciones

- En este comando, `-a fetch-config` provoca que el agente cargue la última versión del archivo de configuración del agente de CloudWatch y `-s` inicia el agente.

  Linux: si ha guardado el archivo de configuración en el almacén de parámetros de Systems Manager, ingrese lo siguiente:

  ```
  sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-
  config -m onPremise -s -c ssm:configuration-parameter-store-name
  ```

  Linux: si ha guardado el archivo de configuración en el equipo local, escriba el siguiente comando: Reemplace *`configuration-file-path`* por la ruta al archivo de configuración del agente. Este archivo se denomina `config.json` si lo creó con el asistente y podría llamarse `amazon-cloudwatch-agent.json` si lo creó manualmente.

  ```
  sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-
  config -m onPremise -s -c file:configuration-file-path
  ```

  Windows Server: si ha guardado el archivo de configuración del agente en el almacén de parámetros de Systems Manager, ingrese lo siguiente en la consola de PowerShell:

  ```
  & "C:\Program Files\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent-ctl.ps1" -
  a fetch-config -m onPremise -s -c ssm:configuration-parameter-store-name
  ```

  Windows Server: si ha guardado el archivo de configuración del agente en el equipo local, escriba lo siguiente en la consola de PowerShell. Reemplace *`configuration-file-path`* por la ruta al archivo de configuración del agente. Este archivo se denomina `config.json` si lo ha

creado con el asistente y podría llamarse `amazon-cloudwatch-agent.json` si lo ha creado manualmente.

```
& "C:\Program Files\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent-ctl.ps1" -
a fetch-config -m onPremise -s -c file:configuration-file-path
```

Instalación del agente de CloudWatch en instancias nuevas mediante AWS CloudFormation

Amazon ha cargado varias plantillas de AWS CloudFormation en GitHub para que pueda instalar y actualizar el agente de CloudWatch en las instancias nuevas de Amazon EC2. Para obtener más información acerca del uso de AWS CloudFormation, consulte ¿Qué es AWS CloudFormation?

La plantilla se encuentra en Deploy the Amazon CloudWatch agent to EC2 instances using AWS CloudFormation (Implementación del agente de Amazon CloudWatch en las instancias de EC2 mediante). Esta ubicación incluye los directorios `inline` y `ssm`. Cada uno de estos directorios contiene plantillas para ambas instancias de Linux y Windows.

- Las plantillas del directorio `inline` (en línea) tienen la configuración del agente de CloudWatch incorporada en la plantilla de AWS CloudFormation. De forma predeterminada, las plantillas de Linux recopilan las métricas `mem_used_percent` y `swap_used_percent`, mientras que las plantillas de Windows recopilan `Memory % Committed Bytes In Use` y `Paging File % Usage`.

  Para modificar estas plantillas de forma que recopilen métricas diferentes, modifique la siguiente sección de la plantilla. En el siguiente ejemplo es de la plantilla para servidores Linux. Siga el formato y la sintaxis del archivo de configuración del agente para realizar estos cambios. Para obtener más información, consulte Cree o edite de forma manual el archivo de configuración del agente de CloudWatch.

```
{
   "metrics":{
      "append_dimensions":{
         "AutoScalingGroupName":"${!aws:AutoScalingGroupName}",
         "ImageId":"${!aws:ImageId}",
         "InstanceId":"${!aws:InstanceId}",
         "InstanceType":"${!aws:InstanceType}"
      },
```

```
      "metrics_collected":{
         "mem":{
            "measurement":[
               "mem_used_percent"
            ]
         },
         "swap":{
            "measurement":[
               "swap_used_percent"
            ]
         }
      }
   }
}
```

> **Note**
>
> En las plantillas en línea, todas las variables de marcador de posición deben tener un signo de exclamación (!) delante como carácter de escape. Puede ver esto en el ejemplo de plantilla. Si añade otras variables de marcador de posición, asegúrese de añadir un signo de exclamación delante del nombre.

- Las plantillas del directorio `ssm` cargan un archivo de configuración del agente desde el almacén de parámetros. Para utilizar estas plantillas, primero se debe crear un archivo de configuración y cargarlo en el almacén de parámetros. A continuación, deberá proporcionar el nombre del almacén de parámetros del archivo en la plantilla. El archivo de configuración lo puede crear manualmente o con el asistente. Para obtener más información, consulte Cree el archivo de configuración del agente de CloudWatch.

Puede utilizar ambos tipos de plantillas para instalar el agente de CloudWatch y para actualizar la configuración del agente.

Tutorial: Instalación y configuración del agente de CloudWatch mediante una plantilla en línea de AWS CloudFormation

En este tutorial aprenderá a utilizar AWS CloudFormation para instalar el agente de CloudWatch en una nueva instancia de Amazon EC2. En este tutorial se instala el agente en una nueva instancia que ejecuta Amazon Linux 2 con las plantillas integradas que no requieren el uso del archivo de configuración JSON ni el almacén de parámetros. La plantilla en línea incluye la configuración del

agente en la plantilla. En este tutorial, utilice la configuración del agente predeterminado incluido en la plantilla.

Tras el procedimiento para instalar el agente, el tutorial continúa con cómo actualizar el agente.

Para utilizar AWS CloudFormation para instalar el agente de CloudWatch en una instancia nueva

1. Descargue la plantilla de GitHub. En este tutorial, descargue la plantilla integrada para Amazon Linux 2 del siguiente modo:

   ```
   curl -O https://raw.githubusercontent.com/awslabs/aws-cloudformation-templates/
   master/aws/solutions/AmazonCloudWatchAgent/inline/amazon_linux.template
   ```

2. Abra la consola de AWS CloudFormation en https://console.aws.amazon.com/cloudformation.
3. Elija Crear pila.
4. Para Choose a template (Elegir una plantilla), seleccione Upload a template to Amazon S3 (Cargar una plantilla en Amazon S3), elija la plantilla descargada y elija Next (Siguiente).
5. En la página Specify Details (Especificar detalles), rellene los parámetros siguientes y elija Next (Siguiente):

   - Nombre de pila: elija un nombre para la pila de AWS CloudFormation.
   - IAMRole: elija un rol de IAM que tenga permisos para escribir métricas, registros y seguimientos de CloudWatch. Para obtener más información, consulte Cree roles de IAM para utilizarlos con el agente de CloudWatch en instancias de Amazon EC2.
   - InstanceAMI: elija una AMI que sea válida en la región en la que va a lanzar la pila.
   - InstanceType: elija un tipo de instancia válido.
   - KeyName: para permitir el acceso de SSH a la nueva instancia, elija un par de claves existente de Amazon EC2. Si aún no tiene un par de claves de Amazon EC2, puede crear uno en la AWS Management Console. Para obtener más información, consulte Amazon EC2 Key Pairs (Pares de claves de Amazon EC2) en la Guía del usuario de Amazon EC2 para instancias de Linux.
   - SSHLocation: especifica el rango de direcciones IP que se puede utilizar para conectarse a la instancia por medio de SSH. El valor predeterminado permite el acceso desde cualquier dirección IP.
6. En la página Options (Opciones), puede elegir marcar los recursos de la pila. Elija Siguiente.
7. En la página Review (Revisar), revise la información, confirme la advertencia de que la pila puede crear recursos de IAM y elija Create (Crear).

Si actualiza la consola, verá que la nueva pila tiene el estado `CREATE_IN_PROGRESS`.

8. Cuando se crea la instancia, puede verla en la consola de Amazon EC2. Si lo desea, puede conectarse al host y comprobar el progreso.

   Utilice el siguiente comando para confirmar que el agente se ha instalado:

   ```
   rpm -qa amazon-cloudwatch-agent
   ```

   Utilice el siguiente comando para confirmar que el agente se está ejecutando:

   ```
   ps aux | grep amazon-cloudwatch-agent
   ```

En el siguiente procedimiento se muestra el uso de AWS CloudFormation para actualizar el agente de CloudWatch con una plantilla en línea. La plantilla en línea predeterminada recopila la métrica `mem_used_percent`. En este tutorial, va a cambiar la configuración del agente para dejar de recopilar esa métrica.

Para utilizar AWS CloudFormation con el fin de actualizar el agente de CloudWatch

1. En la plantilla que descargó en el procedimiento anterior, elimine las siguientes líneas y, a continuación, guarde la plantilla:

   ```
   "mem": {

        "measurement": [
            "mem_used_percent"
          ]
    },
   ```

2. Abra la consola de AWS CloudFormation en https://console.aws.amazon.com/cloudformation.
3. En el panel de AWS CloudFormation, seleccione la pila que creó y elija Update Stack (Actualizar pila).
4. Para Select Template (Seleccionar plantilla), seleccione Upload a template to Amazon S3 (Cargar una plantilla en Amazon S3), elija la plantilla modificada y elija Next (Siguiente).
5. En la página Options (Opciones), elija Next (Siguiente) y después Next (Siguiente).
6. En la página Review (Revisar), revise la información y seleccione Update (Actualizar).

Después de algún tiempo, verá `UPDATE_COMPLETE`.

Tutorial: Instalación del agente de CloudWatch mediante AWS CloudFormation y Parameter Store

En este tutorial aprenderá a utilizar AWS CloudFormation para instalar el agente de CloudWatch en una nueva instancia de Amazon EC2. En el tutorial se instala en una nueva instancia que ejecute Amazon Linux 2 con un archivo de configuración del agente que se cree y guarde en el almacén de parámetros.

Tras el procedimiento para instalar el agente, el tutorial continúa con cómo actualizar el agente.

Para utilizar AWS CloudFormation con el fin de instalar el agente de CloudWatch en una instancia nueva mediante una configuración desde Parameter Store

1. Si aún no lo ha hecho, debe descargue el paquete de agente de CloudWatch en uno de los equipos, de modo que pueda crear el archivo de configuración del agente. Para obtener más información sobre cómo descargar el agente con el almacén de parámetros, consulte Descargue y configure el agente de CloudWatch. Para obtener más información acerca de cómo descargar el paquete con la línea de comandos, consulte Descargue y configure el agente de CloudWatch con la línea de comandos.
2. Cree el archivo de configuración del agente y guárdelo en el almacén de parámetros. Para obtener más información, consulte Cree el archivo de configuración del agente de CloudWatch.
3. Descargue la plantilla de GitHub del modo siguiente.

   ```
   curl -O https://raw.githubusercontent.com/awslabs/aws-cloudformation-templates/
   master/aws/solutions/AmazonCloudWatchAgent/ssm/amazon_linux.template
   ```

4. Abra la consola de AWS CloudFormation en https://console.aws.amazon.com/cloudformation.
5. Elija Crear pila.
6. Para Choose a template (Elegir una plantilla), seleccione Upload a template to Amazon S3 (Cargar una plantilla en Amazon S3), elija la plantilla descargada y elija Next (Siguiente).
7. En la página Specify Details (Especificar detalles), rellene los parámetros siguientes según corresponda y elija Next (Siguiente):

   - Nombre de pila: elija un nombre para la pila de AWS CloudFormation.

- IAMRole: elija un rol de IAM que tenga permisos para escribir métricas, registros y seguimientos de CloudWatch. Para obtener más información, consulte Cree roles de IAM para utilizarlos con el agente de CloudWatch en instancias de Amazon EC2.
- InstanceAMI: elija una AMI que sea válida en la región en la que va a lanzar la pila.
- InstanceType: elija un tipo de instancia válido.
- KeyName: para permitir el acceso de SSH a la nueva instancia, elija un par de claves existente de Amazon EC2. Si aún no tiene un par de claves de Amazon EC2, puede crear uno en la AWS Management Console. Para obtener más información, consulte Amazon EC2 Key Pairs (Pares de claves de Amazon EC2) en la Guía del usuario de Amazon EC2 para instancias de Linux.
- SSHLocation: especifica el rango de direcciones IP que se puede utilizar para conectarse a la instancia por medio de SSH. El valor predeterminado permite el acceso desde cualquier dirección IP.
- SSMKey: especifica el archivo de configuración del agente que ha creado y guardado en el almacén de parámetros.

8. En la página Options (Opciones), puede elegir marcar los recursos de la pila. Elija Siguiente.
9. En la página Review (Revisar), revise la información, confirme la advertencia de que la pila puede crear recursos de IAM y elija Create (Crear).

   Si actualiza la consola, verá que la nueva pila tiene el estado `CREATE_IN_PROGRESS`.
10. Cuando se crea la instancia, puede verla en la consola de Amazon EC2. Si lo desea, puede conectarse al host y comprobar el progreso.

    Utilice el siguiente comando para confirmar que el agente se ha instalado:

    ```
    rpm -qa amazon-cloudwatch-agent
    ```

    Utilice el siguiente comando para confirmar que el agente se está ejecutando:

    ```
    ps aux | grep amazon-cloudwatch-agent
    ```

En el siguiente procedimiento se observa el uso de AWS CloudFormation para actualizar el agente de CloudWatch mediante una configuración del agente que se guardó en Parameter Store.

Para utilizar AWS CloudFormation con el fin de actualizar el agente de CloudWatch mediante una configuración en Parameter Store

1. Cambie el archivo de configuración del agente que se encuentra almacenado en el almacén de parámetros a la nueva configuración que desee.
2. En la plantilla de AWS CloudFormation que ha descargado en el tema the section called "Tutorial: Instalación del agente de CloudWatch mediante AWS CloudFormation y Parameter Store", cambie el número de versión. Por ejemplo, podría cambiar `VERSION=1.0` por `VERSION=2.0`.
3. Abra la consola de AWS CloudFormation en https://console.aws.amazon.com/cloudformation.
4. En el panel de AWS CloudFormation, seleccione la pila que creó y elija Update Stack (Actualizar pila).
5. Para Select Template (Seleccionar plantilla), seleccione Upload a template to Amazon S3 (Cargar una plantilla en Amazon S3), seleccione la plantilla que acaba de modificar y elija Next (Siguiente).
6. En la página Options (Opciones), elija Next (Siguiente) y después Next (Siguiente).
7. En la página Review (Revisar), revise la información y seleccione Update (Actualizar).

   Después de algún tiempo, verá `UPDATE_COMPLETE`.

Solución de problemas de la instalación del agente de CloudWatch con AWS CloudFormation

Esta sección lo ayudará a solucionar los problemas de instalación y actualización del agente de CloudWatch mediante AWS CloudFormation.

Detección de un error de actualización

Si utiliza AWS CloudFormation para actualizar la configuración del agente de CloudWatch y utiliza una configuración inválida, el agente deja de enviar las métricas a CloudWatch. Una forma rápida de comprobar si la configuración del agente se actualiza correctamente es observar el archivo `cfn-init-cmd.log`. En un servidor Linux, el archivo se encuentra en `/var/log/cfn-init-cmd.log`. En una instancia de Windows, el archivo se encuentra en `C:\cfn\log\cfn-init-cmd.log`.

Ausencia de métricas

Si no ve las métricas que espera ver después de instalar o actualizar el agente, confirme que el agente esté configurado para recopilar esa métrica. Para ello, compruebe el archivo `amazon-`

`cloudwatch-agent.json` para asegurarse de que la métrica figure en la lista y compruebe que está mirando en el espacio de nombres de métricas correcto. Para obtener más información, consulte Archivos y ubicaciones del agente de CloudWatch.

Preferencia de credenciales del agente de CloudWatch

En esta sección se describe la cadena de proveedores de credenciales que el agente de CloudWatch utiliza para obtener las credenciales al comunicarse con otros servicios y API de AWS. El orden es el siguiente. Las preferencias que aparecen en los números dos a cinco de la siguiente lista siguen el mismo orden de preferencia definido en el SDK de AWS. Para obtener más información, consulte Specifying Credentials en la documentación del SDK.

1. Archivos de configuración y credenciales compartidos tal y como se definen en el archivo `common-config.toml` del agente de CloudWatch. Para obtener más información, consulte (Opcional) Modifique la configuración común para la información del proxy o de la región.
2. Variables de entorno del SDK de AWS

> **Important**
>
> En Linux, si ejecuta el agente de CloudWatch mediante el script `amazon-cloudwatch-agent-ctl`, el script inicia el agente como un servicio `systemd`. En este caso, el agente no puede acceder a variables de entorno como `HOME`, `AWS_ACCESS_KEY_ID` y `AWS_SECRET_ACCESS_KEY`.

3. Archivos de configuración y credenciales compartidos que se encuentran en `$HOME/%USERPROFILE%`

> **Note**
>
> El agente de CloudWatch busca `.aws/credentials` en `$HOME` para Linux y macOS y busca en `%USERPROFILE%` para Windows. A diferencia del SDK de AWS, el agente de CloudWatch no tiene métodos alternativos para determinar el directorio principal si no se puede acceder a las variables de entorno. Esta diferencia de comportamiento sirve para mantener la compatibilidad con versiones anteriores del SDK de AWS.
> Además, a diferencia de las credenciales compartidas que se encuentran en `common-config.toml`, si las credenciales compartidas derivadas del SDK de AWS caducan y se

rotan, el agente de CloudWatch no recogerá automáticamente las credenciales renovadas y será necesario reiniciar el agente para hacerlo.

4. Un rol de AWS Identity and Access Management para tareas si hay una aplicación presente que utiliza una definición de tarea de Amazon Elastic Container Service o una operación de API RunTask.
5. un perfil de instancias adjunto a una instancia de Amazon EC2

Como práctica recomendada, le recomendamos que especifique las credenciales en el orden siguiente cuando utilice el agente de CloudWatch.

1. Utilice roles de IAM para las tareas si su aplicación utiliza una definición de tarea de Amazon Elastic Container Service o una operación de la API RunTask.
2. Utilice roles de IAM si su aplicación se ejecuta en una instancia de Amazon EC2.
3. Utilice el archivo `common-config.toml` de agente de CloudWatch para especificar el archivo de credenciales. Este archivo de credenciales es el mismo que utilizan los demás SDK de AWS y la AWS CLI. Si ya está utilizando un archivo de credenciales compartido, también puede utilizarlo para este fin. Si lo proporciona mediante el archivo `common-config.toml` de agente de CloudWatch, se asegura de que el agente consumirá las credenciales rotadas cuando caduquen y se sustituirán sin necesidad de reiniciar el agente.
4. Utilice variables de entorno. Configurar variables de entorno es útil si está realizando trabajos de desarrollo en una computadora que no sea una instancia de Amazon EC2.

> **Note**
>
> Si envía la telemetría a una cuenta diferente, como se explica en Envío de métricas, registros y seguimientos a una cuenta diferente, el agente de CloudWatch utilizará la cadena de proveedores de credenciales que se describe en esta sección para obtener el conjunto inicial de credenciales. A continuación, utiliza esas credenciales al asumir el rol de IAM especificado por `role_arn` en el archivo de configuración del agente de CloudWatch.

Verificación de la firma del paquete del agente de CloudWatch

Se incluyen archivos de firma GPG para los paquetes de agente de CloudWatch en servidores Linux. Puede utilizar la clave pública para verificar que el archivo de descarga del agente es original y no se ha modificado.

Para Windows Server, puede utilizar MSI para verificar la firma.

En el caso de los equipos macOS, la firma se incluye en el paquete de descarga del agente.

Para buscar el archivo de firma correcto, consulte la siguiente tabla. Para cada arquitectura y sistema operativo hay un enlace general y enlaces para cada región. Por ejemplo, para Amazon Linux 2023, Amazon Linux 2 y la arquitectura x86-64, tres de los enlaces válidos son:

- `https://amazoncloudwatch-agent.s3.amazonaws.com/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm.sig`
- `https://amazoncloudwatch-agent-us-east-1.s3.us-east-1.amazonaws.com/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm`
- `https://amazoncloudwatch-agent-eu-central-1.s3.eu-central-1.amazonaws.com/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm`

> **Note**
>
> Para descargar el agente de CloudWatch, la conexión deben usar la TLS 1.2 o una versión posterior.

Arquitectura	Plataforma	Enlace de descarga	Enlace de archivo de firma
x86-64	Amazon Linux 2023 y Amazon Linux 2	https://amazoncloudwatch-agent.s3.amazonaws.com/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm	https://amazoncloudwatch-agent.s3.amazonaws.com/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm.sig
		https://amazoncloudwatch-agent-*region*.s3.*region*.amazon	https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaw

Arquitectura	Plataforma	Enlace de descarga	Enlace de archivo de firma
		s.com/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm	s.com/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm.sig
x86-64	Centos	https://amazoncloudwatch-agent.s3.amazonaws.com/centos/amd64/latest/amazon-cloudwatch-agent.rpm https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/centos/amd64/latest/amazon-cloudwatch-agent.rpm	https://amazoncloudwatch-agent.s3.amazonaws.com/centos/amd64/latest/amazon-cloudwatch-agent.rpm.sig https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/centos/amd64/latest/amazon-cloudwatch-agent.rpm.sig
x86-64	Redhat	https://amazoncloudwatch-agent.s3.amazonaws.com/redhat/amd64/latest/amazon-cloudwatch-agent.rpm https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/redhat/amd64/latest/amazon-cloudwatch-agent.rpm	https://amazoncloudwatch-agent.s3.amazonaws.com/redhat/amd64/latest/amazon-cloudwatch-agent.rpm.sig https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/redhat/amd64/latest/amazon-cloudwatch-agent.rpm.sig
x86-64	SUSE	https://amazoncloudwatch-agent.s3.amazonaws.com/suse/amd64/latest/amazon-cloudwatch-agent.rpm https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/suse/amd64/latest/amazon-cloudwatch-agent.rpm	https://amazoncloudwatch-agent.s3.amazonaws.com/suse/amd64/latest/amazon-cloudwatch-agent.rpm.sig https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/suse/amd64/latest/amazon-cloudwatch-agent.rpm.sig

Arquitectura	Plataforma	Enlace de descarga	Enlace de archivo de firma
x86-64	Debian	https://amazoncloudwatch-agent.s3.amazonaws.com/debian/amd64/latest/amazon-cloudwatch-agent.deb https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/debian/amd64/latest/amazon-cloudwatch-agent.deb	https://amazoncloudwatch-agent.s3.amazonaws.com/debian/amd64/latest/amazon-cloudwatch-agent.deb.sig https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/debian/amd64/latest/amazon-cloudwatch-agent.deb.sig
x86-64	Ubuntu	https://amazoncloudwatch-agent.s3.amazonaws.com/ubuntu/amd64/latest/amazon-cloudwatch-agent.deb https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/ubuntu/amd64/latest/amazon-cloudwatch-agent.deb	https://amazoncloudwatch-agent.s3.amazonaws.com/ubuntu/amd64/latest/amazon-cloudwatch-agent.deb.sig https://s3.*region*.amazonaws.com/amazoncloudwatch-agent-*region*/ubuntu/amd64/latest/amazon-cloudwatch-agent.deb.sig
x86-64	Oracle	https://amazoncloudwatch-agent.s3.amazonaws.com/oracle_linux/amd64/latest/amazon-cloudwatch-agent.rpm https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/oracle_linux/amd64/latest/amazon-cloudwatch-agent.rpm	https://amazoncloudwatch-agent.s3.amazonaws.com/oracle_linux/amd64/latest/amazon-cloudwatch-agent.rpm.sig https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/oracle_linux/amd64/latest/amazon-cloudwatch-agent.rpm.sig

Arquitectura	Plataforma	Enlace de descarga	Enlace de archivo de firma
x86-64	macOS	https://amazoncloudwatch-agent.s3.amazonaws.com/darwin/amd64/latest/amazon-cloudwatch-agent.pkg https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/darwin/amd64/latest/amazon-cloudwatch-agent.pkg	https://amazoncloudwatch-agent.s3.amazonaws.com/darwin/amd64/latest/amazon-cloudwatch-agent.pkg.sig https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/darwin/amd64/latest/amazon-cloudwatch-agent.pkg.sig
x86-64	Windows	https://amazoncloudwatch-agent.s3.amazonaws.com/windows/amd64/latest/amazon-cloudwatch-agent.msi https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/windows/amd64/latest/amazon-cloudwatch-agent.msi	https://amazoncloudwatch-agent.s3.amazonaws.com/windows/amd64/latest/amazon-cloudwatch-agent.msi.sig https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/windows/amd64/latest/amazon-cloudwatch-agent.msi.sig
ARM64	Amazon Linux 2023 y Amazon Linux 2	https://amazoncloudwatch-agent.s3.amazonaws.com/amazon_linux/arm64/latest/amazon-cloudwatch-agent.rpm https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/amazon_linux/arm64/latest/amazon-cloudwatch-agent.rpm	https://amazoncloudwatch-agent.s3.amazonaws.com/amazon_linux/arm64/latest/amazon-cloudwatch-agent.rpm.sig https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/amazon_linux/arm64/latest/amazon-cloudwatch-agent.rpm.sig

Arquitectura	Plataforma	Enlace de descarga	Enlace de archivo de firma
ARM64	Redhat	https://amazoncloudwatch-agent.s3.amazonaws.com/redhat/arm64/latest/amazon-cloudwatch-agent.rpm https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/redhat/arm64/latest/amazon-cloudwatch-agent.rpm	https://amazoncloudwatch-agent.s3.amazonaws.com/redhat/arm64/latest/amazon-cloudwatch-agent.rpm.sig https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/redhat/arm64/latest/amazon-cloudwatch-agent.rpm.sig
ARM64	Ubuntu	https://amazoncloudwatch-agent.s3.amazonaws.com/ubuntu/arm64/latest/amazon-cloudwatch-agent.deb https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/ubuntu/arm64/latest/amazon-cloudwatch-agent.deb	https://amazoncloudwatch-agent.s3.amazonaws.com/ubuntu/arm64/latest/amazon-cloudwatch-agent.deb.sig https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/ubuntu/arm64/latest/amazon-cloudwatch-agent.deb.sig
ARM64	SUSE	https://amazoncloudwatch-agent.s3.amazonaws.com/suse/arm64/latest/amazon-cloudwatch-agent.rpm https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/suse/arm64/latest/amazon-cloudwatch-agent.rpm	https://amazoncloudwatch-agent.s3.amazonaws.com/suse/arm64/latest/amazon-cloudwatch-agent.rpm.sig https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/suse/arm64/latest/amazon-cloudwatch-agent.rpm.sig

Para verificar el paquete de agente de CloudWatch en un servidor Linux

1. Descargue la clave pública.

```
shell$ wget https://amazoncloudwatch-agent.s3.amazonaws.com/assets/amazon-
cloudwatch-agent.gpg
```

2. Importe la clave pública en su llavero.

```
shell$  gpg --import amazon-cloudwatch-agent.gpg
gpg: key 3B789C72: public key "Amazon CloudWatch Agent" imported
gpg: Total number processed: 1
gpg: imported: 1 (RSA: 1)
```

   Anote el valor de clave, ya que lo necesitará en el siguiente paso. En el ejemplo anterior, el valor de la clave es 3B789C72.

3. Verifique la huella digital ejecutando el siguiente comando, sustituyendo el *valor de la clave* por el valor del paso anterior:

```
shell$  gpg --fingerprint key-value
pub   2048R/3B789C72 2017-11-14
      Key fingerprint = 9376 16F3 450B 7D80 6CBD  9725 D581 6730 3B78 9C72
uid                  Amazon CloudWatch Agent
```

   La cadena de huella debería ser igual a la siguiente:

   ```
   9376 16F3 450B 7D80 6CBD 9725 D581 6730 3B78 9C72
   ```

   Si la cadena de huellas digitales no coincide, no instale el agente. Contáctese con Amazon Web Services.

   Después de haber verificado la huella, puede utilizarla para verificar la firma del paquete de agente de CloudWatch.

4. Descargue el archivo de firma del paquete mediante wget. Para determinar el archivo de firma correcto, consulte la tabla anterior:

```
wget Signature File Link
```

5. Para verificar la firma, ejecute gpg --verify.

```
shell$ gpg --verify signature-filename agent-download-filename
gpg: Signature made Wed 29 Nov 2017 03:00:59 PM PST using RSA key ID 3B789C72
gpg: Good signature from "Amazon CloudWatch Agent"
```

```
gpg: WARNING: This key is not certified with a trusted signature!
gpg:          There is no indication that the signature belongs to the owner.
Primary key fingerprint: 9376 16F3 450B 7D80 6CBD  9725 D581 6730 3B78 9C72
```

Si el resultado incluye la expresión `BAD signature`, compruebe si ha realizado el procedimiento correctamente. Si sigue recibiendo esta respuesta, contáctese con Amazon Web Services y evite usar el archivo descargado.

Tenga en cuenta la advertencia sobre confianza. Una clave solo es de confianza si la ha firmado usted o alguien en quien confíe. Esto no significa que la firma no sea válida, solo que no ha verificado la clave pública.

Para verificar el paquete de agente de CloudWatch en un servidor con Windows Server

1. Descargue e instale GnuPG para Windows desde https://gnupg.org/download/. Al realizar la instalación, incluya la opción Shell Extension (GpgEx).

   Puede realizar los pasos restantes en Windows PowerShell.

2. Descargue la clave pública.

   ```
   PS> wget https://amazoncloudwatch-agent.s3.amazonaws.com/assets/amazon-cloudwatch-agent.gpg -OutFile amazon-cloudwatch-agent.gpg
   ```

3. Importe la clave pública en su llavero.

   ```
   PS>  gpg --import amazon-cloudwatch-agent.gpg
   gpg: key 3B789C72: public key "Amazon CloudWatch Agent" imported
   gpg: Total number processed: 1
   gpg: imported: 1 (RSA: 1)
   ```

   Anote el valor de clave, ya que lo necesitará en el siguiente paso. En el ejemplo anterior, el valor de la clave es `3B789C72`.

4. Verifique la huella digital ejecutando el siguiente comando, sustituyendo el *`valor de la clave`* por el valor del paso anterior:

   ```
   PS>  gpg --fingerprint key-value
   pub   rsa2048 2017-11-14 [SC]
         9376 16F3 450B 7D80 6CBD  9725 D581 6730 3B78 9C72
   ```

```
uid           [ unknown] Amazon CloudWatch Agent
```

La cadena de huella debería ser igual a la siguiente:

```
9376 16F3 450B 7D80 6CBD 9725 D581 6730 3B78 9C72
```

Si la cadena de huellas digitales no coincide, no instale el agente. Contáctese con Amazon Web Services.

Después de haber verificado la huella, puede utilizarla para verificar la firma del paquete de agente de CloudWatch.

5. Descargue el archivo de firma del paquete mediante wget. Para determinar cuál es el archivo de firma correcto, consulte Enlaces de descarga del agente de CloudWatch.
6. Para verificar la firma, ejecute gpg --verify.

```
PS> gpg --verify sig-filename agent-download-filename
gpg: Signature made 11/29/17 23:00:45 Coordinated Universal Time
gpg:                using RSA key D58167303B789C72
gpg: Good signature from "Amazon CloudWatch Agent" [unknown]
gpg: WARNING: This key is not certified with a trusted signature!
gpg:          There is no indication that the signature belongs to the owner.
Primary key fingerprint: 9376 16F3 450B 7D80 6CBD  9725 D581 6730 3B78 9C72
```

Si el resultado incluye la expresión `BAD signature`, compruebe si ha realizado el procedimiento correctamente. Si sigue recibiendo esta respuesta, contáctese con Amazon Web Services y evite usar el archivo descargado.

Tenga en cuenta la advertencia sobre confianza. Una clave solo es de confianza si la ha firmado usted o alguien en quien confíe. Esto no significa que la firma no sea válida, solo que no ha verificado la clave pública.

Para verificar el paquete del agente de CloudWatch en un equipo con macOS

- Existen dos métodos para la verificación de firma en macOS.
  - Verifique la huella mediante la ejecución del siguiente comando:

```
pkgutil --check-signature amazon-cloudwatch-agent.pkg
```

Se debería ver un resultado similar al siguiente:

```
Package "amazon-cloudwatch-agent.pkg":
       Status: signed by a developer certificate issued by Apple for
 distribution
       Signed with a trusted timestamp on: 2020-10-02 18:13:24 +0000
       Certificate Chain:
       1. Developer ID Installer: AMZN Mobile LLC (94KV3E626L)
       Expires: 2024-10-18 22:31:30 +0000
       SHA256 Fingerprint:
       81 B4 6F AF 1C CA E1 E8 3C 6F FB 9E 52 5E 84 02 6E 7F 17 21 8E FB
       0C 40 79 13 66 8D 9F 1F 10 1C

 ------------------------------------------------------------------------
       2. Developer ID Certification Authority
       Expires: 2027-02-01 22:12:15 +0000
       SHA256 Fingerprint:
       7A FC 9D 01 A6 2F 03 A2 DE 96 37 93 6D 4A FE 68 09 0D 2D E1 8D 03
       F2 9C 88 CF B0 B1 BA 63 58 7F

 ------------------------------------------------------------------------
       3. Apple Root CA
       Expires: 2035-02-09 21:40:36 +0000
       SHA256 Fingerprint:
       B0 B1 73 0E CB C7 FF 45 05 14 2C 49 F1 29 5E 6E DA 6B CA ED 7E 2C
       68 C5 BE 91 B5 A1 10 01 F0 24
```

- O bien, descargue y use el archivo .sig; para utilizar este método, siga estos pasos.
  - Instale la aplicación GPG en el host de macOS al ingresar el siguiente comando.

    ```
    brew install GnuPG
    ```

- Descargue el archivo SIGNATURE del paquete mediante curl. Para determinar cuál es el archivo de firma correcto, consulte Enlaces de descarga del agente de CloudWatch.
- Para verificar la firma, ejecute gpg --verify.

```
PS> gpg --verify sig-filename agent-download-filename
gpg: Signature made 11/29/17 23:00:45 Coordinated Universal Time
gpg:                using RSA key D58167303B789C72
gpg: Good signature from "Amazon CloudWatch Agent" [unknown]
gpg: WARNING: This key is not certified with a trusted signature!
```

```
gpg:          There is no indication that the signature belongs to the owner.
Primary key fingerprint: 9376 16F3 450B 7D80 6CBD  9725 D581 6730 3B78 9C72
```

Si el resultado incluye la expresión `BAD signature`, compruebe si ha realizado el procedimiento correctamente. Si sigue recibiendo esta respuesta, contáctese con Amazon Web Services y evite usar el archivo descargado.

Tenga en cuenta la advertencia sobre confianza. Una clave solo es de confianza si la ha firmado usted o alguien en quien confíe. Esto no significa que la firma no sea válida, solo que no ha verificado la clave pública.

Cree el archivo de configuración del agente de CloudWatch

Antes de ejecutar el agente de CloudWatch en un servidor, debe crear uno o más archivos de configuración del agente de CloudWatch.

El archivo de configuración del agente es un archivo JSON que especifica las métricas, registros y seguimientos que debe recopilar el agente, incluidas las métricas personalizadas. Puede crearlo mediante el asistente o desde cero. También puede utilizar el asistente para crear inicialmente el archivo de configuración y, a continuación, modificarlo manualmente. Si lo crea o lo modifica manualmente, el proceso es más complejo, pero dispone de más control sobre las métricas recopiladas y puede especificar métricas no disponibles en el asistente.

Cada vez que cambie el archivo de configuración del agente, debe reiniciar el agente para que los cambios surtan efecto. Para reiniciar el agente, siga las instrucciones que se describen en Inicie el agente de CloudWatch.

Una vez que haya creado un archivo de configuración, puede guardarlo manualmente como un archivo JSON y, a continuación, utilizar este archivo al instalar el agente en sus servidores. Si lo prefiere, se puede almacenar en el almacén de parámetros de Systems Manager si va a utilizar Systems Manager al instalar el agente en los servidores.

El agente de CloudWatch admite el uso de varios archivos de configuración. Para obtener más información, consulte Varios archivos de configuración del agente de CloudWatch.

Las métricas, los registros y los seguimientos recopilados por el agente de CloudWatch conllevan cargos. Para obtener más información sobre precios, consulte Precios de Amazon CloudWatch.

Contenido

Cree el archivo de configuración del agente de CloudWatch con el asistente

El asistente de archivos de configuración del agente, `amazon-cloudwatch-agent-config-wizard`, hace una serie de preguntas para ayudarle a configurar el agente de CloudWatch según sus necesidades.

Credenciales obligatorias

El asistente puede detectar automáticamente las credenciales y la región de AWS que debe usar si dispone de los archivos de credenciales y de configuración de AWS antes de iniciar el asistente. Para obtener más información acerca de estos archivos, consulte Configuration and Credential Files (Archivos de configuración y credenciales) en la Guía del usuario de AWS Systems Manager.

En el archivo de credenciales de AWS, el asistente comprueba las credenciales predeterminadas y también busca una sección `AmazonCloudWatchAgent` como la siguiente:

```
[AmazonCloudWatchAgent]
aws_access_key_id = my_access_key
aws_secret_access_key = my_secret_key
```

El asistente muestra las credenciales predeterminadas, las credenciales de `AmazonCloudWatchAgent` y una opción `Others`. Puede seleccionar las credenciales que desea utilizar. Si elige `Others`, puede especificar las credenciales.

En *`my_access_key`* y *`my_secret_key`*, utilice las claves del usuario de IAM que tiene los permisos para realizar escrituras en el almacén de parámetros de Systems Manager. Para obtener más información sobre los usuarios de IAM necesarios para el agente de CloudWatch, consulte Cree usuarios de IAM para utilizarlos con el agente de CloudWatch en servidores en las instalaciones.

En el archivo de configuración de AWS, puede especificar la región a la que el agente envía métricas, si es diferente de la sección `[default]`. El valor predeterminado es publicar las métricas en la región donde se encuentra la instancia de Amazon EC2. Si las métricas se deben publicar en otra región, puede especificar dicha región aquí. En el siguiente ejemplo, las métricas se publican en la región `us-west-1`.

```
[AmazonCloudWatchAgent]
```

```
region = us-west-1
```

Ejecute el asistente de configuración del agente de CloudWatch

Para crear el archivo de configuración del agente de CloudWatch

1. Inicie el asistente de configuración del agente de CloudWatch al ingresar lo siguiente:

   ```
   sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-config-wizard
   ```

   En un servidor que ejecute Windows Server, ejecute los siguientes comandos para que puedan iniciar el asistente:

   ```
   cd "C:\Program Files\Amazon\AmazonCloudWatchAgent"
   ```

   ```
   .\amazon-cloudwatch-agent-config-wizard.exe
   ```

2. Responda a las preguntas para personalizar el archivo de configuración de su servidor.
3. Si almacena el archivo de configuración localmente, el archivo de configuración `config.json` se almacena en `/opt/aws/amazon-cloudwatch-agent/bin/` en servidores Linux y se almacena en `C:\Program Files\Amazon\AmazonCloudWatchAgent` en Windows Server. A continuación, puede copiar este archivo en otros servidores en los que desee instalar el agente.

   Si va a utilizar Systems Manager para instalar y configurar el agente, asegúrese de responder Yes (Sí) cuando se le pregunte si desea almacenar el archivo en el almacén de parámetros de Systems Manager. También puede optar por almacenar el archivo en el almacén de parámetros aunque no utilice SSM Agent para instalar el agente de CloudWatch. Para poder almacenar el archivo en el almacén de parámetros, debe utilizar un rol de IAM con los permisos suficientes. Para obtener más información, consulte Cree roles y usuarios de IAM para usarlos con el agente de CloudWatch.

Conjuntos predefinidos de métricas del agente de CloudWatch

El asistente está configurado con conjuntos predefinidos de métricas, con diferentes niveles de detalle. Estos conjuntos de métricas se muestran en las siguientes tablas. Para obtener más información sobre estas métricas, consulte Métricas que el agente de CloudWatch ha recopilado.

> **Note**
>
> El almacén de parámetros admite parámetros en los niveles estándar y avanzado. Estos niveles de parámetros no están relacionados con los niveles de detalles de métricas Basic, Estándar y Avanzado que se describen en estas tablas.

Instancias de Amazon EC2 con Linux

Nivel de detalle	Métricas incluidas
Básica	Memoria: mem_used_percent Disco: disk_used_percent Las métricas de `disk` como `disk_used_percent` tienen una dimensión para `Partition`, lo que significa que el número de métricas personalizadas generadas depende del número de particiones asociadas a la instancia. El número de particiones de disco depende de la AMI que esté utilizando y del número de volúmenes de Amazon EBS que adjunte al servidor.
Estándar	CPU: `cpu_usage_idle`, `cpu_usage_iowait`, `cpu_usage_user`, `cpu_usage_system` Disco: `disk_used_percent`, `disk_inodes_free` E/S de disco: `diskio_io_time` Memoria: `mem_used_percent` Intercambio: `swap_used_percent`
Advanced (Avanzado)	CPU: `cpu_usage_idle`, `cpu_usage_iowait`, `cpu_usage_user`, `cpu_usage_system` Disco: `disk_used_percent`, `disk_inodes_free` E/S de disco: `diskio_io_time`, `diskio_write_bytes`, `diskio_read_bytes`, `diskio_writes`, `diskio_reads`

Nivel de detalle	Métricas incluidas
	Memoria: `mem_used_percent` Estado de red: `netstat_tcp_established` , `netstat_tcp_time_wait` Intercambio: `swap_used_percent`

Servidores locales con Linux

Nivel de detalle	Métricas incluidas
Básica	Disco: `disk_used_percent` E/S de disco: `diskio_write_bytes` , `diskio_read_bytes` , `diskio_writes` , `diskio_reads` Memoria: `mem_used_percent` Red: `net_bytes_sent` , `net_bytes_recv` , `net_packets_sent` , `net_packets_recv` Intercambio: `swap_used_percent`
Estándar	CPU: `cpu_usage_idle` , `cpu_usage_iowait` Disco: `disk_used_percent` , `disk_inodes_free` E/S de disco: `diskio_io_time` , `diskio_write_bytes` , `diskio_read_bytes` , `diskio_writes` , `diskio_reads` Memoria: `mem_used_percent` Red: `net_bytes_sent` , `net_bytes_recv` , `net_packets_sent` , `net_packets_recv` Intercambio: `swap_used_percent`

Nivel de detalle	Métricas incluidas
Advanced (Avanzado)	CPU: `cpu_usage_guest` , `cpu_usage_idle` , `cpu_usage_iowait` , `cpu_usage_steal` , `cpu_usage_user` , `cpu_usage_system` Disco: `disk_used_percent` , `disk_inodes_free` E/S de disco: `diskio_io_time` , `diskio_write_bytes` , `diskio_read_bytes` , `diskio_writes` , `diskio_reads` Memoria: `mem_used_percent` Red: `net_bytes_sent` , `net_bytes_recv` , `net_packets_sent` , `net_packets_recv` Estado de red: `netstat_tcp_established` , `netstat_tcp_time_wait` Intercambio: `swap_used_percent`

Instancias de Amazon EC2 con Windows Server

> **Note**
>
> Los nombres de las métricas que aparecen en esta tabla muestran el aspecto que tienen las métricas cuando se visualizan en la consola. Es posible que el nombre real de la métrica no incluya la primera palabra. Por ejemplo, el nombre real de la métrica `LogicalDisk % Free Space` es simplemente `% Free Space`.

Nivel de detalle	Métricas incluidas
Básica	Memoria: `Memory % Committed Bytes In Use` Disco lógico: `LogicalDisk % Free Space`
Estándar	Memoria: `Memory % Committed Bytes In Use` Paginación: `Paging File % Usage`

Nivel de detalle	Métricas incluidas
	Procesador: `Processor % Idle Time`, `Processor % Interrupt Time`, `Processor % User Time` Disco físico: `PhysicalDisk % Disk Time` Disco lógico: `LogicalDisk % Free Space`
Advanced (Avanzado)	Memoria: `Memory % Committed Bytes In Use` Paginación: `Paging File % Usage` Procesador: `Processor % Idle Time`, `Processor % Interrupt Time`, `Processor % User Time` Disco lógico: `LogicalDisk % Free Space` Disco físico: `PhysicalDisk % Disk Time` , `PhysicalDisk Disk Write Bytes/sec` , `PhysicalDisk Disk Read Bytes/sec` , `PhysicalDisk Disk Writes/sec` , `PhysicalDisk Disk Reads/sec` TCP: `TCPv4 Connections Established` , `TCPv6 Connections Established`

Servidor local con Windows Server

> **Note**
>
> Los nombres de las métricas que aparecen en esta tabla muestran el aspecto que tienen las métricas cuando se visualizan en la consola. Es posible que el nombre real de la métrica no incluya la primera palabra. Por ejemplo, el nombre real de la métrica `LogicalDisk % Free Space` es simplemente `% Free Space`.

Nivel de detalle	Métricas incluidas
Básica	Paginación:`Paging File % Usage`

Nivel de detalle	Métricas incluidas
	Procesador: `Processor % Processor Time` Disco lógico: `LogicalDisk % Free Space` Disco físico: `PhysicalDisk Disk Write Bytes/sec` , `PhysicalDisk Disk Read Bytes/sec` , `PhysicalDisk Disk Writes/sec` , `PhysicalDisk Disk Reads/sec` Memoria: `Memory % Committed Bytes In Use` Interfaz de red: `Network Interface Bytes Sent/sec`, `Network Interface Bytes Received/sec` , `Network Interface Packets Sent/sec`, `Network Interface Packets Received/sec`
Estándar	Paginación: `Paging File % Usage` Procesador: `Processor % Processor Time`, `Processor % Idle Time`, `Processor % Interrupt Time` Disco lógico: `LogicalDisk % Free Space` Disco físico: `PhysicalDisk % Disk Time` , `PhysicalDisk Disk Write Bytes/sec` , `PhysicalDisk Disk Read Bytes/sec` , `PhysicalDisk Disk Writes/sec` , `PhysicalDisk Disk Reads/sec` Memoria: `Memory % Committed Bytes In Use` Interfaz de red: `Network Interface Bytes Sent/sec`, `Network Interface Bytes Received/sec` , `Network Interface Packets Sent/sec`, `Network Interface Packets Received/sec`

Nivel de detalle	Métricas incluidas
Advanced (Avanzado)	Paginación:`Paging File % Usage` Procesador: `Processor % Processor Time`, `Processor % Idle Time`, `Processor % Interrupt Time`, `Processor % User Time` Disco lógico: `LogicalDisk % Free Space` Disco físico: `PhysicalDisk % Disk Time` , `PhysicalDisk Disk Write Bytes/sec` , `PhysicalDisk Disk Read Bytes/sec` , `PhysicalDisk Disk Writes/sec` , `PhysicalDisk Disk Reads/sec` Memoria: `Memory % Committed Bytes In Use` Interfaz de red: `Network Interface Bytes Sent/sec`, `Network Interface Bytes Received/sec` , `Network Interface Packets Sent/sec`, `Network Interface Packets Received/sec` TCP: `TCPv4 Connections Established` , `TCPv6 Connections Established`

Cree o edite de forma manual el archivo de configuración del agente de CloudWatch

El archivo de configuración del agente de CloudWatch es un archivo JSON con cuatro secciones: `agent`, `metrics`, `logs` y `traces`, que se describen a continuación:

- La sección `agent` incluye los campos de la configuración general del agente.
- La sección `metrics` especifica las métricas personalizadas para la recopilación y publicación en CloudWatch. Si utiliza el agente únicamente para recopilar registros, puede omitir la sección `metrics` del archivo.
- La sección `logs` especifica qué archivos de registro se publican en CloudWatch Logs. Se pueden incluir los eventos del registro de eventos de Windows, si el servidor ejecuta Windows Server.
- La sección `traces` especifica las fuentes de los seguimientos que se recopilan y a las que se envían a AWS X-Ray.

En las secciones siguientes se explican la estructura y los campos de este archivo JSON. También puede ver la definición de esquema de este archivo de configuración. La definición de esquema se encuentra en *installation-directory*/doc/amazon-cloudwatch-agent-schema.json en los servidores Linux y en *installation-directory*/amazon-cloudwatch-agent-schema.json en los servidores con Windows Server.

Si crea o edita el archivo de configuración del agente de manualmente, puede asignarle cualquier nombre. Para simplificar la solución de problemas, le recomendamos que asigne el nombre /opt/aws/amazon-cloudwatch-agent/etc/amazon-cloudwatch-agent.json en un servidor Linux y $Env:ProgramData\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent.json en los servidores con Windows Server. Después de crear el archivo, puede copiarlo en otros servidores en los que desee instalar el agente.

> **Note**
>
> Las métricas, los registros y los seguimientos recopilados por el agente de CloudWatch conllevan cargos. Para obtener más información sobre precios, consulte Precios de Amazon CloudWatch.

Archivo de configuración del agente de CloudWatch: sección del agente

La sección agent puede incluir los siguientes campos. El asistente no crea una sección agent. En su lugar, el asistente la omite y usa los valores predeterminados de todos los campos de esta sección.

- metrics_collection_interval: opcional. Especifica la frecuencia con que se deben recopilar todas las métricas especificadas en este archivo de configuración. Puede invalidar este valor para determinados tipos de métricas.

  Este valor se especifica en segundos. Por ejemplo, indicar 10 hace que las métricas se deben recopilar cada 10 segundos y establecerlo en 300 especifica que las métricas se deben recopilar cada 5 minutos.

  Si configura este valor por debajo de 60 segundos, cada métrica se recopila como una métrica de alta resolución. Para obtener más información acerca de las métricas de alta resolución, consulte Métricas de alta resolución.

  El valor predeterminado es 60.

- `region`: especifica la región que se va a utilizar para el punto de enlace de CloudWatch cuando se supervisa una instancia de Amazon EC2. Las métricas recopiladas se envían a esta región, como `us-west-1`. Si omite este campo, el agente envía métricas a la región donde se encuentra la instancia de Amazon EC2.

  Si está supervisando un servidor en las instalaciones, este campo no se utiliza y el agente lee la región del perfil `AmazonCloudWatchAgent` del archivo de configuración de AWS.
- `credentials`: especifica un rol de IAM que se usará al enviar métricas, registros y seguimientos a una cuenta de AWS diferente. Si se especifica, este campo contiene un parámetro, `role_arn`.
  - `role_arn`: especifica el nombre de recurso de Amazon (ARN) de un rol de IAM que se va a utilizar para la autenticación al enviar métricas, registros y seguimientos a una cuenta de AWS diferente. Para obtener más información, consulte Envío de métricas, registros y seguimientos a una cuenta diferente.
- `debug`: opcional. Especifica la ejecución del agente de CloudWatch con mensajes de registro de depuración. El valor predeterminado es `false`.
- `aws_sdk_log_level`: opcional. Se admite en la versión 1.247350.0 y versiones posteriores del agente de CloudWatch.

  Puede especificar este campo para que el agente realice el registro para puntos de conexión SDK AWS. El valor de este campo puede incluir una o más de las siguientes opciones. Separe varias opciones con el character `|`.
  - `LogDebug`
  - `LogDebugWithSigning`
  - `LogDebugWithHTTPBody`
  - `LogDebugRequestRetries`
  - `LogDebugWithEventStreamBody`

  Para obtener más información sobre estas opciones, consulte LogLevelType.
- `logfile`: especifica la ubicación en la que el agente de CloudWatch ingresa los mensajes de registro. Si especifica una cadena vacía, el registro se envía a stderr. Si no especifica esta opción, las ubicaciones predeterminadas son las siguientes:
  - Linux: `/opt/aws/amazon-cloudwatch-agent/logs/amazon-cloudwatch-agent.log`
  - Windows Server: `c:\\ProgramData\\Amazon\\CloudWatchAgent\\Logs\\amazon-cloudwatch-agent.log`

El agente de CloudWatch rota automáticamente el archivo de registro que crea. Un archivo de registro rota cuando su tamaño alcanza los 100 MB. El agente conserva los archivos de registro rotados durante un máximo de siete días y mantiene cinco archivos de registro de copia de seguridad que se han rotado. Los archivos de registro de copia de seguridad tienen una marca temporal añadida al nombre de archivo. La marca de tiempo muestra la fecha y la hora en que se rotó el archivo (por ejemplo, `amazon-cloudwatch-agent-2018-06-08T21-01-50.247.log.gz`).

- `omit_hostname`: opcional. De forma predeterminada, el nombre del host se publica como una dimensión de métricas que el agente recopila.`append_dimensionsmetrics` Establezca `omit_hostname` a `true` para evitar que el nombre de host se publique como una dimensión incluso si usted no está utilizando `append_dimensions`. El valor predeterminado es `false`.
- `run_as_user`: opcional. Especifica el usuario que se va a utilizar para ejecutar el agente de CloudWatch. Si no especifica este parámetro, se utiliza el usuario raíz. Esta opción solo es válida en los servidores Linux.

  Si especifica esta opción, el usuario debe haber sido creado antes de iniciar el agente de CloudWatch. Para obtener más información, consulte Ejecución del agente de CloudWatch como otro usuario.
- `user_agent`: opcional. Especifique la secuencia `user-agent` que utiliza el agente de CloudWatch cuando realiza llamadas a la API al backend de CloudWatch. El valor predeterminado es una cadena que consta de la versión del agente, la versión del lenguaje de programación Go que se ha utilizado para compilar el agente, el tiempo de ejecución del sistema operativo y de la arquitectura, el tiempo de compilación y los complementos habilitados.
- `usage_data`: opcional. De forma predeterminada, el agente de CloudWatch envía datos de estado y rendimiento sobre sí mismo a CloudWatch cada vez que publica métricas o registros en CloudWatch. Estos datos no suponen ningún coste para usted. Puede impedir que el agente envíe estos datos especificando `false` para `usage_data`. Si omite este parámetro, `true` se utiliza el valor por defecto y el agente envían datos de mantenimiento y rendimiento.

  Si establece este valor en `false`, debe detener y reiniciar el agente para que surta efecto.

A continuación se muestra un ejemplo de una sección `agent`.

```
"agent": {
   "metrics_collection_interval": 60,
   "region": "us-west-1",
```

```
  "logfile": "/opt/aws/amazon-cloudwatch-agent/logs/amazon-cloudwatch-agent.log",
  "debug": false,
  "run_as_user": "cwagent"
 }
```

Archivo de configuración del agente de CloudWatch: sección de métricas

Campos comunes para Linux y Windows

En los servidores con Linux o Windows Server, la sección `metrics` incluye los siguientes campos:

- `namespace`: opcional. El espacio de nombres que se usará para las métricas recopiladas por el agente. El valor predeterminado es `CWAgent`. La longitud máxima es de 255 caracteres. A continuación, se muestra un ejemplo:

  ```
  {
    "metrics": {
      "namespace": "Development/Product1Metrics",
     ......
     },
  }
  ```

- `append_dimensions`: opcional. Agrega las dimensiones de métricas de Amazon EC2 a todas las métricas que el agente recopila. Esto también hace que el agente no publique el nombre del host como una dimensión.

  Los únicos pares clave-valor admitidos se muestran en la siguiente lista.`append_dimensions` Cualquier otro par clave-valor se omite. El agente admite estos pares de clave-valor tal y como se muestran en la siguiente lista. No puede cambiar los valores clave para publicar nombres de dimensiones diferentes para ellos.

  - `"ImageId":"${aws:ImageId}"` establece el ID de AMI de la instancia como el valor de la dimensión `ImageId`.
  - `"InstanceId":"${aws:InstanceId}"` establece el ID de instancia de la instancia como el valor de la dimensión `InstanceId`.
  - `"InstanceType":"${aws:InstanceType}"` establece el tipo de instancia de la instancia como el valor de la dimensión `InstanceType`.
  - `"AutoScalingGroupName":"${aws:AutoScalingGroupName}"` establece el nombre del grupo de Auto Scaling de la instancia como el valor de la dimensión `AutoScalingGroupName`.

Si desea añadir dimensiones a las métricas con pares de clave-valor arbitrarios, utilice el parámetro `append_dimensions` en el campo para ese tipo concreto de métrica.

Si especifica un valor que depende de metadatos de Amazon EC2 y utiliza proxies, debe asegurarse de que el servidor puede obtener acceso al punto de enlace de Amazon EC2. Para obtener más información sobre estos puntos de conexión, consulte Amazon Elastic Compute Cloud (Amazon EC2) en Referencia general de Amazon Web Services.

- `aggregation_dimensions`: opcional. Especifica las dimensiones que se agregarán a las métricas recopiladas. Por ejemplo, si acumula métricas en la dimensión `AutoScalingGroupName`, se agregan las métricas de todas las instancias de cada grupo de Auto Scaling y se pueden ver como un conjunto.

  Puede acumular las métricas en una o varias dimensiones. Por ejemplo, si se especifica `[["InstanceId"], ["InstanceType"], ["InstanceId","InstanceType"]]`, se agrupan las métricas por ID de instancia individualmente, por tipo de instancia individualmente y por la combinación de las dos dimensiones.

  También puede especificar `[]` para acumular todas las métricas en una sola colección y descartar todas las dimensiones.

- `endpoint_override`: especifica un punto de enlace FIPS o un vínculo privado que se va a utilizar como punto final donde el agente envía métricas. Si lo especifica y establece un enlace privado, podrá enviar las métricas a un punto de enlace de la Amazon VPC. Para obtener más información, consulte What Is Amazon VPC? (¿Qué es Amazon VPC?)

  El valor de `endpoint_override` debe ser una cadena que sea una URL.

  Por ejemplo, la siguiente parte de la sección de métricas del archivo de configuración establece que el agente utilizará un punto de enlace de la VPC al enviar métricas.

```
{
  "metrics": {
    "endpoint_override": "vpce-XXXXXXXXXXXXXXXXXXXXXXXX.monitoring.us-
east-1.vpce.amazonaws.com",
   ......
   },
}
```

- `metrics_collected`: obligatorio. Especifica qué métricas se van a recopilar, incluidas las métricas personalizadas recopiladas a través de `StatsD` o `collectd`. Esta sección incluye varias subsecciones.

  El contenido de la sección `metrics_collected` dependerá de si este archivo de configuración es para un servidor con Linux o Windows Server.
- `force_flush_interval`: especifica en segundos la cantidad máxima de tiempo que las métricas permanecen en el búfer de memoria antes de enviarse al servidor. Independientemente del valor de esta opción, si el tamaño de las métricas en el búfer alcanza 1 MB o 1000 métricas diferentes, las métricas se envían inmediatamente al servidor.

  El valor predeterminado es 60.
- `credentials`: especifica un rol de IAM que se usará al enviar métricas a una cuenta diferente. Si se especifica, este campo contiene un parámetro, `role_arn`.
  - `role_arn`: especifica el ARN de un rol de IAM que se utilizará para la autenticación al enviar métricas a una cuenta diferente. Para obtener más información, consulte Envío de métricas, registros y seguimientos a una cuenta diferente. Si se especifica aquí, este valor sobrescribe el valor de `role_arn` especificado en la sección `agent` del archivo de configuración, si existe.

Sección de Linux

En los servidores que ejecutan Linux, la sección `metrics_collected` del archivo de configuración también puede contener los siguientes campos.

Muchos de estos campos pueden incluir una sección `measurement` donde se muestran las métricas que se deben recopilar para ese recurso. Estas secciones `measurement` pueden especificar el nombre completo de la métrica, por ejemplo, `swap_used`, o simplemente la parte del nombre de la métrica que se añadirá al tipo de recurso. Por ejemplo, si se especifica `reads` en la sección `measurement` de la sección `diskio`, se recopilará la métrica `diskio_reads`.

- `collectd`: opcional. Indica que desea recuperar métricas personalizadas mediante el protocolo `collectd`. El software `collectd` se utiliza para enviar las métricas al agente de CloudWatch. Para obtener más información sobre las opciones de configuración disponibles para collectd, consulte Recuperación de las métricas personalizadas con collectd.
- `cpu`: opcional. Especifica qué métricas de CPU se van a recopilar. Esta sección solo es válida para las instancias de Linux. Debe incluir al menos uno de los campos `resources` y `totalcpu` para todas las métricas de CPU que se recopilen. Esta sección puede incluir los siguientes campos.

- `drop_original_metrics`: opcional. Si utiliza el campo `aggregation_dimensions` de la sección `metrics` para agrupar las métricas en resultados agregados, de forma predeterminada, el agente envía tanto las métricas agregadas como las métricas originales, separadas para cada valor de la dimensión. Si no desea que las métricas originales se envíen a CloudWatch, puede especificar este parámetro con una lista de métricas. No se notifican a CloudWatch las métricas especificadas junto con este parámetro por dimensión. En su lugar, solo se registran las métricas agregadas. Esto reduce la cantidad de métricas que recopila el agente, lo que reduce los costes.
- `resources`: opcional. Especifique este campo con un valor de `*` para provocar que se recopilen métricas por CPU. El único valor permitido es `*`.
- `totalcpu`: opcional. Especifica si se registrarán las métricas de cpu agregadas en todos los núcleos de cpu. El valor predeterminado es true.
- `measurement`: especifica la matriz de métricas de la cpu que se recopilarán. Los posibles valores son `time_active`, `time_guest`, `time_guest_nice`, `time_idle`, `time_iowait`, `time_irq`, `time_nice`, `time_softirq`, `time_steal`, `time_system`, `time_user`, `usage_active`, `usage_guest`, `usage_guest_nice`, `usage_idle`, `usage_iowait`, `usage_irq`, `usage_nice`, `usage_softirq`, `usage_steal`, `usage_system` y `usage_user`. Este campo es obligatorio si incluye `cpu`.

  De forma predeterminada, la unidad de las métricas de `cpu_usage_*` es `Percent`, y las métricas de `cpu_time_*` no tienen una unidad.

  En la entrada de cada métrica individual, puede especificar, si lo desea, uno o los dos valores siguientes:

  - `rename`: especifica un nombre diferente para esta métrica.
  - `unit`: especifica la unidad que se usará para esta métrica, lo que anula la unidad predeterminada de `None` para la métrica. La unidad que especifique debe ser una unidad de métrica de CloudWatch válida, tal como se muestra en la descripción de `Unit` en MetricDatum.
- `metrics_collection_interval`: opcional. Especifica la frecuencia con la que se recopilarán las métricas de cpu, lo que anula el valor de `metrics_collection_interval` global especificado en la sección `agent` del archivo de configuración.

  Este valor se especifica en segundos. Por ejemplo, indicar 10 hace que las métricas se deben recopilar cada 10 segundos y establecerlo en 300 especifica que las métricas se deben recopilar cada 5 minutos.

Si configura este valor por debajo de 60 segundos, cada métrica se recopila como una métrica de alta resolución. Para obtener más información acerca de las métricas de alta resolución, consulte Métricas de alta resolución.

- `append_dimensions`: opcional. Las dimensiones adicionales que se utilizarán únicamente para las métricas de cpu. Si especifica este campo, se usa además de las dimensiones especificadas en el campo `append_dimensions` global que se utiliza para todos los tipos de métricas recopiladas por el agente.

- `disk`: opcional. Especifica qué métricas de disk se recopilarán. Esta sección solo es válida para las instancias de Linux. Esta sección puede incluir los siguientes campos.
  - `drop_original_metrics`: opcional. Si utiliza el campo `aggregation_dimensions` de la sección `metrics` para agrupar las métricas en resultados agregados, de forma predeterminada, el agente envía tanto las métricas agregadas como las métricas originales, separadas para cada valor de la dimensión. Si no desea que las métricas originales se envíen a CloudWatch, puede especificar este parámetro con una lista de métricas. No se notifican a CloudWatch las métricas especificadas junto con este parámetro por dimensión. En su lugar, solo se registran las métricas agregadas. Esto reduce la cantidad de métricas que recopila el agente, lo que reduce los costes.
  - `resources`: opcional. Especifica una matriz de puntos de montaje de disco. Este campo limita a CloudWatch para recopilar métricas únicamente de los puntos de montaje en la lista. Puede especificar * como el valor para recopilar métricas de todos los puntos de montaje. El valor predeterminado es recopilar métricas de todos los puntos de montaje.
  - `measurement`: especifica la matriz de métricas de disco que se recopilarán. Los posibles valores son `free`, `total`, `used`, `used_percent`, `inodes_free`, `inodes_used` y `inodes_total`. Este campo es obligatorio si incluye `disk`.

    > **Note**
    >
    > Las métricas de `disk` tienen una dimensión para `Partition`, lo que significa que el número de métricas personalizadas generadas depende del número de particiones asociadas a la instancia. El número de particiones de disco del que dispone depende de la AMI que esté utilizando y del número de volúmenes de Amazon EBS que adjunte al servidor.

Para ver las unidades predeterminadas de cada métrica de `disk`, consulte Métricas que el agente de CloudWatch recopila en instancias de Linux y de macOS.

En la entrada de cada métrica individual, puede especificar, si lo desea, uno o los dos valores siguientes:

- `rename`: especifica un nombre diferente para esta métrica.
- `unit`: especifica la unidad que se usará para esta métrica, lo que anula la unidad predeterminada de `None` de `None` para la métrica. La unidad que especifique debe ser una unidad de métrica de CloudWatch válida, tal como se muestra en la descripción de `Unit` en MetricDatum.

- `ignore_file_system_types`: especifica los tipos de sistema de archivos que se excluirán al recopilar métricas de disco. Los valores válidos incluyen `sysfs`, `devtmpfs`, etc.
- `drop_device`: cambiarlo a `true` hace que el `Device` no se incluya como una dimensión para las métricas de disco.

  Evitar que el `Device` se utilice como dimensión puede resultar útil en las instancias que utilizan el sistema Nitro, ya que en dichas instancias los nombres de dispositivo cambian para cada montaje de disco cuando se reinicia la instancia. Esto puede provocar que los datos de las métricas sean incoherentes y que las alarmas basadas en estas métricas pasen al estado `INSUFFICIENT DATA`.

  El valor predeterminado es `false`.
- `metrics_collection_interval`: opcional. Especifica la frecuencia con la que se recopilarán las métricas de disco, lo que anula el valor de `metrics_collection_interval` global especificado en la sección `agent` del archivo de configuración.

  Este valor se especifica en segundos.

  Si configura este valor por debajo de 60 segundos, cada métrica se recopila como una métrica de alta resolución. Para obtener más información, consulte Métricas de alta resolución.
- `append_dimensions`: opcional. Las dimensiones adicionales que se utilizarán únicamente para las métricas de disco. Si especifica este campo, se usa además de las dimensiones especificadas en el campo `append_dimensions` que se usa para todos los tipos de métricas recopiladas por el agente.

- `diskio`: opcional. Especifica qué métricas de entrada/salida de disco se van a recopilar. Esta sección solo es válida para las instancias de Linux. Esta sección puede incluir los siguientes campos.
  - `drop_original_metrics`: opcional. Si utiliza el campo `aggregation_dimensions` de la sección `metrics` para agrupar las métricas en resultados agregados, de forma predeterminada, el agente envía tanto las métricas agregadas como las métricas originales, separadas para cada valor de la dimensión. Si no desea que las métricas originales se envíen a CloudWatch, puede especificar este parámetro con una lista de métricas. No se notifican a CloudWatch las métricas especificadas junto con este parámetro por dimensión. En su lugar, solo se registran las métricas agregadas. Esto reduce la cantidad de métricas que recopila el agente, lo que reduce los costes.
  - `resources`: opcional. Si especifica una matriz de dispositivos, CloudWatch recopila métricas solo de esos dispositivos. De lo contrario, se recopilan las métricas de todos los dispositivos. También puede especificar * como el valor para recopilar métricas de todos los dispositivos.
  - `measurement`: especifica la matriz de las métricas de disco que se recopilarán. Los valores posibles son `reads`, `writes`, `read_bytes`, `write_bytes`, `read_time`, `write_time`, `io_time` e `iops_in_progress`. Este campo es obligatorio si incluye `diskio`.

    En la entrada de cada métrica individual, puede especificar, si lo desea, uno o los dos valores siguientes:
    - `rename`: especifica un nombre diferente para esta métrica.
    - `unit`: especifica la unidad que se usará para esta métrica, lo que anula la unidad predeterminada de `None` de `None` para la métrica. La unidad que especifique debe ser una unidad de métrica de CloudWatch válida, tal como se muestra en la descripción de `Unit` en MetricDatum.
  - `metrics_collection_interval`: opcional. Especifica la frecuencia con la que se recopilarán las métricas de entrada/salida de disco, lo que anula el valor de `metrics_collection_interval` global especificado en la sección `agent` del archivo de configuración.

    Este valor se especifica en segundos.

    Si configura este valor por debajo de 60 segundos, cada métrica se recopila como una métrica de alta resolución. Para obtener más información acerca de las métricas de alta resolución, consulte Métricas de alta resolución.

- `append_dimensions`: opcional. Las dimensiones adicionales que se utilizarán únicamente para las métricas de entrada/salida de disco. Si especifica este campo, se usa además de las dimensiones especificadas en el campo `append_dimensions` que se usa para todos los tipos de métricas recopiladas por el agente.

- `swap`: opcional. Especifica qué métricas de memoria de intercambio se recopilarán. Esta sección solo es válida para las instancias de Linux. Esta sección puede incluir los siguientes campos.
  - `drop_original_metrics`: opcional. Si utiliza el campo `aggregation_dimensions` de la sección `metrics` para agrupar las métricas en resultados agregados, de forma predeterminada, el agente envía tanto las métricas agregadas como las métricas originales, separadas para cada valor de la dimensión. Si no desea que las métricas originales se envíen a CloudWatch, puede especificar este parámetro con una lista de métricas. No se notifican a CloudWatch las métricas especificadas junto con este parámetro por dimensión. En su lugar, solo se registran las métricas agregadas. Esto reduce la cantidad de métricas que recopila el agente, lo que reduce los costes.
  - `measurement`: especifica la matriz de métricas de intercambio que se recopilarán. Los posibles valores son `free`, `used` y `used_percent`. Este campo es obligatorio si incluye `swap`.

    Para ver las unidades predeterminadas de cada métrica de `swap`, consulte Métricas que el agente de CloudWatch recopila en instancias de Linux y de macOS.

    En la entrada de cada métrica individual, puede especificar, si lo desea, uno o los dos valores siguientes:
    - `rename`: especifica un nombre diferente para esta métrica.
    - `unit`: especifica la unidad que se usará para esta métrica, lo que anula la unidad predeterminada de `None` de `None` para la métrica. La unidad que especifique debe ser una unidad de métrica de CloudWatch válida, tal como se muestra en la descripción de `Unit` en MetricDatum.
  - `metrics_collection_interval`: opcional. Especifica la frecuencia con la que se recopilarán las métricas de intercambio, lo que anula el valor de `metrics_collection_interval` global especificado en la sección `agent` del archivo de configuración.

    Este valor se especifica en segundos.

    Si configura este valor por debajo de 60 segundos, cada métrica se recopila como una métrica de alta resolución. Para obtener más información acerca de las métricas de alta resolución, consulte Métricas de alta resolución.

  - `append_dimensions`: opcional. Las dimensiones adicionales que se utilizarán únicamente para las métricas de intercambio. Si especifica este campo, se usa además de las dimensiones especificadas en el campo `append_dimensions` global que se usa para todos los tipos de métricas recopiladas por el agente. Se recopila como una métrica de alta resolución.
- `mem`: opcional. Especifica qué métricas de memoria se recopilarán. Esta sección solo es válida para las instancias de Linux. Esta sección puede incluir los siguientes campos.
  - `drop_original_metrics`: opcional. Si utiliza el campo `aggregation_dimensions` de la sección `metrics` para agrupar las métricas en resultados agregados, de forma predeterminada, el agente envía tanto las métricas agregadas como las métricas originales, separadas para cada valor de la dimensión. Si no desea que las métricas originales se envíen a CloudWatch, puede especificar este parámetro con una lista de métricas. No se notifican a CloudWatch las métricas especificadas junto con este parámetro por dimensión. En su lugar, solo se registran las métricas agregadas. Esto reduce la cantidad de métricas que recopila el agente, lo que reduce los costes.
  - `measurement`: especifica qué métricas de memoria se recopilarán. Los posibles valores son `active`, `available`, `available_percent`, `buffered`, `cached`, `free`, `inactive`, `total`, `used` y `used_percent`. Este campo es obligatorio si incluye `mem`.

    Para ver las unidades predeterminadas de cada métrica de `mem`, consulte Métricas que el agente de CloudWatch recopila en instancias de Linux y de macOS.

    En la entrada de cada métrica individual, puede especificar, si lo desea, uno o los dos valores siguientes:

    - `rename`: especifica un nombre diferente para esta métrica.
    - `unit`: especifica la unidad que se usará para esta métrica, lo que anula la unidad predeterminada de `None` para la métrica. La unidad que especifique debe ser una unidad de métrica de CloudWatch válida, tal como se muestra en la descripción de `Unit` en MetricDatum.
  - `metrics_collection_interval`: opcional. Especifica la frecuencia con la que se recopilarán las métricas de memoria, lo que anula el valor de `metrics_collection_interval` global especificado en la sección `agent` del archivo de configuración.

    Este valor se especifica en segundos.

Si configura este valor por debajo de 60 segundos, cada métrica se recopila como una métrica de alta resolución. Para obtener más información acerca de las métricas de alta resolución, consulte Métricas de alta resolución.

- `append_dimensions`: opcional. Las dimensiones adicionales que se utilizarán únicamente para las métricas de memoria. Si especifica este campo, se usa además de las dimensiones especificadas en el campo `append_dimensions` que se utiliza para todos los tipos de métricas recopiladas por el agente.

- `net`: opcional. Especifica qué métricas de red se recopilarán. Esta sección solo es válida para las instancias de Linux. Esta sección puede incluir los siguientes campos.
  - `drop_original_metrics`: opcional. Si utiliza el campo `aggregation_dimensions` de la sección `metrics` para agrupar las métricas en resultados agregados, de forma predeterminada, el agente envía tanto las métricas agregadas como las métricas originales, separadas para cada valor de la dimensión. Si no desea que las métricas originales se envíen a CloudWatch, puede especificar este parámetro con una lista de métricas. No se notifican a CloudWatch las métricas especificadas junto con este parámetro por dimensión. En su lugar, solo se registran las métricas agregadas. Esto reduce la cantidad de métricas que recopila el agente, lo que reduce los costes.
  - `resources`: opcional. Si especifica una matriz de interfaces de red, CloudWatch recopila métricas solo de esas interfaces. De lo contrario, se recopilan las métricas de todos los dispositivos. También puede especificar `*` como el valor para recopilar métricas de todas las interfaces.
  - `measurement`: especifica la matriz de métricas de red que se recopilarán. Los valores posibles son `bytes_sent`, `bytes_recv`, `drop_in`, `drop_out`, `err_in`, `err_out`, `packets_sent` e `packets_recv`. Este campo es obligatorio si incluye `net`.

    Para ver las unidades predeterminadas de cada métrica de `net`, consulte Métricas que el agente de CloudWatch recopila en instancias de Linux y de macOS.

    En la entrada de cada métrica individual, puede especificar, si lo desea, uno o los dos valores siguientes:
    - `rename`: especifica un nombre diferente para esta métrica.
    - `unit`: especifica la unidad que se usará para esta métrica, lo que anula la unidad predeterminada de `None` para la métrica. La unidad que especifique debe ser una unidad de métrica de CloudWatch válida, tal como se muestra en la descripción de `Unit` en MetricDatum.

- `metrics_collection_interval`: opcional. Especifica la frecuencia con la que se recopilarán las métricas de red, lo que anula el valor de `metrics_collection_interval` global especificado en la sección `agent` del archivo de configuración.

  Este valor se especifica en segundos. Por ejemplo, indicar 10 hace que las métricas se deben recopilar cada 10 segundos y establecerlo en 300 especifica que las métricas se deben recopilar cada 5 minutos.

  Si configura este valor por debajo de 60 segundos, cada métrica se recopila como una métrica de alta resolución. Para obtener más información acerca de las métricas de alta resolución, consulte Métricas de alta resolución.

- `append_dimensions`: opcional. Las dimensiones adicionales que se utilizarán únicamente para las métricas de red. Si especifica este campo, se usa además de las dimensiones especificadas en el campo `append_dimensions` que se utiliza para todos los tipos de métricas recopiladas por el agente.

- `netstat`: opcional. Especifica que se recopilen las métricas de estado de conexión TCP y de contraseñas UDP. Esta sección solo es válida para las instancias de Linux. Esta sección puede incluir los siguientes campos.

  - `drop_original_metrics`: opcional. Si utiliza el campo `aggregation_dimensions` de la sección `metrics` para agrupar las métricas en resultados agregados, de forma predeterminada, el agente envía tanto las métricas agregadas como las métricas originales, separadas para cada valor de la dimensión. Si no desea que las métricas originales se envíen a CloudWatch, puede especificar este parámetro con una lista de métricas. No se notifican a CloudWatch las métricas especificadas junto con este parámetro por dimensión. En su lugar, solo se registran las métricas agregadas. Esto reduce la cantidad de métricas que recopila el agente, lo que reduce los costes.

  - `measurement`: especifica la matriz de métricas de netstat que se recopilarán. Los posibles valores son `tcp_close`, `tcp_close_wait`, `tcp_closing`, `tcp_established`, `tcp_fin_wait1`, `tcp_fin_wait2`, `tcp_last_ack`, `tcp_listen`, `tcp_none`, `tcp_syn_sent`, `tcp_syn_recv`, `tcp_time_wait` y `udp_socket`. Este campo es obligatorio si incluye `netstat`.

    Para ver las unidades predeterminadas de cada métrica de `netstat`, consulte Métricas que el agente de CloudWatch recopila en instancias de Linux y de macOS.

    En la entrada de cada métrica individual, puede especificar, si lo desea, uno o los dos valores siguientes:

    - `rename`: especifica un nombre diferente para esta métrica.
    - `unit`: especifica la unidad que se usará para esta métrica, lo que anula la unidad predeterminada de `None` para la métrica. La unidad que especifique debe ser una unidad de métrica de CloudWatch válida, tal como se muestra en la descripción de `Unit` en MetricDatum.
  - `metrics_collection_interval`: opcional. Especifica la frecuencia con la que se recopilarán las métricas de estado de red, lo que anula el valor de `metrics_collection_interval` global especificado en la sección `agent` del archivo de configuración.

    Este valor se especifica en segundos.

    Si configura este valor por debajo de 60 segundos, cada métrica se recopila como una métrica de alta resolución. Para obtener más información acerca de las métricas de alta resolución, consulte Métricas de alta resolución.
  - `append_dimensions`: opcional. Las dimensiones adicionales que se utilizarán únicamente para las métricas de estado de red. Si especifica este campo, se usa además de las dimensiones especificadas en el campo `append_dimensions` que se utiliza para todos los tipos de métricas recopiladas por el agente.
- `processes`: opcional. Especifica qué métricas de proceso se recopilarán. Esta sección solo es válida para las instancias de Linux. Esta sección puede incluir los siguientes campos.
  - `drop_original_metrics`: opcional. Si utiliza el campo `aggregation_dimensions` de la sección `metrics` para agrupar las métricas en resultados agregados, de forma predeterminada, el agente envía tanto las métricas agregadas como las métricas originales, separadas para cada valor de la dimensión. Si no desea que las métricas originales se envíen a CloudWatch, puede especificar este parámetro con una lista de métricas. No se notifican a CloudWatch las métricas especificadas junto con este parámetro por dimensión. En su lugar, solo se registran las métricas agregadas. Esto reduce la cantidad de métricas que recopila el agente, lo que reduce los costes.
  - `measurement`: especifica la matriz de métricas de procesos que se recopilarán. Los posibles valores son `blocked`, `dead`, `idle`, `paging`, `running`, `sleeping`, `stopped`, `total`, `total_threads`, `wait` y `zombies`. Este campo es obligatorio si incluye `processes`.

    Para todas las métricas de `processes`, la unidad predeterminada es `None`.

    En la entrada de cada métrica individual, puede especificar, si lo desea, uno o los dos valores siguientes:

- `rename`: especifica un nombre diferente para esta métrica.
- `unit`: especifica la unidad que se usará para esta métrica, lo que anula la unidad predeterminada de `None` para la métrica. La unidad que especifique debe ser una unidad de métrica de CloudWatch válida, tal como se muestra en la descripción de `Unit` en MetricDatum.

- `metrics_collection_interval`: opcional. Especifica la frecuencia con la que se recopilarán las métricas de proceso, lo que anula el valor de `metrics_collection_interval` global especificado en la sección `agent` del archivo de configuración.

  Este valor se especifica en segundos. Por ejemplo, indicar 10 hace que las métricas se deben recopilar cada 10 segundos y establecerlo en 300 especifica que las métricas se deben recopilar cada 5 minutos.

  Si configura este valor por debajo de 60 segundos, cada métrica se recopila como una métrica de alta resolución. Para obtener más información, consulte Métricas de alta resolución.

- `append_dimensions`: opcional. Las dimensiones adicionales que se utilizarán únicamente para las métricas de proceso. Si especifica este campo, se usa además de las dimensiones especificadas en el campo `append_dimensions` que se utiliza para todos los tipos de métricas recopiladas por el agente.

- `nvidia_gpu`: opcional. Especifica que se recopilarán las métricas de GPU NVIDIA. Esta sección es válida solo para instancias Linux en hosts configurados con un acelerador de GPU NVIDIA y que tienen instalada la interfaz de administración del sistema NVIDIA (nvidia-smi).

  Las métricas de GPU NVIDIA que se recopilan llevan como prefijo la cadena `nvidia_smi_` para distinguirlas de las métricas recopiladas para otros tipos de aceleradores. Esta sección puede incluir los siguientes campos.

  - `drop_original_metrics`: opcional. Si utiliza el campo `aggregation_dimensions` de la sección `metrics` para agrupar las métricas en resultados agregados, de forma predeterminada, el agente envía tanto las métricas agregadas como las métricas originales, separadas para cada valor de la dimensión. Si no desea que las métricas originales se envíen a CloudWatch, puede especificar este parámetro con una lista de métricas. No se notifican a CloudWatch las métricas especificadas junto con este parámetro por dimensión. En su lugar, solo se registran las métricas agregadas. Esto reduce la cantidad de métricas que recopila el agente, lo que reduce los costes.

- `measurement`: especifica la matriz de métricas de GPU NVIDIA que se recopilarán. Para ver una lista de los valores posibles que se utilizarán aquí, consulte la columna Metric (Métrica) de la tabla de Recopilación de métricas de GPU NVIDIA.

  En la entrada de cada métrica, puede especificar, si así lo desea, uno de los valores siguientes o ambos:

  - `rename`: especifica un nombre diferente para esta métrica.
  - `unit`: especifica la unidad que se usará para esta métrica, lo que anula la unidad predeterminada de `None` para la métrica. La unidad que especifique debe ser una unidad de métrica de CloudWatch válida, tal como se muestra en la descripción de `Unit` en MetricDatum.
- `metrics_collection_interval`: opcional. Especifica la frecuencia con la que se recopilarán las métricas de GPU NVIDIA, lo que anula el valor de `metrics_collection_interval` global especificado en la sección `agent` del archivo de configuración.

- `procstat`: Opcional. Especifica que desea recuperar las métricas de procesos individuales. Para obtener más información sobre las opciones de configuración disponibles para procstat, consulte Recopilación de las métricas de procesos con el complemento procstat.
- `statsd`: opcional. Indica que desea recuperar métricas personalizadas mediante el protocolo `StatsD`. El agente de CloudWatch actúa como un daemon para el protocolo. Puede usar cualquier cliente `StatsD` estándar para enviar las métricas al agente de CloudWatch. Para obtener más información sobre las opciones de configuración disponibles para StatsD, consulte Recuperación de las métricas personalizadas con StatsD .
- `ethtool`: opcional. Indica que desea recuperar métricas personalizadas de red mediante el complemento `ethtool`. Este complemento puede importar tanto las métricas que recopila la herramienta estándar ethtool, como también las métricas de rendimiento de red de instancias de Amazon EC2. Para obtener más información sobre las opciones de configuración disponibles para ethtool, consulte Recopilación de las métricas de rendimiento de la red.

A continuación, se ofrece un ejemplo de una sección `metrics` de un servidor Linux. En este ejemplo, se recopilan tres métricas de CPU, tres métricas de estado de red, tres métricas de proceso y una métrica de disco, y se configura el agente para recibir métricas adicionales desde un cliente de `collectd`.

```
"metrics": {
    "aggregation_dimensions" : [["AutoScalingGroupName"], ["InstanceId",
 "InstanceType"],[]],
```

```
    "metrics_collected": {
      "collectd": {},
      "cpu": {
        "resources": [
          "*"
        ],
        "measurement": [
          {"name": "cpu_usage_idle", "rename": "CPU_USAGE_IDLE", "unit": "Percent"},
          {"name": "cpu_usage_nice", "unit": "Percent"},
          "cpu_usage_guest"
        ],
        "totalcpu": false,
        "drop_original_metrics": [ "cpu_usage_guest" ],
        "metrics_collection_interval": 10,
        "append_dimensions": {
          "test": "test1",
          "date": "2017-10-01"
        }
      },
      "netstat": {
        "measurement": [
          "tcp_established",
          "tcp_syn_sent",
          "tcp_close"
        ],
        "metrics_collection_interval": 60
      },
       "disk": {
        "measurement": [
          "used_percent"
        ],
        "resources": [
          "*"
        ],
        "drop_device": true
      },
      "processes": {
        "measurement": [
          "running",
          "sleeping",
          "dead"
        ]
      }
    },
```

```
    "append_dimensions": {
      "ImageId": "${aws:ImageId}",
      "InstanceId": "${aws:InstanceId}",
      "InstanceType": "${aws:InstanceType}",
      "AutoScalingGroupName": "${aws:AutoScalingGroupName}"
    }
  }
```

Windows Server

En la sección `metrics_collected` de Windows Server, puede tener subsecciones para cada objeto de rendimiento de Windows, como `Memory`, `Processor` y `LogicalDisk`. Para obtener información sobre qué objetos y contadores están disponibles, consulte Contadores de rendimiento en la documentación de Microsoft Windows.

En la subsección de cada uno de los objetos, especifique una matriz `measurement` de los contadores que se recopilarán. La matriz `measurement` es obligatoria para cada objeto que especifique en el archivo de configuración. También puede especificar un campo `resources` para asignar un nombre a las instancias de las que se recopilarán métricas. También puede especificar * para `resources` para recopilar métricas independientes de cada instancia. Si omite `resources` para los contadores que tienen instancias, los datos de todas las instancias se agregan en un conjunto. Si omite `resources` para los contadores que no tienen instancias, el agente de CloudWatch no recopila los contadores. Para determinar si los contadores tienen instancias, pueden usar uno de los siguientes comandos.

Powershell:

```
Get-Counter -ListSet *
```

Línea de comandos (no Powershell):

```
TypePerf.exe -q
```

En cada sección de objetos, también puede especificar los siguientes campos opcionales:

- `metrics_collection_interval`: opcional. Especifica la frecuencia con la que se recopilarán las métricas de este objeto, lo que anula el valor de `metrics_collection_interval` global especificado en la sección `agent` del archivo de configuración.

Este valor se especifica en segundos. Por ejemplo, indicar 10 hace que las métricas se deben recopilar cada 10 segundos y establecerlo en 300 especifica que las métricas se deben recopilar cada 5 minutos.

Si configura este valor por debajo de 60 segundos, cada métrica se recopila como una métrica de alta resolución. Para obtener más información, consulte Métricas de alta resolución.

- `append_dimensions`: opcional. Especifica las dimensiones adicionales que se van a utilizar únicamente para las métricas de este objeto. Si especifica este campo, se usa además de las dimensiones especificadas en el campo `append_dimensions` global que se utiliza para todos los tipos de métricas recopiladas por el agente.
- `drop_original_metrics`: opcional. Si utiliza el campo `aggregation_dimensions` de la sección `metrics` para agrupar las métricas en resultados agregados, de forma predeterminada, el agente envía tanto las métricas agregadas como las métricas originales, separadas para cada valor de la dimensión. Si no desea que las métricas originales se envíen a CloudWatch, puede especificar este parámetro con una lista de métricas. No se notifican a CloudWatch las métricas especificadas junto con este parámetro por dimensión. En su lugar, solo se registran las métricas agregadas. Esto reduce la cantidad de métricas que recopila el agente, lo que reduce los costes.

En cada sección de contadores, también puede especificar los siguientes campos opcionales:

- `rename`: especifica un nombre diferente que se usará en CloudWatch para esta métrica.
- `unit`: especifica la unidad que se usará para esta métrica. La unidad que especifique debe ser una unidad de métrica de CloudWatch válida, tal como se muestra en la descripción de `Unit` en MetricDatum.

Existen otras dos secciones opcionales que puede incluir en `metrics_collected`:

- `statsd`: permite recuperar las métricas personalizadas mediante el protocolo `StatsD`. El agente de CloudWatch actúa como un daemon para el protocolo. Puede usar cualquier cliente `StatsD` estándar para enviar las métricas al agente de CloudWatch. Para obtener más información, consulte Recuperación de las métricas personalizadas con StatsD .
- `procstat`: permite recuperar métricas de procesos individuales. Para obtener más información, consulte Recopilación de las métricas de procesos con el complemento procstat.

A continuación, se muestra una sección `metrics` de ejemplo para usarla en Windows Server. En este ejemplo, se recopilan muchas métricas de Windows y además se configura el equipo para que reciba métricas adicionales de un cliente de `StatsD`.

```
"metrics": {
    "metrics_collected": {
      "statsd": {},
      "Processor": {
        "measurement": [
          {"name": "% Idle Time", "rename": "CPU_IDLE", "unit": "Percent"},
          "% Interrupt Time",
          "% User Time",
          "% Processor Time"
        ],
        "resources": [
          "*"
        ],
        "append_dimensions": {
          "d1": "win_foo",
          "d2": "win_bar"
        }
      },
      "LogicalDisk": {
        "measurement": [
          {"name": "% Idle Time", "unit": "Percent"},
          {"name": "% Disk Read Time", "rename": "DISK_READ"},
          "% Disk Write Time"
        ],
        "resources": [
          "*"
        ]
      },
      "Memory": {
        "metrics_collection_interval": 5,
        "measurement": [
          "Available Bytes",
          "Cache Faults/sec",
          "Page Faults/sec",
          "Pages/sec"
        ],
        "append_dimensions": {
          "d3": "win_bo"
        }
```

```
      },
      "Network Interface": {
        "metrics_collection_interval": 5,
        "measurement": [
          "Bytes Received/sec",
          "Bytes Sent/sec",
          "Packets Received/sec",
          "Packets Sent/sec"
        ],
        "resources": [
          "*"
        ],
        "append_dimensions": {
          "d3": "win_bo"
        }
      },
      "System": {
        "measurement": [
          "Context Switches/sec",
          "System Calls/sec",
          "Processor Queue Length"
        ],
        "append_dimensions": {
          "d1": "win_foo",
          "d2": "win_bar"
        }
      }
    },
    "append_dimensions": {
      "ImageId": "${aws:ImageId}",
      "InstanceId": "${aws:InstanceId}",
      "InstanceType": "${aws:InstanceType}",
      "AutoScalingGroupName": "${aws:AutoScalingGroupName}"
    },
    "aggregation_dimensions" : [["ImageId"], ["InstanceId", "InstanceType"], ["d1"],[]]
    }
  }
```

Archivo de configuración del agente de CloudWatch: sección de registros

La sección `logs` incluye los siguientes campos:

- `logs_collected`: obligatorio si la sección `logs` está incluida. Especifica qué archivos de registros y registros de eventos de Windows se recopilarán del servidor. Puede incluir dos campos: `files` y `windows_events`.
  - `files`: especifica qué archivos de registro regulares va a recopilar el agente de CloudWatch. Contiene un campo, `collect_list`, que define estos archivos.
    - `collect_list`: obligatorio si `files` se incluye. Contiene una matriz de entradas, cada una de las cuales especifica un archivo de registro para recopilar. Cada una de estas entradas puede incluir los siguientes campos:
      - `file_path`: especifica la ruta del archivo de registro que se cargará en CloudWatch Logs. Se aceptan las reglas de concordancia glob de Unix estándar, como la adición de `**` como un superasterisco. Por ejemplo, especificar `/var/log/**.log` provoca que se recopilen todos los archivos `.log` del árbol de directorio `/var/log`. Para obtener más ejemplos, consulte la biblioteca Glob.

        También puede usar el asterisco estándar como comodín estándar. Por ejemplo, `/var/log/system.log*` busca archivos coincidentes, como `system.log_1111`, `system.log_2222`, etc., en `/var/log`.

        Solo se envía el último archivo a CloudWatch Logs en función de la hora de modificación del archivo. Le recomendamos utilizar comodines para especificar una serie de archivos del mismo tipo, como `access_log.2018-06-01-01` y `access_log.2018-06-01-02`, pero no varios tipos de archivos, como, por ejemplo, `access_log_80` y `access_log_443`. Para especificar varios tipos de archivos, agregue otra entrada de flujo de registros al archivo de configuración del agente para que cada tipo de archivo de registro vaya a un flujo de registros distinto.
      - `auto_removal`: opcional. Si es `true`, el agente de CloudWatch elimina automáticamente este archivo de registro tras leerlo y lo ha rotado. Por lo general, los archivos de registro se eliminan después de cargar todo su contenido en Registros de CloudWatch, pero si el agente llega al EOF (final del archivo) y también detecta otro archivo de registro más reciente que coincide con la misma `file_path`, el archivo de registro se elimina aunque el agente no haya podido enviar todos los registros del archivo de registro anterior. Por lo tanto, debe asegurarse de haber terminado de escribir en el archivo ANTIGUO antes de crear el archivo NUEVO. La biblioteca de rastreo RUST tiene una incompatibilidad conocida porque podría crear un nuevo archivo de registro y, a continuación, seguir intentando escribir en el archivo de registro ANTIGUO.

El agente solo quita archivos completos de los registros que crean varios archivos, como los registros que crean archivos distintos para cada fecha. Si un registro escribe continuamente en un solo archivo, no se elimina.

Si ya dispone de un método de rotación o eliminación de archivos de registro, le recomendamos que omita este campo o que lo establezca en `false`.

Si omite este campo, se usa el valor predeterminado de `false`.

- `log_group_name`: opcional. Especifica qué se usará como nombre de grupo de registros en CloudWatch Logs.

  Le recomendamos que use este campo para especificar un nombre de grupo de registro para evitar confusiones. Si omite `log_group_name`, el valor de `file_path` hasta el punto final se usa como el nombre del grupo de registro. Por ejemplo, si la ruta de archivo es `/tmp/TestLogFile.log.2017-07-11-14`, el nombre de grupo de registros es `/tmp/TestLogFile.log`.

  Si especifica un nombre del grupo de registro, puede usar `{instance_id}`, `{hostname}`, `{local_hostname}` y `{ip_address}` como variables en el nombre. `{hostname}` recupera el nombre de host de los metadatos de EC2, mientras que `{local_hostname}` usa el nombre de host del archivo de configuración de red.

  Si usa estas variables para crear muchos grupos de registro diferentes, tenga en cuenta el límite de 1 000 000 grupos de registro por cuenta y región.

  Entre los caracteres permitidos se incluyen a-z, A-Z, 0-9, “_” (guion bajo), “-” (guion medio), “/” (barra diagonal) y “.” (punto).

- `log_group_class`: opcional. Especifica qué clase de grupo de registros usar para el nuevo grupo de registros. Para obtener más información sobre las clases de grupos de registros, consulte Clases de registros.

  Los valores válidos son `STANDARD` y `INFREQUENT_ACCESS`. Si omite este campo, se usa el valor predeterminado de `STANDARD`.

  > **Important**
  >
  > Después de crear un grupo de registro, la clase no se puede cambiar.

- `log_stream_name`: opcional. Especifica qué se usará como nombre de flujo de registro en CloudWatch Logs. Como parte del nombre, puede usar `{instance_id}`, `{hostname}` `{local_hostname}` y `{ip_address}` como variables en el nombre. `{hostname}` recupera el nombre de host de los metadatos de EC2, mientras que `{local_hostname}` usa el nombre de host del archivo de configuración de red.

  Si omite este campo, se utilizará el valor del parámetro `log_stream_name` en la sección global `logs`. Si esto también se omite, se utiliza el valor predeterminado de `{instance_id}`.

  Si un flujo de registros no existe todavía, se crea automáticamente.

- `retention_in_days`: Opcional. Especifica la cantidad de días que se conservan los eventos de registro en el grupo de registros especificado.
  - Si el agente está creando este grupo de registros y especifica u omite este campo, la retención del nuevo grupo de registro se establecerá de manera que nunca venza.
  - Si este grupo de registros ya existe y especifica este campo, se utiliza la nueva retención que especifique. Si omite este campo para un grupo de registros que ya existe, no se modifica la retención del grupo de registros.

    El asistente del agente de CloudWatch utiliza `-1` como valor predeterminado para este campo cuando se utiliza para crear el archivo de configuración del agente y no se especifica un valor para la retención de registros. Este valor de `-1` establecido por el asistente especifica que los eventos del grupo de registro no caducarán. Sin embargo, editar manualmente este valor a `-1` no tiene ningún efecto.

  Los valores válidos son 1, 3, 5, 7, 14, 30, 60, 90, 120, 150, 180, 365, 400, 545, 731, 1827, 2192, 2557, 2922, 3288 y 3653.

  Si configura el agente para que escriba varios flujos de registro en el mismo grupo de registros, especificar la `retention_in_days` en un solo lugar establecerá la retención de registros para todo el grupo de registros. Si especifica la `retention_in_days` para el mismo grupo de registros en varios lugares, la retención se establece si todos esos valores son iguales. Sin embargo, si se especifican diferentes valores de `retention_in_days` para el mismo grupo de registros en varios lugares, no se establecerá la retención de registros y el agente se detendrá, lo que devolverá un error.

> **Note**
>
> El rol o el usuario de IAM del agente deben tener la `logs:PutRetentionPolicy` para que puedan establecer políticas de retención. Para obtener más información, consulte Permitir que el agente de CloudWatch establezca la política de retención de registros.

> **Warning**
>
> Si establece la `retention_in_days` para un grupo de registros que ya existe, se eliminarán todos los registros de ese grupo de registros que se hayan publicado antes del número de días que especificó. Por ejemplo, si se configurara en 3, se eliminarían todos los registros de hace 3 días y más tiempo.

- `filters`: opcional. Puede contener una matriz de entradas. Cada una de estas entradas especifica una expresión regular y un tipo de filtro para determinar si se van a publicar o dejar de lado entradas de registro que coincidan con el filtro. Si omite este campo, todos los registros del archivo de registro se publican en CloudWatch Logs. Si incluye este campo, el agente procesa cada mensaje de registro con todos los filtros que especifique y solo los eventos de registro que pasen todos los filtros se publicarán en CloudWatch Logs. Las entradas de registro que no pasen todos los filtros se mantendrán en el archivo de registro del host igualmente, pero no se enviarán a CloudWatch Logs.

  Cada entrada de la matriz de filtros puede incluir los siguientes campos:

  - `type`: indica el tipo de filtro. Los valores válidos son `include` y `exclude`. Con `include`, la entrada de registro debe coincidir con la expresión que se publicará en CloudWatch Logs. Con `exclude`, cada entrada de registro que coincida con el filtro no se enviará a CloudWatch Logs.
  - `expression`: se trata de una cadena de expresión regular que sigue la Sintaxis de RE2.

    > **Note**
    >
    > El agente de CloudWatch no revisa el rendimiento de ninguna expresión regular que proporcione ni restringe el tiempo de ejecución de la evaluación de las expresiones regulares. Le recomendamos que tenga cuidado de no escribir

una expresión que resulte costosa de evaluar. Para obtener más información acerca de los posibles problemas, consulte Regular expression Denial of Service - ReDoS (Denegación de servicio de expresión regular: ReDoS)

Por ejemplo, el siguiente extracto del archivo de configuración del agente de CloudWatch publica registros que son solicitudes PUT y POST en CloudWatch Logs, pero excluye los registros procedentes de Firefox.

```
"collect_list": [
  {
    "file_path": "/opt/aws/amazon-cloudwatch-agent/logs/test.log",
    "log_group_name": "test.log",
    "log_stream_name": "test.log",
    "filters": [
      {
        "type": "exclude",
        "expression": "Firefox"
      },
      {
        "type": "include",
        "expression": "P(UT|OST)"
      }
    ]
  },
  .....
]
```

> **Note**
>
> El orden de los filtros en el archivo de configuración es importante en cuanto al rendimiento. En el ejemplo anterior, el agente deja de lado todos los registros que pasen el filtro de `Firefox` antes de empezar a evaluar el segundo filtro. Para que más de un filtro evalúe menos entradas de registro, aplique primero el filtro que espera que descarte más registros al archivo de configuración.

- `timezone`: Opcional. Especifica la zona horaria que se debe utilizar a la hora de poner marcas temporales en eventos de registro. Los valores válidos son `UTC` y `Local`. El valor predeterminado es `Local`.

Este parámetro se ignora si no especifica un valor para `timestamp_format`.

- `timestamp_format`: opcional. Especifica el formato de marca temporal, con texto sin formato y símbolos especiales que empiezan por %. Si omite este campo, se usa la hora actual. Si utiliza este campo, puede utilizar los símbolos de la lista siguiente como parte del formato.

  Si una única entrada de registro contiene dos marcas temporales que coinciden con el formato, se utiliza la primera marca temporal.

  Esta lista de símbolos es diferente de la lista que utiliza el agente anterior de CloudWatch Logs. Para ver un resumen de estas diferencias, consulte Diferencias de marcas de tiempo entre el agente unificado de CloudWatch y el agente de CloudWatch Logs anterior.

  `%y`

  Año sin siglo como un número decimal rellenado con ceros. Por ejemplo, 19 para representar 2019.

  `%Y`

  Año con siglo como número decimal. Por ejemplo, 2019.

  `%b`

  Mes como el nombre abreviado de la configuración regional

  `%B`

  Mes como el nombre completo de la configuración regional

  `%m`

  mes como número decimal relleno con ceros

  `%-m`

  Mes como número decimal (sin rellenar con ceros)

  `%d`

  día del mes como número decimal relleno con ceros

  `%-d`

  Día del mes como número decimal (sin rellenar con ceros)

`%A`

Nombre completo de semana como, por ejemplo, `Monday`

`%a`

Abreviatura de día de la semana como, por ejemplo, `Mon`

`%H`

Hora (en un formato de 24 horas) como número decimal rellenado con ceros

`%I`

Hora (en un formato de 12 horas) como número decimal rellenado con ceros

`%-I`

Hora (en un formato de 12 horas) como número decimal (sin rellenar con ceros)

`%p`

AM o PM

`%M`

Minutos como número decimal rellenado con ceros

`%-M`

Minutos como número decimal (sin rellenar con ceros)

`%S`

Segundos como número decimal rellenado con ceros

`%-S`

Segundos como número decimal (sin rellenar con ceros)

`%f`

Segundos fraccionarios como un número decimal (1-9 dígitos), rellenado con ceros a la izquierda.

`%Z`

Zona horaria, por ejemplo `PST`

`%z`

Zona horaria, expresada como la diferencia entre la zona horaria local y UTC. Por ejemplo, `-0700`. Solo se admite este formato. Por ejemplo, `-07:00` no es un formato válido.

- `multi_line_start_pattern`: especifica el patrón para identificar el inicio de un mensaje de registro. Un mensaje de registro consta de una línea que coincide con el patrón y de líneas siguientes que no coinciden con el patrón.

  Si omite este campo, se deshabilita el modo multilínea y cualquier línea que comience con un carácter sin espacios en blanco cierra el mensaje de log anterior y comienza un nuevo mensaje de log.

  Si incluye este campo, puede especificar `{timestamp_format}` para usar la misma expresión regular que el formato de marca temporal. De lo contrario, se puede especificar otra expresión regular para que CloudWatch Logs la utilice para determinar las líneas iniciales de las entradas de líneas múltiples.
- `encoding`: especifica la codificación del archivo de registro, de modo que se pueda leer correctamente. Si especifica una codificación incorrecta, podría haber pérdida de datos porque los caracteres que no se puedan descodificar se sustituirán por otros.

  El valor predeterminado es `utf-8`. Los valores posibles son los siguientes:

  `ascii, big5, euc-jp, euc-kr, gbk, gb18030, ibm866, iso2022-jp, iso8859-2, iso8859-3, iso8859-4, iso8859-5, iso8859-6, iso8859-7, iso8859-8, iso8859-8-i, iso8859-10, iso8859-13, iso8859-14, iso8859-15, iso8859-16, koi8-r, koi8-u, macintosh, shift_jis, utf-8, utf-16, utf-16le, UTF-16, UTF-16LE, windows-874, windows-1250, windows-1251, windows-1252, windows-1253, windows-1254, windows-1255, windows-1256, windows-1257, windows-1258, x-mac-cyrillic`

- La sección `windows_events` especifica el tipo de eventos de Windows que se recopilan de los servidores con Windows Server. Contiene los campos siguientes:
  - `collect_list`: es obligatorio si `windows_events` está incluido. Especifica los tipos y niveles de eventos de Windows que se recopilarán. Cada log que se recopilará tiene una entrada en esta sección, que puede incluir los siguientes campos:

- `event_name`: especifica el tipo de eventos de Windows que se van a registrar. Esto equivale al nombre del canal del registro de eventos de Windows (como `System`, `Security`, `Application`, etc.). Este campo es obligatorio para cada tipo de evento de Windows que se registrará.

  > **Note**
  >
  > Cuando CloudWatch recupera mensajes de un canal de registro de Windows, busca el canal de registro en función de la propiedad `Full Name`. Mientras tanto, el panel de navegación del visor de eventos de Windows muestra la propiedad `Log Name` de los canales de registro. `Full Name` y `Log Name` no siempre coinciden. Para confirmar el `Full Name` de un canal, haga clic derecho en él en el visor de eventos de Windows y abra Properties (Propiedades).

- `event_levels`: especifica los niveles de evento que se registrarán. Debe especificar cada nivel que se registrará. Entre los valores posibles se incluyen `INFORMATION`, `WARNING`, `ERROR`, `CRITICAL` y `VERBOSE`. Este campo es obligatorio para cada tipo de evento de Windows que se registrará.
- `log_group_name`: obligatorio. Especifica qué se usará como nombre de grupo de registros en CloudWatch Logs.
- `log_stream_name`: opcional. Especifica qué se usará como nombre de flujo de registro en CloudWatch Logs. Como parte del nombre, puede usar `{instance_id}`, `{hostname}` `{local_hostname}` y `{ip_address}` como variables en el nombre. `{hostname}` recupera el nombre de host de los metadatos de EC2, mientras que `{local_hostname}` usa el nombre de host del archivo de configuración de red.

  Si omite este campo, se utilizará el valor del parámetro `log_stream_name` en la sección global `logs`. Si esto también se omite, se utiliza el valor predeterminado de `{instance_id}`.

  Si un flujo de registros no existe todavía, se crea automáticamente.
- `event_format`: Opcional. Especifica el formato que se debe utilizar al almacenar eventos de Windows en CloudWatch Logs. `xml` utiliza el formato XML como en el visor de eventos de Windows. `text` utiliza el formato de agente de CloudWatch Logs heredado.
- `retention_in_days`: opcional. Especifica la cantidad de días que se conservan los eventos de Windows en el grupo de registro especificado.

- Si el agente está creando este grupo de registros y especifica u omite este campo, la retención del nuevo grupo de registro se establecerá de manera que nunca venza.
- Si este grupo de registros ya existe y especifica este campo, se utiliza la nueva retención que especifique. Si omite este campo para un grupo de registros que ya existe, no se modifica la retención del grupo de registros.

  El asistente del agente de CloudWatch utiliza `-1` como valor predeterminado para este campo cuando se utiliza para crear el archivo de configuración del agente y no se especifica un valor para la retención de registros. Esta valor de `-1` especificado por el asistente especifica que los eventos del grupo de registro no caducan. Sin embargo, editar manualmente este valor a `-1` no tiene ningún efecto.

Los valores válidos son 1, 3, 5, 7, 14, 30, 60, 90, 120, 150, 180, 365, 400, 545, 731, 1827, 2192, 2557, 2922, 3288 y 3653.

Si configura el agente para que escriba varios flujos de registro en el mismo grupo de registros, especificar la `retention_in_days` en un solo lugar establecerá la retención de registros para todo el grupo de registros. Si especifica la `retention_in_days` para el mismo grupo de registros en varios lugares, la retención se establece si todos esos valores son iguales. Sin embargo, si se especifican diferentes valores de `retention_in_days` para el mismo grupo de registros en varios lugares, no se establecerá la retención de registros y el agente se detendrá, lo que devolverá un error.

> **Note**
>
> El rol o el usuario de IAM del agente deben tener la `logs:PutRetentionPolicy` para que puedan establecer políticas de retención. Para obtener más información, consulte Permitir que el agente de CloudWatch establezca la política de retención de registros.

> **Warning**
>
> Si establece la `retention_in_days` para un grupo de registros que ya existe, se eliminarán todos los registros de ese grupo de registros que se hayan publicado antes del número de días que especificó. Por ejemplo, si se configurara en 3, se eliminarían todos los registros de hace 3 días y más tiempo.

- `log_stream_name`: obligatorio. Especifica el nombre de secuencia de registros predeterminado que se usará para los registros o los eventos de Windows que no tengan nombres de secuencia de registros individuales definidos en el parámetro `log_stream_name` dentro de su entrada en `collect_list`.
- `endpoint_override`: especifica un punto de enlace de FIPS o un enlace privado que se va a utilizar como punto de enlace donde el agente envía los registros. Al especificar este campo y establecer un enlace privado podrá enviar los registros a un punto de enlace de la Amazon VPC. Para obtener más información, consulte What Is Amazon VPC? (¿Qué es Amazon VPC?).

  El valor de `endpoint_override` debe ser una cadena que sea una URL.

  Por ejemplo, la siguiente parte de la sección de registros del archivo de configuración establece que el agente utilizará un punto de enlace de la VPC al enviar registros.

  ```
  {
    "logs": {
      "endpoint_override": "vpce-XXXXXXXXXXXXXXXXXXXXXXXX.logs.us-
  east-1.vpce.amazonaws.com",
     ......
     },
  }
  ```

- `force_flush_interval`: especifica en segundos la cantidad máxima de tiempo que los registros permanecen en el búfer de memoria antes de enviarse al servidor. Independientemente del valor de este campo, si el tamaño de los registros en el búfer alcanza 1 MB, los registros se envían inmediatamente al servidor. El valor predeterminado es 5.

  Si utiliza el agente para generar informes de métricas de alta resolución en un formato de métrica integrado y está configurando alarmas en esas métricas, mantenga este parámetro establecido en el valor predeterminado de 5. De lo contrario, las métricas se notifican con un retraso que puede provocar alarmas en los datos parciales o incompletos.
- `credentials`: especifica un rol de IAM que se va a utilizar cuando se envían registros a una cuenta de AWS. Si se especifica, este campo contiene un parámetro, `role_arn`.
  - `role_arn`: especifica el ARN de un rol de IAM que se va a utilizar para la autenticación cuando se envían registros a una cuenta de AWS. Para obtener más información, consulte Envío de métricas, registros y seguimientos a una cuenta diferente. Si se especifica aquí, se sobrescribe el valor de `role_arn` especificado en la sección `agent` del archivo de configuración, si existe.

- `metrics_collected`: este campo puede contener secciones para especificar que el agente debe recopilar los registros para habilitar casos de uso como CloudWatch Application Signals e Información de contenedores con una observabilidad mejorada en Amazon EKS.
  - `app_signals`: (opcional) especifica que desea habilitar CloudWatch Application Signals. Para obtener más información sobre esta configuración, consulte Habilitación de CloudWatch Application Signals.
  - `kubernetes`: este campo puede contener un parámetro `enhanced_container_insights` que puede utilizar para habilitar Información de contenedores con una observabilidad mejorada en Amazon EKS.
    - `enhanced_container_insights`: establézcalo en `true` para habilitar Información de contenedores con una observabilidad mejorada en Amazon EKS. Para obtener más información, consulte Información de contenedores con observabilidad mejorada para Amazon EKS.
    - `accelerated_compute_metrics`: establezca el valor como `false` para dejar de recopilar métricas de GPU de Nvidia en los clústeres de Amazon EKS. Para obtener más información, consulte Métricas de GPU de NVIDIA.
  - `emf`: para recopilar las métricas integradas en los registros, ya no es necesario añadir este campo `emf`. Este es un campo legal que especifica que el agente debe recopilar los registros que están en formato de métrica integradas. Puede generar datos de métricas a partir de estos registros. Para obtener más información, consulte Incrustar métricas en los registros.

A continuación se muestra un ejemplo de una sección `logs`.

```
"logs":
   {
       "logs_collected": {
           "files": {
               "collect_list": [
                   {
                       "file_path": "c:\\ProgramData\\Amazon\\AmazonCloudWatchAgent\
\Logs\\amazon-cloudwatch-agent.log",
                       "log_group_name": "amazon-cloudwatch-agent.log",
                       "log_stream_name": "my_log_stream_name_1",
                       "timestamp_format": "%H: %M: %S%y%b%-d"
                   },
                   {
                       "file_path": "c:\\ProgramData\\Amazon\\AmazonCloudWatchAgent\
\Logs\\test.log",
```

```
                    "log_group_name": "test.log",
                    "log_stream_name": "my_log_stream_name_2"
                }
            ]
        },
        "windows_events": {
            "collect_list": [
                {
                    "event_name": "System",
                    "event_levels": [
                        "INFORMATION",
                        "ERROR"
                    ],
                    "log_group_name": "System",
                    "log_stream_name": "System"
                },
                {
                    "event_name": "CustomizedName",
                    "event_levels": [
                        "INFORMATION",
                        "ERROR"
                    ],
                    "log_group_name": "CustomizedLogGroup",
                    "log_stream_name": "CustomizedLogStream"
                }
            ]
        }
     },
     "log_stream_name": "my_log_stream_name",
     "metrics_collected": {
         "kubernetes": {
              "enhanced_container_insights": true
    }
  }
}
```

Archivo de configuración del agente de CloudWatch: sección de seguimientos

Al añadir una sección `traces` al archivo de configuración del agente de CloudWatch, se puede utilizar CloudWatch Application Signals o recopilar seguimientos desde X-Ray y el SDK de instrumentación de OpenTelemetry y se envían a X-Ray.

> **Important**
>
> El rol de IAM o usuario de IAM del agente debe tener la política AWSXrayWriteOnlyAccess para enviar datos de seguimiento a X-Ray. Para obtener más información, consulte Cree roles de IAM y usuarios para utilizarlos con el agente de CloudWatch.

Para empezar rápidamente a recopilar seguimientos, puede añadir lo siguiente al archivo de configuración del agente de CloudWatch.

```
"traces_collected": {
        "xray": {
        },
        "otlp": {
        }
      }
```

Si añade la sección anterior al archivo de configuración del agente de CloudWatch y reinicia el agente, el agente empezará a recopilar seguimientos con las siguientes opciones y valores predeterminados. Para obtener más información sobre estos parámetros, consulte las definiciones de los parámetros más adelante en esta sección.

```
"traces_collected": {
        "xray": {
            "bind_address": "127.0.0.1:2000",
            "tcp_proxy": {
              "bind_address": "127.0.0.1:2000"
            }
        },
        "otlp": {
            "grpc_endpoint": "127.0.0.1:4317",
            "http_endpoint": "127.0.0.1:4318"
        }
      }
```

La sección `traces` puede incluir los siguientes campos:

- `traces_collected`: obligatorio si la sección `traces` está incluida. Especifica los SDK de los que se recopilarán los seguimientos. Puede incluir los siguientes campos:

- `app_signals`: opcional. Especifica que desea habilitar CloudWatch Application Signals. Para obtener más información sobre esta configuración, consulte Habilitación de CloudWatch Application Signals.
- `xray`: opcional. Especifica que desea recopilar seguimientos del SDK de X-Ray. Esta sección puede incluir los siguientes campos.
  - `bind_address`: opcional. Especifica la dirección UDP que el agente de CloudWatch debe usar para escuchar los seguimientos de X-Ray. El formato es `ip:port`. Esta dirección debe coincidir con la que se establece en el X-Ray.

    Si omite este campo, se usa el valor predeterminado de `127.0.0.1:2000`.
  - `tcp_proxy`: opcional. Configura la dirección de un proxy que se utiliza para admitir el muestreo remoto de X-Ray. Para obtener más información, consulte Configuración de reglas de muestreo en la documentación de X-Ray.

    Esta sección puede contener el siguiente campo.
    - `bind_address`: opcional. Especifica la dirección TCP en la que el agente de CloudWatch debe establecer el proxy. El formato es `ip:port`. Esta dirección debe coincidir con la que se establece en el X-Ray.

      Si omite este campo, se usa el valor predeterminado de `127.0.0.1:2000`.
- `otlp`: opcional. Especifica que desea recopilar seguimientos del SDK de OpenTelemetry. Para obtener más información acerca de AWS Distro para OpenTelemetry, consulte AWS Distro for OpenTelemetry. Para obtener más información sobre los SDK de AWS Distro para OpenTelemetry, consulte la Introducción.

  Esta sección puede incluir los siguientes campos.
  - `grpc_endpoint`: opcional. Especifica la dirección que debe utilizar el agente de CloudWatch para escuchar los seguimientos de OpenTelemetry enviados mediante llamadas a procedimientos remotos de gRPC. El formato es `ip:port`. Esta dirección debe coincidir con la dirección establecida para el exportador de gRPC en el SDK de OpenTelemetry.

    Si omite este campo, se usa el valor predeterminado de `127.0.0.1:4317`.
  - `http_endpoint`: opcional. Especifica la dirección que el agente CloudWatch debe utilizar para escuchar los seguimientos OTLP enviadas a través de HTTP. El formato es `ip:port`. Esta dirección debe coincidir con la dirección establecida para el exportador HTTP en el SDK de OpenTelemetry.

Si omite este campo, se usa el valor predeterminado de `127.0.0.1:4318`.

- `concurrency`: opcional. Especifica el número máximo de llamadas simultáneas a X-Ray que se pueden utilizar para cargar seguimientos. El valor predeterminado es 8
- `local_mode`: opcional. Si `true`, el agente no recopila los metadatos de la instancia de Amazon EC2. El valor predeterminado es `false`
- `endpoint_override`: opcional. Especifica un punto de conexión FIPS o un vínculo privado que se va a utilizar como punto de conexión donde el agente envía seguimientos. Al especificar este campo y establecer un enlace privado podrá enviar los registros a un punto de conexión de VPC de Amazon. Para obtener más información, consulte ¿Qué es Amazon VPC

  El valor de `endpoint_override` debe ser una cadena que sea una URL.
- `region_override`: opcional. Especifica la región que se debe utilizar para el punto de conexión de X-Ray. El agente de CloudWatch envía los seguimientos a X-Ray en la región especificada. Si omite este campo, el agente envía los seguimientos a la región donde se encuentra la instancia de Amazon EC2.

  Si especifica una región aquí, esta tendrá prioridad sobre la configuración del parámetro `region` en la sección `agent` del archivo de configuración.
- `proxy_override`: opcional. Especifica la dirección del servidor proxy que utilizará el agente de CloudWatch al enviar solicitudes a X-Ray. El protocolo del servidor proxy debe especificarse como parte de esta dirección.
- `credentials`: especifica un rol de IAM que se va a utilizar cuando se envían seguimientos a una cuenta de AWS. Si se especifica, este campo contiene un parámetro, `role_arn`.
  - `role_arn`: especifica el ARN de un rol de IAM que se va a utilizar para la autenticación cuando se envían seguimientos a una cuenta de AWS. Para obtener más información, consulte Envío de métricas, registros y seguimientos a una cuenta diferente. Si se especifica aquí, se sobrescribe el valor de `role_arn` especificado en la sección `agent` del archivo de configuración, si existe.

Archivo de configuración del agente de CloudWatch: ejemplos completos

A continuación se ofrece un ejemplo de un archivo de configuración completo del agente de CloudWatch de un servidor Linux.

Los elementos indicados en las secciones `measurement` para las métricas que desea recopilar pueden especificar el nombre completo de la métrica o simplemente la parte del nombre de la

métrica que se añadirá al tipo de recurso. Por ejemplo, si especifica `reads` o `diskio_reads` en la sección `measurement` de la sección `diskio`, se recopilará la métrica `diskio_reads`.

En este ejemplo, se incluyen las dos formas de especificar métricas en la sección `measurement`.

```
   {
     "agent": {
       "metrics_collection_interval": 10,
       "logfile": "/opt/aws/amazon-cloudwatch-agent/logs/amazon-cloudwatch-agent.log"
     },
     "metrics": {
       "namespace": "MyCustomNamespace",
       "metrics_collected": {
         "cpu": {
           "resources": [
             "*"
           ],
           "measurement": [
             {"name": "cpu_usage_idle", "rename": "CPU_USAGE_IDLE", "unit":
"Percent"},
             {"name": "cpu_usage_nice", "unit": "Percent"},
             "cpu_usage_guest"
           ],
           "totalcpu": false,
           "metrics_collection_interval": 10,
           "append_dimensions": {
             "customized_dimension_key_1": "customized_dimension_value_1",
             "customized_dimension_key_2": "customized_dimension_value_2"
           }
         },
         "disk": {
           "resources": [
             "/",
             "/tmp"
           ],
           "measurement": [
             {"name": "free", "rename": "DISK_FREE", "unit": "Gigabytes"},
             "total",
             "used"
           ],
            "ignore_file_system_types": [
             "sysfs", "devtmpfs"
           ],
           "metrics_collection_interval": 60,
```

```
            "append_dimensions": {
              "customized_dimension_key_3": "customized_dimension_value_3",
              "customized_dimension_key_4": "customized_dimension_value_4"
            }
          },
          "diskio": {
            "resources": [
              "*"
            ],
            "measurement": [
              "reads",
              "writes",
              "read_time",
              "write_time",
              "io_time"
            ],
            "metrics_collection_interval": 60
          },
          "swap": {
            "measurement": [
              "swap_used",
              "swap_free",
              "swap_used_percent"
            ]
          },
          "mem": {
            "measurement": [
              "mem_used",
              "mem_cached",
              "mem_total"
            ],
            "metrics_collection_interval": 1
          },
          "net": {
            "resources": [
              "eth0"
            ],
            "measurement": [
              "bytes_sent",
              "bytes_recv",
              "drop_in",
              "drop_out"
            ]
          },
```

```
          "netstat": {
            "measurement": [
              "tcp_established",
              "tcp_syn_sent",
              "tcp_close"
            ],
            "metrics_collection_interval": 60
          },
          "processes": {
            "measurement": [
              "running",
              "sleeping",
              "dead"
            ]
          }
        },
        "append_dimensions": {
          "ImageId": "${aws:ImageId}",
          "InstanceId": "${aws:InstanceId}",
          "InstanceType": "${aws:InstanceType}",
          "AutoScalingGroupName": "${aws:AutoScalingGroupName}"
        },
        "aggregation_dimensions" : [["ImageId"], ["InstanceId", "InstanceType"],
 ["d1"],[]],
        "force_flush_interval" : 30
      },
      "logs": {
        "logs_collected": {
          "files": {
            "collect_list": [
              {
                "file_path": "/opt/aws/amazon-cloudwatch-agent/logs/amazon-cloudwatch-
agent.log",
                "log_group_name": "amazon-cloudwatch-agent.log",
                "log_stream_name": "amazon-cloudwatch-agent.log",
                "timezone": "UTC"
              },
              {
                "file_path": "/opt/aws/amazon-cloudwatch-agent/logs/test.log",
                "log_group_name": "test.log",
                "log_stream_name": "test.log",
                "timezone": "Local"
              }
            ]
```

```
          }
        },
        "log_stream_name": "my_log_stream_name",
        "force_flush_interval" : 15,
        "metrics_collected": {
           "kubernetes": {
                "enhanced_container_insights": true
      }
    }
  }
}
```

A continuación se ofrece un ejemplo de un archivo de configuración completo del agente de CloudWatch de un servidor con Windows Server.

```
{
      "agent": {
        "metrics_collection_interval": 60,
        "logfile": "c:\\ProgramData\\Amazon\\AmazonCloudWatchAgent\\Logs\\amazon-
cloudwatch-agent.log"
      },
      "metrics": {
        "namespace": "MyCustomNamespace",
        "metrics_collected": {
          "Processor": {
            "measurement": [
              {"name": "% Idle Time", "rename": "CPU_IDLE", "unit": "Percent"},
              "% Interrupt Time",
              "% User Time",
              "% Processor Time"
            ],
            "resources": [
              "*"
            ],
            "append_dimensions": {
              "customized_dimension_key_1": "customized_dimension_value_1",
              "customized_dimension_key_2": "customized_dimension_value_2"
            }
          },
          "LogicalDisk": {
            "measurement": [
              {"name": "% Idle Time", "unit": "Percent"},
              {"name": "% Disk Read Time", "rename": "DISK_READ"},
```

```
            "% Disk Write Time"
          ],
          "resources": [
            "*"
          ]
        },
        "customizedObjectName": {
          "metrics_collection_interval": 60,
          "customizedCounterName": [
            "metric1",
            "metric2"
          ],
          "resources": [
            "customizedInstances"
          ]
        },
        "Memory": {
          "metrics_collection_interval": 5,
          "measurement": [
            "Available Bytes",
            "Cache Faults/sec",
            "Page Faults/sec",
            "Pages/sec"
          ]
        },
        "Network Interface": {
          "metrics_collection_interval": 5,
          "measurement": [
            "Bytes Received/sec",
            "Bytes Sent/sec",
            "Packets Received/sec",
            "Packets Sent/sec"
          ],
          "resources": [
            "*"
          ],
          "append_dimensions": {
            "customized_dimension_key_3": "customized_dimension_value_3"
          }
        },
        "System": {
          "measurement": [
            "Context Switches/sec",
            "System Calls/sec",
```

```
              "Processor Queue Length"
            ]
          }
        },
        "append_dimensions": {
          "ImageId": "${aws:ImageId}",
          "InstanceId": "${aws:InstanceId}",
          "InstanceType": "${aws:InstanceType}",
          "AutoScalingGroupName": "${aws:AutoScalingGroupName}"
        },
        "aggregation_dimensions" : [["ImageId"], ["InstanceId", "InstanceType"],
 ["d1"],[]]
      },
      "logs": {
        "logs_collected": {
          "files": {
            "collect_list": [
              {
                "file_path": "c:\\ProgramData\\Amazon\\AmazonCloudWatchAgent\\Logs\
\amazon-cloudwatch-agent.log",
                "log_group_name": "amazon-cloudwatch-agent.log",
                "timezone": "UTC"
              },
              {
                "file_path": "c:\\ProgramData\\Amazon\\AmazonCloudWatchAgent\\Logs\
\test.log",
                "log_group_name": "test.log",
                "timezone": "Local"
              }
            ]
          },
          "windows_events": {
            "collect_list": [
              {
                "event_name": "System",
                "event_levels": [
                  "INFORMATION",
                  "ERROR"
                ],
                "log_group_name": "System",
                "log_stream_name": "System",
                "event_format": "xml"
              },
              {
```

```
                "event_name": "CustomizedName",
                "event_levels": [
                  "WARNING",
                  "ERROR"
                ],
                "log_group_name": "CustomizedLogGroup",
                "log_stream_name": "CustomizedLogStream",
                "event_format": "xml"
              }
            ]
          }
        },
        "log_stream_name": "example_log_stream_name"
      }
    }
```

Guarde al archivo de configuración del agente de CloudWatch manualmente

Si se crea o se edita el archivo de configuración del agente de CloudWatch manualmente, se le puede asignar cualquier nombre. Para simplificar la solución de problemas, le recomendamos que asigne el nombre `/opt/aws/amazon-cloudwatch-agent/etc/amazon-cloudwatch-agent.json` en un servidor Linux y `$Env:ProgramData\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent.json` en los servidores con Windows Server. Después de crear el archivo, puede copiarlo en otros servidores en los que desee ejecutar el agente.

Carga del archivo de configuración del agente de CloudWatch al almacén de parámetros de Systems Manager

Si se planea utilizar SSM Agent para instalar el agente de CloudWatch en los servidores, después de editar manualmente el archivo de configuración del agente de CloudWatch, se puede cargar en el almacén de parámetros de Systems Manager. Para ello, utilice el comando `put-parameter` de Systems Manager.

Para poder almacenar el archivo en el almacén de parámetros, debe utilizar un rol de IAM con los permisos suficientes. Para obtener más información, consulte Cree roles y usuarios de IAM para usarlos con el agente de CloudWatch.

Utilice el siguiente comando, donde *`nombre del parámetro`* es el nombre que se usará para este archivo en el almacén de parámetros y *`configuration_file_pathname`* es la ruta y el nombre del archivo de configuración que ha editado.

```
aws ssm put-parameter --name "parameter name" --type "String" --value
 file://configuration_file_pathname
```

Habilitación de CloudWatch Application Signals

Utilice CloudWatch Application Signals para instrumentar de forma automática sus aplicaciones en AWS para que pueda realizar un seguimiento del rendimiento de las aplicaciones en relación con sus objetivos empresariales. Application Signals le proporciona una vista unificada y centrada en las aplicaciones de sus aplicaciones Java, sus dependencias y sus periferias. Para obtener más información, consulte Application Signals.

CloudWatch Application Signals utiliza el agente de CloudWatch para recibir métricas y seguimientos de sus aplicaciones autoinstrumentadas, aplicar reglas de manera opcional para reducir la alta cardinalidad y, a continuación, publicar la telemetría procesada en CloudWatch. Puede proporcionar una configuración personalizada al agente de CloudWatch específicamente para Application Signals mediante el archivo de configuración del agente. Para empezar, la presencia de una sección `app_signals` en la sección `metrics_collected` dentro de la sección `logs` del archivo de configuración del agente especifica que el agente de CloudWatch recibirá las métricas de las aplicaciones autoinstrumentadas. Del mismo modo, la presencia de una sección `app_signals` en la sección `traces_collected` dentro de la sección `traces` del archivo de configuración del agente especifica que el agente de CloudWatch está habilitado para recibir los seguimientos de las aplicaciones autoinstrumentadas. Además, si lo desea, puede introducir reglas de configuración personalizadas para reducir la publicación de telemetría de alta cardinalidad, tal como se describe en esta sección.

- En el caso de los clústeres de Amazon EKS, al instalar el complemento de observabilidad de EKS de Amazon CloudWatch, el agente de CloudWatch está habilitado de forma predeterminada para recibir métricas y seguimientos de sus aplicaciones autoinstrumentadas. Si desea transferir de forma opcional las reglas de configuración personalizadas, puede hacerlo pasando una configuración de agente personalizada al complemento Amazon EKS al crearla o actualizarla mediante una configuración adicional, tal y como se describe en Configuraciones adicionales (Opcional).
- Para otras plataformas compatibles, incluida Amazon EC2, debe iniciar el agente de CloudWatch con una configuración de agente que habilite Application Signals especificando las secciones `app_signals` y, opcionalmente, cualquier regla de configuración personalizada, tal como se describe más adelante en esta sección.

A continuación, se ofrece un resumen de los campos en el archivo de configuración del agente de CloudWatch que se relacionan con CloudWatch Application Signals.

- `logs`
  - `metrics_collected`: este campo puede contener secciones para especificar que el agente debe recopilar los registros para habilitar casos de uso como CloudWatch Application Signals e Información de contenedores con una observabilidad mejorada en Amazon EKS.

> **Note**
>
> Anteriormente, esta sección también se usaba para especificar que el agente debe recopilar registros que están en formato de métrica integradas. Estos ajustes ya no son necesarios.

    - `app_signals`: (opcional) especifica que desea habilitar CloudWatch Application Signals para recibir métricas de sus aplicaciones autoinstrumentadas para facilitar CloudWatch Application Signals.
      - `rules`: (opcional) una serie de reglas para seleccionar métricas y seguimientos de forma condicional y aplicar acciones para administrar escenarios de alta cardinalidad. Cada regla puede incluir los siguientes campos:
        - `rule_name`: (opcional) el nombre de la regla.
        - `selectors`: (opcional) un conjunto de métricas y seguimientos comparadores de dimensiones. Cada selector debe proporcionar los siguientes campos:
          - `dimension`: obligatorio si `selectors` no está vacío. Especifica la dimensión de las métricas y los seguimientos que se van a utilizar como filtro.
          - `match`: obligatorio si `selectors` no está vacío. Un patrón comodín que se utiliza para hacer coincidir valores de la dimensión especificada.
        - `action`: (opcional) la acción que se aplicará a las métricas y los seguimientos que coincidan con los selectores especificados. El valor de `action` tiene que ser una de las siguientes palabras clave:
          - `keep`: especifica que solo se envíen las métricas y los seguimientos a CloudWatch si coinciden con los `selectors`.
          - `drop`: especifica que se eliminen las métricas y los seguimientos que coincidan con los `selectors`.

- `replace`: especifica que se sustituyan las dimensiones de las métricas y los seguimientos que coincidan con los `selectors`. Se sustituyen según la sección `replacements`.

- `replacements` Obligatorio si `action` es `replace`. Una variedad de pares de dimensiones y valores que se aplicarán a las métricas y seguimientos que coincidan con los `selectors` especificados cuando la `action` es `replace`. Cada reemplazo, debe proporcionar los siguientes campos:

  - `target_dimension`: obligatorio si `replacements` no está vacío. Especifica la dimensión que se debe reemplazar.

  - `value`: obligatorio si `replacements` no está vacío. El valor por el que se va a reemplazar el valor original de `target_dimension`.

- `limiter` (Opcional) Utilice esta sección para limitar el número de métricas y dimensiones que Application Signals envía a CloudWatch, a fin de optimizar sus costos.

  - `disabled` (Opcional) Si es `true`, la característica de limitación de métricas está deshabilitada. El valor predeterminado es `false`

  - `drop_threshold` (Opcional) El número máximo de métricas distintas por servicio en un intervalo de rotación que un agente de CloudWatch puede exportar. El valor predeterminado es 500.

  - `rotation_interval` (Opcional) El intervalo en el que el limitador restablece los registros de métricas para el recuento de distinciones. Se expresa como una cadena con una secuencia de números y un sufijo de unidad. Se admiten fracciones. Los sufijos de unidad admitidos son `s`, `m`, `h`, `ms`, `us` y `ns`

    El valor predeterminado es `1h` por una hora.

  - `log_dropped_metrics` (Opcional) Especifica si el agente debe escribir registros en los registros del agente de CloudWatch cuando se eliminan las métricas de Application Signals. El valor predeterminado es `false`.

    > **Note**
    >
    > Para activar este registro, el parámetro `debug` de la sección `agent` también debe estar establecido en `true`.

- `traces`

  - `traces_collected`

- `app_signals`: opcional. Especifique esto para permitir que el agente de CloudWatch reciba los seguimientos de sus aplicaciones autoinstrumentadas para facilitar CloudWatch Application Signals.

> **Note**
>
> Si bien las reglas `app_signals` personalizadas se especifican en la sección `metrics_collected`, incluida en la sección `logs`, también se aplican implícitamente a la sección `traces_collected`. Se aplicará el mismo conjunto de reglas tanto a las métricas como a los seguimientos.

Cuando hay varias reglas con acciones diferentes, se aplican en la siguiente secuencia: `keep`, luego `drop` y después `replace`.

A continuación, se muestra un ejemplo de un archivo de configuración completo del agente de CloudWatch que aplica reglas personalizadas.

```
{
  "logs": {
    "metrics_collected": {
      "app_signals": {
        "rules": [
          {
            "rule_name": "keep01",
            "selectors": [
              {
                "dimension": "Service",
                "match": "pet-clinic-frontend"
              },
              {
                "dimension": "RemoteService",
                "match": "customers-service"
              }
            ],
            "action": "keep"
          },
          {
            "rule_name": "drop01",
            "selectors": [
```

```
              {
                "dimension": "Operation",
                "match": "GET /api/customer/owners/*"
              }
            ],
            "action": "drop"
          },
          {
            "rule_name": "replace01",
            "selectors": [
              {
                "dimension": "Operation",
                "match": "PUT /api/customer/owners/*/pets/*"
              },
              {
                "dimension": "RemoteOperation",
                "match": "PUT /owners"
              }
            ],
            "replacements": [
              {
                "target_dimension": "Operation",
                "value": "PUT /api/customer/owners/{ownerId}/pets{petId}"
              }
            ],
            "action": "replace"
          }
        ]
      }
    }
  },
  "traces": {
    "traces_collected": {
      "app_signals": {}
    }
  }
}
```

En el caso del archivo de configuración del ejemplo anterior, las `rules` se procesan de la siguiente manera:

1. La regla `keep01` asegura que se mantengan todas las métricas y seguimientos con la dimensión `Service` como `pet-clinic-frontend` y la dimensión `RemoteService` como `customers-service`.
2. En el caso de las métricas y los seguimientos procesados una vez que se aplique `keep01`, la regla `drop01` asegura que se eliminen las métricas y los seguimientos con la dimensión `Operation` como `GET /api/customer/owners/*`.
3. En el caso de las métricas y los seguimientos procesados una vez que se aplique `drop01`, la regla `replace01` actualiza las métricas y los seguimientos que tienen la dimensión `Operation` como `PUT /api/customer/owners/*/pets/*` y la dimensión `RemoteOperation` como `PUT /owners`, de tal manera que la dimensión `Operation` se reemplace para ser `PUT /api/customer/owners/{ownerId}/pets{petId}`.

El siguiente es un ejemplo completo de un archivo de configuración de CloudWatch que administra la cardinalidad en Application Signals cambiando el límite de métricas a 100, habilitando el registro de las métricas descartadas y estableciendo el intervalo de rotación en dos horas.

```
{
    "logs": {
        "metrics_collected": {
            "app_signals": {
                "limiter": {
                    "disabled": false,
                    "drop_threshold": 100,
                    "rotation_interval": "2h",
                    "log_dropped_metrics": true
                }
            }
        },
        "traces": {
            "traces_collected": {
                "app_signals": {}
            }
        }
    }
}
```

Recopilación de las métricas de rendimiento de la red

Las instancias EC2 que se ejecutan en Linux y utilizan Elastic Network Adapter (ENA) publican métricas de rendimiento de la red. La versión 1.246396.0 y las posteriores del agente de CloudWatch le permiten importar estas métricas de rendimiento de red a CloudWatch. Al importar estas métricas de rendimiento de red en CloudWatch se cargan como métricas personalizadas de CloudWatch.

Para obtener más información acerca del controlador ENA, consulte Enabling enhanced networking with the Elastic Network Adapter (ENA) on Linux instances (Habilitación de las redes mejoradas con Elastic Network Adapter (ENA) en las instancias Linux) y Enabling enhanced networking with the Elastic Network Adapter (ENA) on Windows instances (Habilitación de las redes mejoradas con Elastic Network Adapter (ENA) en las instancias de Windows).

La forma en que configura la colección de métricas de rendimiento de red difiere en los servidores Linux y Windows.

En la siguiente tabla se enumeran estas métricas de rendimiento de red que el adaptador ENA habilita. Cuando el agente de CloudWatch importa estas métricas en CloudWatch desde instancias de Linux, antepone `ethtool_` al principio de cada uno de estos nombres de métricas.

Métrica	Descripción
Nombre en los servidores Linux: **`bw_in_allowance_exceeded`** Nombre en los servidores Windows: **`Aggregate inbound BW allowance exceeded`**	El número de paquetes en cola o eliminados debido a que la banda ancha de bajada agregada superó el máximo de la instancia. Esta métrica sólo se recopila si la ha incluido en la subsección `ethtool` de la sección `metrics_collected` del archivo de configuración del agente de CloudWatch Para obtener más información, consulte Recopilación de las métricas de rendimiento de la red. Unidad: ninguna
Nombre en los servidores Linux: **`bw_out_allowance_exceeded`**	El número de paquetes en cola o eliminados debido a que la banda ancha de subida agregada superó el máximo de la instancia.

Métrica	Descripción
Nombre en los servidores Windows: **`Aggregate outbound BW allowance exceeded`**	Esta métrica sólo se recopila si se la ha incluido en la subsección `ethtool` de la sección `metrics_collected` del archivo de configuración del agente de CloudWatch Para obtener más información, consulte Recopilación de las métricas de rendimiento de la red. Unidad: ninguna
Nombre en los servidores Linux: **`conntrack_allowance_available`** Nombre en los servidores Windows: **`Available connection tracking allowance`**	Informa de la cantidad de conexiones rastreadas que puede establecer la instancia antes de alcanzar el límite de conexiones rastreadas de ese tipo de instancia. Esta métrica solo está disponible en las instancias de EC2 basadas en Nitro que utilizan el controlador Linux para Elastic Network Adapter (ENA) a partir de la versión 2.8.1 y en aquellas computadoras que utilizan el controlador Windows para Elastic Network Adapter (ENA) a partir de la versión 2.6.0. Esta métrica sólo se recopila si la ha incluido en la subsección `ethtool` de la sección `metrics_collected` del archivo de configuración del agente de CloudWatch Para obtener más información, consulte Recopilación de las métricas de rendimiento de la red. Unidad: ninguna

Métrica	Descripción
Nombre en los servidores Linux: **ena_srd_mode** Nombre en los servidores Windows: **ena srd mode**	Se describen qué características de ENA Express están habilitadas. Para obtener más información sobre ENA Express, consulte Mejorar el rendimiento de la red con ENA Express en instancias de Linux. Los valores son los siguientes: • 0 = ENA Express desactivado, UDP desactivado • 1 = ENA Express activado, UDP desactivado • 2 = ENA Express desactivado, UDP activado **Note** Esto solo ocurre cuando ENA Express se habilitó originalmente y UDP se configuró para usarlo. El valor anterior se retiene para el tráfico UDP. • 3 = ENA Express activado, UDP activado
Nombre en los servidores Linux: **ena_srd_eligible_tx_pkts** Nombre en los servidores Windows: **ena srd eligible tx pkts**	La cantidad de paquetes de red enviados dentro de un periodo determinado que cumplen con los requisitos de elegibilidad de datagramas fiables escalables (SRD) de AWS, como se indica a continuación: • Se admiten los tipos de instancia de envío y recepción. • Tanto las instancias de envío como las de recepción deben tener configurado ENA Express. • Las instancias de envío y recepción deben estar en la misma subred. • La ruta de red entre las instancias no debe incluir cajas de middleware. ENA Express no admite actualmente cajas de middleware.

Métrica	Descripción
Nombre en los servidores Linux: **ena_srd_tx_pkts** Nombre en los servidores Windows: **ena srd tx pkts**	El número de paquetes SRD transmitidos en un periodo determinado.
Nombre en los servidores Linux: **ena_srd_rx_pkts** Nombre en los servidores Windows: **ena srd rx pkts**	El número de paquetes SRD recibidos en un periodo determinado.
Nombre en los servidores Linux: **ena_srd_resource_utilization** Nombre en los servidores Windows: **ena srd resource utilization**	El porcentaje de uso máximo de memoria permitido para conexiones SRD simultáneas que ha consumido la instancia.
Nombre en los servidores Linux: **linklocal_allowance_exceeded** Nombre en los servidores Windows: **Link local packet rate allowance exceeded**	El número de paquetes eliminados porque el PPS del tráfico a los servicios proxy locales superó el máximo para la interfaz de red. Esto afecta al tráfico hacia el servicio de DNS, el servicio de metadatos de instancia y el Servicio de sincronización temporal de Amazon. Esta métrica sólo se recopila si la ha incluido en la subsección `ethtool` de la sección `metrics_collected` del archivo de configuración del agente de CloudWatch Para obtener más información, consulte Recopilación de las métricas de rendimiento de la red. Unidad: ninguna

Métrica	Descripción
Nombre en los servidores Linux: **`linklocal_allowance_exceeded`** Nombre en los servidores Windows: **`Link local packet rate allowance exceeded`**	El número de paquetes eliminados porque el PPS del tráfico a los servicios proxy locales superó el máximo para la interfaz de red. Esto afecta al tráfico hacia el servicio de DNS, el servicio de metadatos de instancia y el Servicio de sincronización temporal de Amazon. Esta métrica sólo se recopila si la ha incluido en la subsección `ethtool` de la sección `metrics_collected` del archivo de configuración del agente de CloudWatch Para obtener más información, consulte Recopilación de las métricas de rendimiento de la red. Unidad: ninguna
Nombre en los servidores Linux: **`pps_allowance_exceeded`** Nombre en los servidores Windows: **`PPS allowance exceeded`**	El número de paquetes en cola o eliminados debido a que la PPS bidireccional superó el máximo de la instancia. Esta métrica sólo se recopila si la ha incluido en la subsección `ethtool` de la sección `metrics_collected` del archivo de configuración del agente de CloudWatch Para obtener más información, consulte Recopilación de las métricas de rendimiento de la red. Unidad: ninguna

Configuración de Linux

En los servidores Linux, el complemento ethtool le permite importar las métricas de rendimiento de red en CloudWatch.

Ethtool es una herramienta estándar de Linux que es capaz de recopilar estadísticas sobre dispositivos Ethernet en servidores Linux. Las estadísticas que recopila dependen del dispositivo

de red y del controlador. Entre los ejemplos de estas estadísticas se incluyen `tx_cnt`, `rx_bytes`, `tx_errors`, y `align_errors`. Cuando se utiliza el complemento ethtool con el agente de CloudWatch, también se pueden importar estas estadísticas en CloudWatch, junto con las métricas de rendimiento de red de EC2 enumeradas anteriormente en esta sección.

> **Tip**
>
> Para encontrar las estadísticas disponibles en nuestro sistema operativo y dispositivo de red, utilice el comando `ethtool -S`.

Cuando el agente de CloudWatch importa métricas en CloudWatch, agrega un prefijo `ethtool_` a los nombres de todas las métricas importadas. Por lo tanto, la estadística estándar de `rx_bytes` de ethtool se denomina `ethtool_rx_bytes` en CloudWatch y la métrica de rendimiento de red de EC2 `bw_in_allowance_exceeded` se denomina `ethtool_bw_in_allowance_exceeded` en CloudWatch.

En los servidores Linux, para importar métricas de ethtool, agregue una sección `ethtool` a `metrics_collected` del archivo de configuración del agente de CloudWatch La sección `ethtool` puede incluir las siguientes subsecciones:

- interface_include: al incluir esta sección hace que el agente recopile métricas sólo de las interfaces que tienen nombres enumerados en esta sección. Si omite esta sección, las métricas se recopilan de todas las interfaces Ethernet que no se enumeran en `interface_exclude`.

  La interfaz Ethernet predeterminada es `eth0`.
- interface_exclude: si incluye esta sección, indique las interfaces Ethernet de las que no desea recopilar métricas.

  El complemento ethtool siempre ignora las interfaces de bucle de retorno.
- metrics_include: en esta sección se enumeran las métricas que se van a importar en CloudWatch. Puede incluir tanto estadísticas estándar que ethtool ha recopilado como también las métricas de red de alta resolución de Amazon EC2.

En el siguiente ejemplo se muestra parte del archivo de configuración del agente de CloudWatch. Esta configuración recopila las métricas estándar de ethtool `rx_packets` y `tx_packets` y las métricas de rendimiento de red de Amazon EC2 de la interfaz `eth1`.

Para obtener más información sobre cómo se crea el archivo de configuración del agente de CloudWatch, consulte Cree o edite de forma manual el archivo de configuración del agente de CloudWatch.

```
"metrics": {
    "append_dimensions": {
      "InstanceId": "${aws:InstanceId}"
    },
    "metrics_collected": {
      "ethtool": {
        "interface_include": [
          "eth1"
        ],
        "metrics_include": [
          "rx_packets",
          "tx_packets",
          "bw_in_allowance_exceeded",
          "bw_out_allowance_exceeded",
          "conntrack_allowance_exceeded",
          "linklocal_allowance_exceeded",
          "pps_allowance_exceeded"
         ]
      }
   }
}
```

Configuración en Windows

En los servidores de Windows, las métricas de rendimiento de la red están disponibles a través de los contadores de rendimiento de Windows, de los que el agente de CloudWatch ya recopila las métricas. Por lo tanto, no necesita un complemento para recopilar estas métricas de los servidores de Windows.

A continuación, tiene una configuración de ejemplo para recopilar las métricas de rendimiento de red de Windows. Para obtener más información sobre cómo se crea el archivo de configuración del agente de CloudWatch, consulte Cree o edite de forma manual el archivo de configuración del agente de CloudWatch.

```
{
    "metrics": {
        "append_dimensions": {
            "InstanceId": "${aws:InstanceId}"
```

```
        },
        "metrics_collected": {
            "ENA Packets Shaping": {
                "measurement": [
                    "Aggregate inbound BW allowance exceeded",
                    "Aggregate outbound BW allowance exceeded",
                    "Connection tracking allowance exceeded",
                    "Link local packet rate allowance exceeded",
                    "PPS allowance exceeded"
                ],
                "metrics_collection_interval": 60,
                "resources": [
                    "*"
                ]
            }
        }
    }
}
```

Métricas de rendimiento de la red

Después de importar métricas de rendimiento de red en CloudWatch, puede ver estas métricas como gráficos de series temporales y crear alarmas que puedan ver estas métricas y notificarle si interrumpen un umbral especificado. El siguiente procedimiento muestra cómo se pueden ver las métricas de ethtool como un gráfico de series temporales. Para obtener más información sobre cómo configurar una alarma, consulte Uso de las alarmas de Amazon CloudWatch.

Debido a que todas estas métricas son contadores agregados, puede usar funciones matemáticas métricas de CloudWatch, como `RATE(METRICS())` para calcular la tasa de estas métricas en gráficos o usarlas para establecer alarmas. Para obtener más información acerca de las funciones de cálculo de métricas, consulte Uso de la calculadora de métricas.

Para ver las métricas de rendimiento de las redes en la consola de CloudWatch

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, seleccione Métricas.
3. Elija el espacio de nombres que se usará para las métricas que el agente ha recopilado. De forma predeterminada, este es CWAgent, pero es posible que haya especificado un espacio de nombres diferente en el archivo de configuración del agente de CloudWatch.
4. Elija una dimensión de métrica (por ejemplo, Per-Instance Metrics [Métricas por instancia]).

5. La pestaña All metrics muestra todas las métricas para dicha dimensión en el espacio de nombres. Puede hacer lo siguiente:

   a. Para representar gráficamente una métrica, active la casilla de verificación situada junto a ella. Para seleccionar todas las métricas, seleccione la casilla de verificación en la fila de encabezado de la tabla.
   b. Para ordenar la tabla, utilice el encabezado de columna.
   c. Para filtrar por recurso, elija el ID de recurso y, a continuación, elija Add to search (Añadir a la búsqueda).
   d. Para filtrar por métrica, elija el nombre de la métrica y, a continuación, elija Add to search (Añadir a la búsqueda).
6. (Opcional) Para agregar el gráfico a un panel de CloudWatch, elija Actions (Acciones) y después Add to dashboard (Añadir al panel).

Recopilación de métricas de GPU NVIDIA

Puede utilizar el agente de CloudWatch para recopilar métricas de GPU NVIDIA de los servidores Linux. Para configurar esta acción, agregue una sección de `nvidia_gpu` en la sección de `metrics_collected` del archivo de configuración del agente de CloudWatch. Para obtener más información, consulte Sección de Linux.

Además, la instancia debe tener instalado un controlador NVIDIA. Los controladores de NVIDIA están preinstalados en algunas Imágenes de máquina de Amazon (AMI). De lo contrario, puede instalar el controlador de forma manual. Para obtener más información, consulte Instalar los controladores NVIDIA en instancias de Linux.

Se pueden recopilar las siguientes métricas. Todas estas métricas se recopilan sin `Unit` de CloudWatch, pero puede especificar una unidad para cada métrica agregando un parámetro al archivo de configuración del agente de CloudWatch. Para obtener más información, consulte Sección de Linux.

Métrica	Nombre de métrica en CloudWatch	Descripción
`utilization_gpu`	`nvidia_smi_utilization_gpu`	Porcentaje de tiempo del último periodo de muestra durante el cual se ejecutaban uno o más núcleos en la GPU.

Métrica	Nombre de métrica en CloudWatch	Descripción
temperature_gpu	nvidia_smi_temperature_gpu	Temperatura del núcleo de la GPU en grados Celsius.
power_draw	nvidia_smi_power_draw	Último consumo de potencia medido para toda la placa en vatios.
utilization_memory	nvidia_smi_utilization_memory	Porcentaje de tiempo del último periodo de muestra durante el cual se leía o escribía la memoria global (dispositivo).
fan_speed	nvidia_smi_fan_speed	Porcentaje de velocidad máxima del ventilador con la cual se espera que funcione el ventilador del dispositivo en este momento.
memory_total	nvidia_smi_memory_total	Memoria total informada en MB.
memory_used	nvidia_smi_memory_used	Memoria utilizada en MB.
memory_free	nvidia_smi_memory_free	Memoria libre en MB.
pcie_link_gen_current	nvidia_smi_pcie_link_gen_current	Generación de enlaces actual.
pcie_link_width_current	nvidia_smi_pcie_link_width_current	Ancho de enlace actual.
encoder_stats_session_count	nvidia_smi_encoder_stats_session_count	Número actual de sesiones de codificador.

Métrica	Nombre de métrica en CloudWatch	Descripción
encoder_stats_average_fps	nvidia_smi_encoder_stats_average_fps	Media móvil de los fotogramas de codificación por segundo.
encoder_stats_average_latency	nvidia_smi_encoder_stats_average_latency	Media móvil de la latencia de codificación en microsegundos.
clocks_current_graphics	nvidia_smi_clocks_current_graphics	Frecuencia actual del reloj de gráficos (sombreador).
clocks_current_sm	nvidia_smi_clocks_current_sm	Frecuencia actual del reloj del multiprocesador de streaming (SM).
clocks_current_memory	nvidia_smi_clocks_current_memory	Frecuencia actual del reloj de memoria.
clocks_current_video	nvidia_smi_clocks_current_video	Frecuencia actual de los relojes de video (codificador más decodificador).

Todas estas métricas se recopilan con las siguientes dimensiones:

Dimensión	Descripción
index	Un identificador único de la GPU en este servidor. Representa el índice de la

Dimensión	Descripción
	Biblioteca de administración de NVIDIA (NVML) del dispositivo.
`name`	Tipo de GPU. Por ejemplo, `NVIDIA Tesla A100`
`host`	Nombre del host del servidor.

Recopilación de las métricas de procesos con el complemento procstat

El complemento procstat permite recopilar las métricas de procesos individuales. Es compatible con los servidores Linux y con servidores con versiones compatibles de Windows Server.

Temas

- Configuración del agente de CloudWatch para procstat
- Métricas que procstat ha recopilado
- Visualización de las métricas de proceso que el agente de CloudWatch ha importado.

Configuración del agente de CloudWatch para procstat

Para utilizar el complemento procstat, agregue una sección `procstat` en la sección `metrics_collected` del archivo de configuración del agente de CloudWatch. Existen tres formas de especificar los procesos para supervisar. Puede utilizar solo uno de estos métodos, pero puede utilizar ese método para especificar uno o varios procesos para supervisar.

- `pid_file`: selecciona los procesos por los nombres de los archivos de número de identificación de proceso (PID) que estos procesos crean.

- `exe`: selecciona los procesos que tienen nombres de proceso que coinciden con la cadena que especifique, utilizando reglas coincidentes con expresiones regulares. La coincidencia es una coincidencia «contains» (contiene), lo que significa que si especifica `agent` como el término con el que coincidir, procesa con nombres como `cloudwatchagent` coincidir con el término. Para obtener más información, consulte Sintaxis.
- `pattern`: selecciona los procesos por las líneas de comandos utilizadas para iniciar los procesos. Se seleccionan todos los procesos que tienen líneas de comandos que coinciden con la cadena especificada con reglas de coincidencia de expresiones regulares. Se comprueba toda la línea de comandos, incluidos parámetros y opciones utilizados con el comando.

  La coincidencia es una coincidencia «contains» (contiene), lo que significa que si especifica `-c` como el término para coincidir, procesa con parámetros como `-config` coincidir con el término.
- `drop_original_metrics`: opcional. Si utiliza el campo `aggregation_dimensions` de la sección `metrics` para agrupar las métricas en resultados agregados, de forma predeterminada, el agente envía tanto las métricas agregadas como las métricas originales, separadas para cada valor de la dimensión. Si no desea que las métricas originales se envíen a CloudWatch, puede especificar este parámetro con una lista de métricas. No se notifican a CloudWatch las métricas especificadas junto con este parámetro por dimensión. En su lugar, solo se registran las métricas agregadas. Esto reduce la cantidad de métricas que recopila el agente, lo que reduce los costes.

El agente CloudWatch utiliza solo uno de estos métodos, incluso si incluye más de una de las secciones anteriores. Si especifica más de una sección, el agente de CloudWatch utiliza la sección `pid_file` en caso de que esté presente. En caso contrario, utiliza la sección `exe`.

En servidores Linux las cadenas que especifique en una sección `pattern` o `exe` se evalúan como expresiones regulares. En servidores que ejecutan Windows Server, estas cadenas se evalúan como consultas de WMI. Un ejemplo sería `pattern: "%apache%"`. Para obtener más información, consulte Operador LIKE.

Independientemente del método que utilice, puede incluir un parámetro `metrics_collection_interval` opcional, que especifique la frecuencia en segundos para recopilar dichas métricas. Si omite este parámetro, se utiliza el valor predeterminado de 60 segundos.

En los ejemplos de las siguientes secciones, la sección `procstat` es la única sección incluida en la sección `metrics_collected` del archivo de configuración del agente. Los archivos de configuración reales también incluyen otras secciones en `metrics_collected`. Para obtener

más información, consulte Cree o edite de forma manual el archivo de configuración del agente de CloudWatch.

Configuración con pid_file

En el siguiente ejemplo, la sección `procstat` supervisa los procesos que crean los archivos PID `example1.pid` y `example2.pid`. De cada proceso se recopilan diferentes métricas. Las métricas recopiladas desde el proceso que crea `example2.pid` se recopilan cada 10 segundos y las métricas recopiladas desde el proceso `example1.pid` se recopilan cada 60 segundos, que es el valor predeterminado.

```
{
    "metrics": {
        "metrics_collected": {
            "procstat": [
                {
                    "pid_file": "/var/run/example1.pid",
                    "measurement": [
                        "cpu_usage",
                        "memory_rss"
                    ]
                },
                {
                    "pid_file": "/var/run/example2.pid",
                    "measurement": [
                        "read_bytes",
                        "read_count",
                        "write_bytes"
                    ],
                    "metrics_collection_interval": 10
                }
            ]
        }
    }
}
```

Configuración con exe

En el siguiente ejemplo, la sección `procstat` supervisa todos los procesos con nombres que coinciden con las cadenas `agent` o `plugin`. De cada proceso se recopilan las mismas métricas.

```
{
```

```
    "metrics": {
        "metrics_collected": {
            "procstat": [
                {
                    "exe": "agent",
                    "measurement": [
                        "cpu_time",
                        "cpu_time_system",
                        "cpu_time_user"
                    ]
                },
                {
                    "exe": "plugin",
                    "measurement": [
                        "cpu_time",
                        "cpu_time_system",
                        "cpu_time_user"
                    ]
                }
            ]
        }
    }
}
```

Configuración con patrón

En el siguiente ejemplo, la sección `procstat` supervisa todos los procesos con líneas de comandos que coinciden con las cadenas `config` o `-c`. De cada proceso se recopilan las mismas métricas.

```
{
    "metrics": {
        "metrics_collected": {
            "procstat": [
                {
                    "pattern": "config",
                    "measurement": [
                        "rlimit_memory_data_hard",
                        "rlimit_memory_data_soft",
                        "rlimit_memory_stack_hard",
                        "rlimit_memory_stack_soft"
                    ]
                },
                {
```

```
                    "pattern": "-c",
                    "measurement": [
                        "rlimit_memory_data_hard",
                        "rlimit_memory_data_soft",
                        "rlimit_memory_stack_hard",
                        "rlimit_memory_stack_soft"
                    ]
                }
            ]
        }
    }
}
```

Métricas que procstat ha recopilado

En la siguiente tabla se muestran las métricas que puede recopilar con el complemento `procstat`.

El agente de CloudWatch agrega `procstat` al principio de los siguientes nombres de métricas. Existe una sintaxis diferente en función de si se ha recopilado de un servidor Linux o un servidor con Windows Server. Por ejemplo, la métrica `cpu_time` aparece como `procstat_cpu_time` cuando la recopila de Linux y como `procstat cpu_time` cuando la recopila de Windows Server.

Nombre de métrica	Disponible en	Descripción
`cpu_time`	Linux	El periodo de tiempo que el proceso utiliza la CPU. Esta métrica se mide en centésimas de segundo. Unidad: recuento
`cpu_time_guest`	Linux	El periodo de tiempo que el proceso está en modo

Nombre de métrica	Disponible en	Descripción
		invitado. Esta métrica se mide en centésimas de segundo. Tipo: flotante Unidad: ninguna
`cpu_time_guest_nice`	Linux	La cantidad de tiempo que el proceso se ejecuta en un invitado niced. Esta métrica se mide en centésimas de segundo. Tipo: flotante Unidad: ninguna

Nombre de métrica	Disponible en	Descripción
`cpu_time_idle`	Linux	El periodo de tiempo que el proceso está en modo inactivo. Esta métrica se mide en centésimas de segundo. Tipo: flotante Unidad: ninguna
`cpu_time_iowait`	Linux	El periodo de tiempo que el proceso está a la espera de que se completen las operaciones de entrada/salida. Esta métrica se mide en centésimas de segundo. Tipo: flotante Unidad: ninguna

Nombre de métrica	Disponible en	Descripción
`cpu_time_irq`	Linux	El periodo de tiempo que el proceso está atendiendo interrupciones. Esta métrica se mide en centésimas de segundo. Tipo: flotante Unidad: ninguna
`cpu_time_nice`	Linux	El periodo de tiempo que el proceso está en modo correcto. Esta métrica se mide en centésimas de segundo. Tipo: flotante Unidad: ninguna

Nombre de métrica	Disponible en	Descripción
`cpu_time_soft_irq`	Linux	La cantidad de tiempo que el proceso está atendiendo interrupciones de software. Esta métrica se mide en centésimas de segundo. Tipo: flotante Unidad: ninguna
`cpu_time_steal`	Linux	La cantidad de tiempo que se dedica a ejecutarse en otros sistemas operativos cuando se ejecuta en un entorno virtualizado. Esta métrica se mide en centésimas de segundo. Tipo: flotante Unidad: ninguna

Nombre de métrica	Disponible en	Descripción
`cpu_time_stolen`	Linux, Windows Server	El periodo de tiempo en que el proceso se encuentra en tiempo descartado, que es el tiempo empleado en otros sistemas operativos en un entorno virtualizado. Esta métrica se mide en centésimas de segundo. Tipo: flotante Unidad: ninguna

Nombre de métrica	Disponible en	Descripción
`cpu_time_system`	Linux, Windows Server y macOS	El periodo de tiempo que el proceso está en modo de sistema. Esta métrica se mide en centésimas de segundo. Tipo: flotante Unidad: recuento
`cpu_time_user`	Linux, Windows Server y macOS	El periodo de tiempo que el proceso está en modo de usuario. Esta métrica se mide en centésimas de segundo. Unidad: recuento

Nombre de métrica	Disponible en	Descripción
`cpu_usage`	Linux, Windows Server y macOS	El porcentaje de tiempo que el proceso está activo en cualquier capacidad. Unidad: porcentaje
`memory_data`	Linux y macOS	La cantidad de memoria que el proceso utiliza para datos. Unidades: bytes
`memory_locked`	Linux y macOS	La cantidad de memoria que el proceso tiene bloqueada. Unidades: bytes
`memory_rss`	Linux, Windows Server y macOS	La cantidad de memoria real (conjunto residente) que está utilizando el proceso. Unidades: bytes

Nombre de métrica	Disponible en	Descripción
`memory_stack`	Linux y macOS	La cantidad de memoria de pila que el proceso está utilizando. Unidades: bytes
`memory_swap`	Linux y macOS	La cantidad de memoria de intercambio que el proceso está utilizando. Unidades: bytes
`memory_vms`	Linux, Windows Server y macOS	La cantidad de memoria virtual que el proceso está utilizando. Unidades: bytes
`num_fds`	Linux	El número de descriptores de archivos que tiene abiertos este proceso. Unidad: ninguna

Nombre de métrica	Disponible en	Descripción
`num_threads`	Linux, Windows, macOS	El número de hilos en este proceso. Unidad: ninguna
`pid`	Linux, Windows Server y macOS	Identificador de proceso (ID). Unidad: ninguna

Nombre de métrica	Disponible en	Descripción
`pid_count`	Linux, Windows Server y macOS	El número de ID de proceso asociado con el proceso. En los servidores Linux y en los equipos macOS, el nombre completo de esta métrica es `procstat_lookup_pid_count` y, en Windows Server, `procstat_lookup pid_count` . Unidad: ninguna
`read_bytes`	Linux, Windows Server	El número de bytes que el proceso ha leído de los discos. Unidades: bytes

Nombre de métrica	Disponible en	Descripción
`write_bytes`	Linux, Windows Server	El número de bytes que el proceso ha escrito en los discos. Unidades: bytes
`read_count`	Linux, Windows Server	El número de operaciones de escritura en disco que el proceso ha ejecutado. Unidad: ninguna
`rlimit_realtime_priority_hard`	Linux	El límite estricto de la prioridad en tiempo real que se puede establecer para este proceso. Unidad: ninguna

Nombre de métrica	Disponible en	Descripción
`rlimit_realtime_priority_soft`	Linux	El límite flexible de la prioridad en tiempo real que se puede establecer para este proceso. Unidad: ninguna
`rlimit_signals_pending_hard`	Linux	El límite estricto del número máximo de señales que puede poner en cola este proceso. Unidad: ninguna
`rlimit_signals_pending_soft`	Linux	El límite flexible del número máximo de señales que puede poner en cola este proceso. Unidad: ninguna

Nombre de métrica	Disponible en	Descripción
`rlimit_nice_priority_hard`	Linux	El límite estricto de la prioridad máxima aceptable que se puede establecer para este proceso. Unidad: ninguna
`rlimit_nice_priority_soft`	Linux	El límite flexible de la prioridad máxima aceptable que se puede establecer para este proceso. Unidad: ninguna
`rlimit_num_fds_hard`	Linux	El límite estricto del número máximo de descriptores de archivos que pueden tener abiertos. Unidad: ninguna

Nombre de métrica	Disponible en	Descripción
`rlimit_num_fds_soft`	Linux	El límite flexible del número máximo de descriptores de archivos que pueden tener abiertos. Unidad: ninguna
`write_count`	Linux, Windows Server	El número de operaciones de escritura en disco que el proceso ha ejecutado. Unidad: ninguna
`involuntary_context_switches`	Linux	El número de veces que el proceso se ha cambiado de contexto involuntariamente. Unidad: ninguna

Nombre de métrica	Disponible en	Descripción
`voluntary_context_switches`	Linux	El número de veces que el proceso se ha cambiado de contexto voluntariamente. Unidad: ninguna
`realtime_priority`	Linux	El uso actual de prioridad en tiempo real para el proceso. Unidad: ninguna
`nice_priority`	Linux	El uso actual de prioridad aceptable para el proceso. Unidad: ninguna

Nombre de métrica	Disponible en	Descripción
`signals_pending`	Linux	El número de señales pendiente de ser gestionadas por el proceso. Unidad: ninguna
`rlimit_cpu_time_hard`	Linux	El límite duro de recursos de tiempo de CPU para el proceso. Unidad: ninguna
`rlimit_cpu_time_soft`	Linux	El límite flexible de recursos de tiempo de CPU para el proceso. Unidad: ninguna
`rlimit_file_locks_hard`	Linux	El límite duro de bloqueos de archivos para el proceso. Unidad: ninguna

Nombre de métrica	Disponible en	Descripción
`rlimit_file_locks_soft`	Linux	El límite flexible de bloqueos de archivos para el proceso. Unidad: ninguna
`rlimit_memory_data_hard`	Linux	El límite duro de recursos en el proceso de memoria utilizada para los datos. Unidades: bytes
`rlimit_memory_data_soft`	Linux	El límite flexible de recursos en el proceso de memoria utilizada para los datos. Unidades: bytes
`rlimit_memory_lock ed_hard`	Linux	El límite duro de recursos en el proceso de memoria bloqueada. Unidades: bytes

Nombre de métrica	Disponible en	Descripción
rlimit_memory_locked_soft	Linux	El límite flexible de recursos en el proceso de memoria bloqueada. Unidades: bytes
rlimit_memory_rss_hard	Linux	El límite duro de recursos en el proceso de memoria física. Unidades: bytes
rlimit_memory_rss_soft	Linux	El límite flexible de recursos en el proceso de memoria física. Unidades: bytes
rlimit_memory_stack_hard	Linux	El límite duro de recursos en la pila de procesos. Unidades: bytes

Nombre de métrica	Disponible en	Descripción
`rlimit_memory_stack_soft`	Linux	El límite flexible de recursos en la pila de procesos. Unidades: bytes
`rlimit_memory_vms_hard`	Linux	El límite duro de recursos en el proceso de memoria virtual. Unidades: bytes
`rlimit_memory_vms_soft`	Linux	El límite flexible de recursos en el proceso de memoria virtual. Unidades: bytes

Visualización de las métricas de proceso que el agente de CloudWatch ha importado.

Después de importar métricas de proceso en CloudWatch, puede ver estas métricas como gráficos de series temporales y crear alarmas que puedan ver estas métricas y notificarle si infringen un umbral especificado. El siguiente procedimiento muestra cómo se pueden ven las métricas de proceso como un gráfico de series temporales. Para obtener más información sobre cómo se configura una alarma, consulte Uso de las alarmas de Amazon CloudWatch.

Para ver las métricas en la consola de CloudWatch

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.

2. En el panel de navegación, seleccione Métricas.
3. Elija el espacio de nombres que se usará para las métricas que el agente ha recopilado. De forma predeterminada, este es CWAgent, pero es posible que haya especificado un espacio de nombres diferente en el archivo de configuración del agente de CloudWatch.
4. Elija una dimensión de métrica (por ejemplo, Per-Instance Metrics [Métricas por instancia]).
5. La pestaña All metrics muestra todas las métricas para dicha dimensión en el espacio de nombres. Puede hacer lo siguiente:

   a. Para representar gráficamente una métrica, active la casilla de verificación situada junto a ella. Para seleccionar todas las métricas, seleccione la casilla de verificación en la fila de encabezado de la tabla.
   b. Para ordenar la tabla, utilice el encabezado de columna.
   c. Para filtrar por recurso, seleccione el ID de recurso y, a continuación, elija Add to search (Añadir a la búsqueda).
   d. Para filtrar por métrica, elija el nombre de la métrica y, a continuación, seleccione Add to search (Añadir a búsqueda).
6. (Opcional) Para agregar el gráfico a un panel de CloudWatch, elija Actions (Acciones) y después Add to dashboard (Añadir al panel).

Recuperación de las métricas personalizadas con StatsD

Puede recuperar métricas personalizadas adicionales de sus aplicaciones o servicios con el agente de CloudWatch con el protocolo `StatsD`. StatsD es una solución popular de código abierto que puede recopilar métricas de una amplia variedad de aplicaciones. StatsD es especialmente útil para instrumentar sus propias métricas. Para obtener un ejemplo de cómo se utilizan el agente de CloudWatch y StatsD conjuntamente, consulte How to better monitor your custom application metrics using Amazon CloudWatch Agent (¿Cómo se supervisan de manera más efectiva las métricas de la aplicación personalizada con el agente de Amazon CloudWatch?).

`StatsD` es compatible con los servidores Linux y los servidores con Windows Server. CloudWatch admite el siguiente formato `StatsD`:

```
MetricName:value|type|@sample_rate|#tag1:
  value,tag1...
```

- `MetricName`: una cadena sin dos puntos, barras, caracteres # o @.

- `value`: puede ser entero o flotante.
- `type`: especifique `c` para contador, `g` para medidor, `ms` para temporizador, `h` para el histograma o `s` para conjunto.
- `sample_rate`: (Opcional) un flotador entre 0 y 1, inclusive. Se utiliza únicamente para las métricas de contador, histograma y temporizador. El valor predeterminado es 1 (muestreo del 100% del tiempo).
- `tags`: (Opcional) una lista de etiquetas separadas por comas. Las etiquetas `StatsD` son similares a las dimensiones de CloudWatch. Utilice el carácter de dos puntos para las etiquetas de clave/valor, como `env:prod`.

Puede utilizar cualquier cliente de `StatsD` que utilice este formato para enviar las métricas al agente de CloudWatch. Para obtener más información acerca de algunos de los clientes de `StatsD` disponibles, consulte la página de clientes de StatsD en GitHub.

Para recopilar estas métricas personalizadas, añada una línea `"statsd": {}` a la sección `metrics_collected` del archivo de configuración del agente. Puede añadir esta línea manualmente. Si utiliza el asistente para crear el archivo de configuración, esta línea se añade automáticamente. Para obtener más información, consulte Cree el archivo de configuración del agente de CloudWatch.

La configuración de `StatsD` predeterminada funciona para la mayoría de los usuarios. Hay campos opcionales que puede añadir a la sección statsd del archivo de configuración del agente según sea necesario:

- `service_address`: la dirección del servicio a la que el agente de CloudWatch debe escuchar . El formato es *`ip`*`:`*`port`*. Si se omite la dirección IP, el agente escucha en todas las interfaces disponibles. Solo se admite el formato UDP, por lo que no es necesario especificar un prefijo UDP.

  El valor predeterminado es `:8125`.
- `metrics_collection_interval`: la frecuencia en segundos con la que se utiliza el complemento `StatsD` se ejecuta y recopila métricas. El valor de predeterminado es de 10 segundos. El rango va de 1 a 172 000.
- `metrics_aggregation_interval`: la frecuencia en segundos con la que CloudWatch agrupa las métricas en puntos de datos únicos. El valor de predeterminado es de 60 segundos.

Por ejemplo, si `metrics_collection_interval` es 10 y `metrics_aggregation_interval` es 60, CloudWatch recopila datos cada 10 segundos. Después de cada minuto, las seis lecturas de datos de ese minuto se agrupan en un único punto de datos, que se envía a CloudWatch.

El rango va de 0 a 172 000. Si `metrics_aggregation_interval` se establece en 0, se deshabilita la agrupación de métricas de `StatsD`.

- `allowed_pending_messages`: el número de mensajes UDP que se pueden poner en cola. Cuando la cola está llena, el servidor StatsD comienza a descartar paquetes. El valor predeterminado es 10 000.
- `drop_original_metrics`: opcional. Si utiliza el campo `aggregation_dimensions` de la sección `metrics` para agrupar las métricas en resultados agregados, de forma predeterminada, el agente envía tanto las métricas agregadas como las métricas originales, separadas para cada valor de la dimensión. Si no desea que las métricas originales se envíen a CloudWatch, puede especificar este parámetro con una lista de métricas. No se notifican a CloudWatch las métricas especificadas junto con este parámetro por dimensión. En su lugar, solo se registran las métricas agregadas. Esto reduce la cantidad de métricas que recopila el agente, lo que reduce los costes.

A continuación se muestra un ejemplo de la sección statsd del archivo de configuración del agente, con el puerto predeterminado e intervalos de recopilación y agrupación personalizados.

```
{
   "metrics":{
      "metrics_collected":{
         "statsd":{
            "service_address":":8125",
            "metrics_collection_interval":60,
            "metrics_aggregation_interval":300
         }
      }
   }
}
```

Visualización de métricas StatsD que el agente de CloudWatch ha importado

Después de importar las métricas StatsD en CloudWatch, puede verlas como gráficos de series temporales y crear alarmas que puedan ver las métricas y notificarle si infringen un umbral que especifique. El siguiente procedimiento muestra cómo ver métricas de StatsD como un gráfico de

series temporales. Para obtener más información sobre cómo se configura una alarma, consulte Uso de las alarmas de Amazon CloudWatch.

Para ver las métricas StatsD en la consola de CloudWatch

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, seleccione Métricas.
3. Elija el espacio de nombres que se usará para las métricas que el agente ha recopilado. De forma predeterminada, este es CWAgent, pero es posible que haya especificado un espacio de nombres diferente en el archivo de configuración del agente de CloudWatch.
4. Elija una dimensión de métrica (por ejemplo, Per-Instance Metrics [Métricas por instancia]).
5. La pestaña All metrics muestra todas las métricas para dicha dimensión en el espacio de nombres. Puede hacer lo siguiente:
   a. Para representar gráficamente una métrica, active la casilla de verificación situada junto a ella. Para seleccionar todas las métricas, seleccione la casilla de verificación en la fila de encabezado de la tabla.
   b. Para ordenar la tabla, utilice el encabezado de columna.
   c. Para filtrar por recurso, seleccione el ID de recurso y, a continuación, elija Add to search (Añadir a la búsqueda).
   d. Para filtrar por métrica, elija el nombre de la métrica y, a continuación, seleccione Add to search (Añadir a búsqueda).
6. (Opcional) Para agregar el gráfico a un panel de CloudWatch, elija Actions (Acciones) y después Add to dashboard (Añadir al panel).

Recuperación de las métricas personalizadas con collectd

Puede recuperar métricas adicionales de las aplicaciones o servicios con el agente de CloudWatch con el protocolo collectd, que solo es compatible con servidores Linux. Collectd es una solución popular de código abierto con complementos que pueden recopilar estadísticas del sistema para una amplia variedad de aplicaciones. Al combinar las métricas del sistema que el agente de CloudWatch ya puede recopilar con las métricas adicionales de collectd, puede supervisar, analizar y solucionar mejor los problemas de los sistemas y las aplicaciones. Para obtener más información acerca de collectd, consulte collectd - El daemon de recopilación de estadísticas del sistema.

El software collectd se utiliza para enviar las métricas al agente de CloudWatch. Para las métricas de collectd, el agente de CloudWatch actúa como el servidor, mientras que el complemento collectd actúa como el cliente.

El software collectd no se instala automáticamente en cada servidor. Siga estos pasos para instalar collectd en un servidor que ejecuta Amazon Linux 2.

```
sudo amazon-linux-extras install collectd
```

Para obtener más información acerca de la instalación de collectd en otros sistemas, vea la página de descarga de collectd.

Para recopilar estas métricas personalizadas, añada una línea “collectd”: {} a la sección metrics_collected del archivo de configuración del agente. Puede añadir esta línea manualmente. Si utiliza el asistente para crear el archivo de configuración, esta línea se añade automáticamente. Para obtener más información, consulte Cree el archivo de configuración del agente de CloudWatch.

También hay disponibles otros parámetros opcionales. Si utiliza collectd y no utiliza `/etc/collectd/auth_file` como su collectd_auth_file, debe configurar algunas de estas opciones.

- service_address: la dirección del servicio en la que el agente de CloudWatch debería escuchar. El formato es "`udp://`*`ip`*`:`*`port`*. El valor predeterminado es `udp://127.0.0.1:25826`.
- name_prefix: un prefijo que se adjunta al principio del nombre de cada métrica de collectd. El valor predeterminado es `collectd_`. La longitud máxima es de 255 caracteres.
- collectd_security_level: establece el nivel de seguridad de la comunicación de red. El valor predeterminado es encrypt (cifrar).

  encrypt (cifrar) especifica que solo se aceptan datos cifrados. sign (firmar) especifica que solo se aceptan datos firmados y cifrados. none (ninguno) especifica que se aceptan todos los datos. Si especifica un valor para collectd_auth_file, se descifran los datos cifrados si es posible.

  Para obtener más información, consulte las secciones sobre configuración de cliente e interacciones posibles en el Wiki de collectd.
- collectd_auth_file Establece un archivo en que los nombres de usuario se asignan a contraseñas. Estas contraseñas se utilizan para verificar firmas y para descifrar los paquetes de red cifrados. Si se facilita, se verifican los datos firmada y se descifran los paquetes cifrados. De lo contrario, se aceptan los datos firmados sin comprobar la firma y los datos cifrados no se pueden descifrar.

  El valor predeterminado es `/etc/collectd/auth_file`.

Si collectd_security_level se establece en none (ninguno), esto es opcional. Si establece collectd_security_level en `encrypt` o sign (firmar), debe especificar collectd_auth_file.

Para el formato del archivo de autenticación, cada línea es un nombre de usuario seguido de dos puntos y cualquier número de espacios seguido de la contraseña. Por ejemplo:

```
user1: user1_password

user2: user2_password
```

- collectd_typesdb: una lista de uno o más archivos que contienen descripciones del conjunto de datos. La lista debe estar incluida entre corchetes, aunque solo haya una entrada en la lista. Cada entrada de la lista deben ir entre comillas dobles. Si hay varias entradas, sepárelas con comas. El valor predeterminado en servidores Linux es `["/usr/share/collectd/types.db"]`. El valor predeterminado en equipos macOS depende de la versión de collectd. Por ejemplo, `["/usr/local/Cellar/collectd/5.12.0/share/collectd/types.db"]`.

  Para obtener más información, consulte https://www.collectd.org/documentation/manpages/types.db.html.
- metrics_aggregation_interval: la frecuencia en segundos con la que CloudWatch agrupa las métricas en puntos de datos únicos. El valor predeterminado es de 60 segundos. El rango va de 0 a 172,000. Si se establece en 0, se deshabilita la agrupación de métricas de collectd.

A continuación, se ofrece un ejemplo de la sección collectd del archivo de configuración del agente.

```
{
   "metrics":{
      "metrics_collected":{
         "collectd":{
            "name_prefix":"My_collectd_metrics_",
            "metrics_aggregation_interval":120
         }
      }
   }
}
```

Visualización de métricas collectd que el agente de CloudWatch ha importado.

Después de importar métricas collectd en CloudWatch, puede ver estas métricas como gráficos de series temporales y crear alarmas que puedan ver estas métricas y notificarle si infringen un umbral

especificado. El siguiente procedimiento muestra cómo ver las métricas collectd como un gráfico de series temporales. Para obtener más información sobre cómo se configuran las alarmas, consulte Uso de las alarmas de Amazon CloudWatch.

Para ver las métricas collectd en la consola de CloudWatch

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, seleccione Métricas.
3. Elija el espacio de nombres que se usará para las métricas que el agente ha recopilado. De forma predeterminada, este es CWAgent, pero es posible que haya especificado un espacio de nombres diferente en el archivo de configuración del agente de CloudWatch.
4. Elija una dimensión de métrica (por ejemplo, Per-Instance Metrics [Métricas por instancia]).
5. La pestaña All metrics muestra todas las métricas para dicha dimensión en el espacio de nombres. Puede hacer lo siguiente:
    a. Para representar gráficamente una métrica, active la casilla de verificación situada junto a ella. Para seleccionar todas las métricas, seleccione la casilla de verificación en la fila de encabezado de la tabla.
    b. Para ordenar la tabla, utilice el encabezado de columna.
    c. Para filtrar por recurso, seleccione el ID de recurso y, a continuación, elija Add to search (Añadir a la búsqueda).
    d. Para filtrar por métrica, elija el nombre de la métrica y, a continuación, seleccione Add to search (Añadir a búsqueda).
6. (Opcional) Para agregar el gráfico a un panel de CloudWatch, elija Actions (Acciones) y después Add to dashboard (Añadir al panel).

Configuración de la recopilación de métricas de Prometheus en instancias de Amazon EC2

En las siguientes secciones se explica cómo se instala el agente CloudWatch con supervisión de Prometheus en instancias EC2 y cómo se configura el agente para que elimine destinos adicionales. También proporciona tutoriales para configurar cargas de trabajo de muestra para usar pruebas con supervisión de Prometheus.

Para obtener información sobre los sistemas operativos compatibles con el agente de CloudWatch, consulte Recopile las métricas, registros y seguimientos con el agente de CloudWatch

Requisitos del grupo de seguridad de la VPC

Si utiliza una VPC, se aplican los siguientes requisitos.

- Las reglas de entrada de los grupos de seguridad para las cargas de trabajo de Prometheus deben abrir los puertos de Prometheus al agente de CloudWatch para extraer las métricas de Prometheus por la IP privada.
- Las reglas de salida del grupo de seguridad para el agente de CloudWatch deben permitir que el agente de CloudWatch se conecte al puerto de cargas de trabajo de Prometheus mediante una IP privada.

Temas

Paso 1: Instale el agente de CloudWatch

El primer paso consiste en instalar el agente de CloudWatch en la instancia EC2. Para obtener instrucciones, consulte Instalación del agente de CloudWatch.

Paso 2: Raspe las fuentes Prometheus e importe las métricas

El agente CloudWatch con supervisión de Prometheus necesita dos configuraciones para raspar las métricas de Prometheus. Una de ellas es para la configuración estándar de Prometheus que se describe en <scrape_config> de la documentación de Prometheus. El otro es para la configuración del agente de CloudWatch.

Configuración de raspado de Prometheus

El agente de CloudWatch es compatible con la configuración de raspado estándar de Prometheus como se describe en <scrape_config> en la documentación de Prometheus. Se puede editar esta sección para actualizar las configuraciones que ya están en este archivo y agregar destinos adicionales de raspado de Prometheus. Un archivo de configuración de ejemplo contiene las siguientes líneas de configuración global:

```
PS C:\ProgramData\Amazon\AmazonCloudWatchAgent> cat prometheus.yaml
```

```
global:
  scrape_interval: 1m
  scrape_timeout: 10s
scrape_configs:
- job_name: MY_JOB
  sample_limit: 10000
  file_sd_configs:
    - files: ["C:\\ProgramData\\Amazon\\AmazonCloudWatchAgent\\prometheus_sd_1.yaml",
 "C:\\ProgramData\\Amazon\\AmazonCloudWatchAgent\\prometheus_sd_2.yaml"]
```

La sección `global` especifica los parámetros que son válidos en todos los contextos de configuración. También sirven como valores predeterminados para otras secciones de configuración. Contiene los siguientes parámetros:

- `scrape_interval`: define la frecuencia con la que se deben raspar los objetivos.
- `scrape_timeout`: define cuánto tiempo debe esperar antes de que se agote el tiempo de espera de una solicitud de raspado.

La sección `scrape_configs` especifica un conjunto de objetivos y parámetros que definen cómo se deben raspar. Contiene los siguientes parámetros:

- `job_name`: el nombre del trabajo que se ha asignado a las métricas raspadas de forma predeterminada.
- `sample_limit`: límite por raspado en el número de muestras raspadas que se aceptarán.
- `file_sd_configs`: lista de configuraciones de descubrimiento de servicios de archivos. Lee un conjunto de archivos que contienen una lista de cero o más configuraciones estáticas. La sección `file_sd_configs` contiene un parámetro `files` que define patrones para los archivos de los que se extraen los grupos de destino.

El agente de CloudWatch admite los siguientes tipos de configuración de descubrimiento de servicios.

`static_config` Permite especificar una lista de destinos y un conjunto de etiquetas comunes para ellos. Es la forma canónica de especificar objetivos estáticos en una configuración de raspado.

A continuación, se muestra una configuración estática de ejemplo para extraer métricas de Prometheus de un host local. Las métricas también se pueden extraer de otros servidores si el puerto Prometheus está abierto al servidor donde se ejecuta el agente.

```
PS C:\ProgramData\Amazon\AmazonCloudWatchAgent> cat prometheus_sd_1.yaml
- targets:
    - 127.0.0.1:9404
  labels:
    key1: value1
    key2: value2
```

Este ejemplo contiene los siguientes parámetros:

- `targets`: los objetivos que la configuración estática ha raspado.
- `labels`: etiquetas asignadas a todas las métricas que se raspan de los destinos.

`ec2_sd_config` Permite recuperar destinos de raspado de instancias de Amazon EC2. A continuación, se muestra un ejemplo `ec2_sd_config` para raspar las métricas de Prometheus de una lista de instancias EC2. Los puertos Prometheus de estas instancias tienen que abrirse al servidor donde se ejecuta el agente de CloudWatch. El rol de IAM para la instancia EC2 en la que se ejecuta el agente de CloudWatch debe incluir el permiso `ec2:DescribeInstance`. Por ejemplo, puede adjuntar la política administrada AmazonEC2ReadOnlyAccess a la instancia que ejecuta el agente de CloudWatch.

```
PS C:\ProgramData\Amazon\AmazonCloudWatchAgent> cat prometheus.yaml
global:
  scrape_interval: 1m
  scrape_timeout: 10s
scrape_configs:
  - job_name: MY_JOB
    sample_limit: 10000
    ec2_sd_configs:
      - region: us-east-1
        port: 9404
        filters:
          - name: instance-id
            values:
              - i-98765432109876543
              - i-12345678901234567
```

Este ejemplo contiene los siguientes parámetros:

- `region`: la región de AWS donde se encuentra la instancia EC2 de destino. Si deja este campo en blanco, se usará la región de los metadatos de instancia.

- `port`: el puerto desde el que se raspan las métricas.
- `filters`: filtros opcionales que se utilizarán para filtrar la lista de instancias. Este ejemplo se filtra en función de los ID de las instancias EC2. Para obtener más criterios por los que se puede filtrar, consulte DescribeInstances.

Configuración del agente de CloudWatch para Prometheus

El archivo de configuración del agente de CloudWatch incluye secciones `prometheus` en `logs` y `metrics_collected`. Incluye los siguientes parámetros.

- cluster_name: especifica el nombre del clúster que se va a agregar como etiqueta en el evento de registro. Este campo es opcional.
- log_group_name: especifica el nombre del grupo de registros de las métricas de Prometheus raspadas.
- prometheus_config_path: especifica la ruta del archivo de configuración de descubrimiento de Prometheus.
- emf_processor: especifica la configuración del procesador de formato de métrica integrada. Para obtener más información sobre el formato de métrica integrado, consulte Incrustar métricas en los registros.

  La sección `emf_processor` contiene los siguientes parámetros:

  - metric_declaration_dedup: se establece en VERDADERO, la función de desduplicación para las métricas de formato de métrica incrustada está habilitada.
  - metric_namespace: especifica el espacio de nombres de la métrica para las métricas emitidas de CloudWatch.
  - metric_unit: especifica el nombre de métrica: mapa de unidades métricas. Para obtener más información acerca de las unidades métricas compatibles, consulte MetricDatum.
  - metric_declaration: son secciones que especifican la matriz de registros con el formato de métrica integrada que se van a generar. Hay secciones `metric_declaration` para cada fuente de Prometheus desde las que el agente de CloudWatch importa de forma predeterminada. Cada una de estas secciones incluye los siguientes campos:
    - `source_labels` especifica el valor de las etiquetas que se comprueban con `label_matcher`.

- `label_matcher` es una expresión regular que verifica el valor de las etiquetas que aparecen en `source_labels`. Las métricas que coinciden se habilitan para incorporarse al formato de métrica integrada que se envía a CloudWatch.
- `metric_selectors` es una expresión regular que especifica las métricas que se van a recopilar y enviar a CloudWatch.
- `dimensions` es la lista de etiquetas que se van a utilizar como dimensiones de CloudWatch en cada métrica seleccionada.

A continuación se muestra un ejemplo de configuración del agente de CloudWatch para Prometheus.

```
{
   "logs":{
      "metrics_collected":{
         "prometheus":{
            "cluster_name":"prometheus-cluster",
            "log_group_name":"Prometheus",
            "prometheus_config_path":"C:\\ProgramData\\Amazon\\AmazonCloudWatchAgent\
\prometheus.yaml",
            "emf_processor":{
               "metric_declaration_dedup":true,
               "metric_namespace":"CWAgent-Prometheus",
               "metric_unit":{
                  "jvm_threads_current": "Count",
                  "jvm_gc_collection_seconds_sum": "Milliseconds"
               },
               "metric_declaration":[
                  {
                     "source_labels":[
                        "job", "key2"
                     ],
                     "label_matcher":"MY_JOB;^value2",
                     "dimensions":[
                        [
                           "key1", "key2"
                        ],
                        [
                           "key2"
                        ]
                     ],
                     "metric_selectors":[
                        "^jvm_threads_current$",
```

```
                        "^jvm_gc_collection_seconds_sum$"
                    ]
                }
            ]
        }
    }
  }
}
```

En el ejemplo anterior se configura una sección de formato de métrica integrada para que se envíe como evento de registro si se cumplen las siguientes condiciones:

- El valor de la etiqueta `job` es `MY_JOB`
- El valor de la etiqueta `key2` es `value2`
- Las métricas de Prometheus `jvm_threads_current` y `jvm_gc_collection_seconds_sum` contienen las etiquetas `job` y `key2`.

El evento de registro que se envía incluye la siguiente sección resaltada:

```
{
    "CloudWatchMetrics": [
        {
            "Metrics": [
                {
                    "Unit": "Count",
                    "Name": "jvm_threads_current"
                },
                {
                    "Unit": "Milliseconds",
                    "Name": "jvm_gc_collection_seconds_sum"
                }
            ],
            "Dimensions": [
                [
                    "key1",
                    "key2"
                ],
                [
                    "key2"
                ]
```

```
            ],
            "Namespace": "CWAgent-Prometheus"
        }
    ],
    "ClusterName": "prometheus-cluster",
    "InstanceId": "i-0e45bd06f196096c8",
    "Timestamp": "1607966368109",
    "Version": "0",
    "host": "EC2AMAZ-PDDOIUM",
    "instance": "127.0.0.1:9404",
    "jvm_threads_current": 2,
    "jvm_gc_collection_seconds_sum": 0.006000000000000002,
    "prom_metric_type": "gauge",
    ...
}
```

Ejemplo: Configure las cargas de trabajo de ejemplo de Java/JMX para las pruebas métricas de Prometheus

JMX Exporter es un exportador oficial de Prometheus que puede extraer y exponer mBeans de JMX como métricas de Prometheus. Para obtener más información, consulte prometheus/jmx_exporter.

El agente de CloudWatch recolecta métricas predefinidas de Prometheus de la máquina virtual de Java (JVM), Hjava y de Tomcat (Catalina) de un JMX Exporter en instancias EC2.

Paso 1: Instale el agente de CloudWatch

El primer paso consiste en instalar el agente de CloudWatch en la instancia EC2. Para obtener instrucciones, consulte Instalación del agente de CloudWatch.

Paso 2: Comience la carga de trabajo de Java/JMX

El siguiente paso es comenzar la carga de trabajo de Java/JMX.

Primero, descargue el archivo jar más reciente de JMX exporter desde la siguiente ubicación: prometheus/jmx_exporter.

Use el jar para la aplicación de muestra

Los comandos de ejemplo de las siguientes secciones utilizan `SampleJavaApplication-1.0-SNAPSHOT.jar` como el archivo jar. Reemplace estas partes de los comandos con el jar para la aplicación.

Prepare la configuración de JMX exporter

El archivo `config.yaml` es el archivo de configuración de JMX Exporter. Para obtener más información, consulte Configuration (Configuración) en la documentación de JMX exporter.

Este es un archivo de configuración de ejemplo para Java y Tomcat.

```
---
lowercaseOutputName: true
lowercaseOutputLabelNames: true

rules:
- pattern: 'java.lang<type=OperatingSystem><>(FreePhysicalMemorySize|
TotalPhysicalMemorySize|FreeSwapSpaceSize|TotalSwapSpaceSize|SystemCpuLoad|
ProcessCpuLoad|OpenFileDescriptorCount|AvailableProcessors)'
  name: java_lang_OperatingSystem_$1
  type: GAUGE

- pattern: 'java.lang<type=Threading><>(TotalStartedThreadCount|ThreadCount)'
  name: java_lang_threading_$1
  type: GAUGE

- pattern: 'Catalina<type=GlobalRequestProcessor, name=\"(\w+-\w+)-(\d+)\"><>(\w+)'
  name: catalina_globalrequestprocessor_$3_total
  labels:
    port: "$2"
    protocol: "$1"
  help: Catalina global $3
  type: COUNTER

- pattern: 'Catalina<j2eeType=Servlet, WebModule=//([-a-zA-Z0-9+&@#/%?=~_|!:.,;]*[-
a-zA-Z0-9+&@#/%=~_|]), name=([-a-zA-Z0-9+/$%~_-|!.]*), J2EEApplication=none,
 J2EEServer=none><>(requestCount|maxTime|processingTime|errorCount)'
  name: catalina_servlet_$3_total
  labels:
    module: "$1"
    servlet: "$2"
  help: Catalina servlet $3 total
  type: COUNTER

- pattern: 'Catalina<type=ThreadPool, name="(\w+-\w+)-(\d+)"><>(currentThreadCount|
currentThreadsBusy|keepAliveCount|pollerThreadCount|connectionCount)'
  name: catalina_threadpool_$3
  labels:
```

```
    port: "$2"
    protocol: "$1"
  help: Catalina threadpool $3
  type: GAUGE

- pattern: 'Catalina<type=Manager, host=([-a-zA-Z0-9+&@#/%?=~_|!:.,;]*[-a-zA-
Z0-9+&@#/%=~_|]), context=([-a-zA-Z0-9+/$%~_-|!.]*)><>(processingTime|sessionCounter|
rejectedSessions|expiredSessions)'
  name: catalina_session_$3_total
  labels:
    context: "$2"
    host: "$1"
  help: Catalina session $3 total
  type: COUNTER

- pattern: ".*"
```

Inicie la aplicación Java con el exportador de Prometheus

Inicie la aplicación de ejemplo Esto emitirá métricas de Prometheus al puerto 9404. Asegúrese de sustituir el punto de entrada `com.gubupt.sample.app.App` con la información correcta para la aplicación java de muestra.

En Linux ingrese el siguiente comando:

```
$ nohup java -javaagent:./jmx_prometheus_javaagent-0.14.0.jar=9404:./config.yaml -cp
  ./SampleJavaApplication-1.0-SNAPSHOT.jar com.gubupt.sample.app.App &
```

En Windows ingrese el siguiente comando:

```
PS C:\> java -javaagent:.\jmx_prometheus_javaagent-0.14.0.jar=9404:.\config.yaml -cp  .
\SampleJavaApplication-1.0-SNAPSHOT.jar com.gubupt.sample.app.App
```

Verifique la emisión de métricas de Prometheus

Verifique que se estén emitiendo métricas de Prometheus.

En Linux ingrese el comando siguiente:

```
$ curl localhost:9404
```

En Windows ingrese el siguiente comando:

```
PS C:\> curl  http://localhost:9404
```

Salida de ejemplo en Linux:

```
StatusCode        : 200
StatusDescription : OK
Content           : # HELP jvm_classes_loaded The number of classes that are currently
 loaded in the JVM
                    # TYPE jvm_classes_loaded gauge
                    jvm_classes_loaded 2526.0
                    # HELP jvm_classes_loaded_total The total number of class...
RawContent        : HTTP/1.1 200 OK
                    Content-Length: 71908
                    Content-Type: text/plain; version=0.0.4; charset=utf-8
                    Date: Fri, 18 Dec 2020 16:38:10 GMT

                    # HELP jvm_classes_loaded The number of classes that are
 currentl...
Forms             : {}
Headers           : {[Content-Length, 71908], [Content-Type, text/plain; version=0.0.4;
 charset=utf-8], [Date, Fri, 18
                    Dec 2020 16:38:10 GMT]}
Images            : {}
InputFields       : {}
Links             : {}
ParsedHtml        : System.__ComObject
RawContentLength  : 71908
```

Paso 3: Configure el agente de CloudWatch para raspar las métricas de Prometheus

A continuación, configure la configuración de raspado de Prometheus en el archivo de configuración del agente CloudWatch.

Para establecer la configuración de raspado de Prometheus para el ejemplo de Java/JMX

1. Establezca la configuración de `file_sd_config` y `static_config`.

   En Linux ingrese el siguiente comando:

   ```
   $ cat /opt/aws/amazon-cloudwatch-agent/var/prometheus.yaml
   global:
     scrape_interval: 1m
   ```

```
  scrape_timeout: 10s
scrape_configs:
  - job_name: jmx
    sample_limit: 10000
    file_sd_configs:
      - files: [ "/opt/aws/amazon-cloudwatch-agent/var/prometheus_file_sd.yaml" ]
```

En Windows ingrese el siguiente comando:

```
PS C:\ProgramData\Amazon\AmazonCloudWatchAgent> cat prometheus.yaml
global:
  scrape_interval: 1m
  scrape_timeout: 10s
scrape_configs:
  - job_name: jmx
    sample_limit: 10000
    file_sd_configs:
      - files: [ "C:\\ProgramData\\Amazon\\AmazonCloudWatchAgent\
\prometheus_file_sd.yaml" ]
```

2. Establezca la configuración de los objetivos de raspado.

   En Linux ingrese el siguiente comando:

```
$ cat /opt/aws/amazon-cloudwatch-agent/var/prometheus_file_sd.yaml
- targets:
  - 127.0.0.1:9404
  labels:
    application: sample_java_app
    os: linux
```

   En Windows ingrese el siguiente comando:

```
PS C:\ProgramData\Amazon\AmazonCloudWatchAgent> cat prometheus_file_sd.yaml
- targets:
  - 127.0.0.1:9404
  labels:
    application: sample_java_app
    os: windows
```

3. Establezca la configuración de raspado de Prometheus mediante `ec2_sc_config`. Reemplace *`su-ec2-instance-id`* con el ID de instancia EC2 correcto.

En Linux ingrese el siguiente comando:

```
$ cat .\prometheus.yaml
global:
  scrape_interval: 1m
  scrape_timeout: 10s
scrape_configs:
  - job_name: jmx
    sample_limit: 10000
    ec2_sd_configs:
      - region: us-east-1
        port: 9404
        filters:
          - name: instance-id
            values:
              - your-ec2-instance-id
```

En Windows ingrese el siguiente comando:

```
PS C:\ProgramData\Amazon\AmazonCloudWatchAgent> cat prometheus_file_sd.yaml
- targets:
  - 127.0.0.1:9404
  labels:
    application: sample_java_app
    os: windows
```

4. Establezca la configuración del agente de CloudWatch. En primer lugar, vaya al directorio correcto. En Linux, es `/opt/aws/amazon-cloudwatch-agent/var/cwagent-config.json`. En Windows, es `C:\ProgramData\Amazon\AmazonCloudWatchAgent\cwagent-config.json`.

   La siguiente es una configuración de ejemplo con las métricas definidas de Java/JHX Prometheus. Asegúrese de reemplazar *path-to-Prometheus-Scrape-Configuration-file* con la ruta correcta.

```
{
  "agent": {
    "region": "us-east-1"
  },
  "logs": {
    "metrics_collected": {
```

```
    "prometheus": {
      "cluster_name": "my-cluster",
      "log_group_name": "prometheus-test",
      "prometheus_config_path": "path-to-Prometheus-Scrape-Configuration-file",
      "emf_processor": {
        "metric_declaration_dedup": true,
        "metric_namespace": "PrometheusTest",
        "metric_unit":{
          "jvm_threads_current": "Count",
          "jvm_classes_loaded": "Count",
          "java_lang_operatingsystem_freephysicalmemorysize": "Bytes",
          "catalina_manager_activesessions": "Count",
          "jvm_gc_collection_seconds_sum": "Seconds",
          "catalina_globalrequestprocessor_bytesreceived": "Bytes",
          "jvm_memory_bytes_used": "Bytes",
          "jvm_memory_pool_bytes_used": "Bytes"
        },
        "metric_declaration": [
          {
            "source_labels": ["job"],
            "label_matcher": "^jmx$",
            "dimensions": [["instance"]],
            "metric_selectors": [
              "^jvm_threads_current$",
              "^jvm_classes_loaded$",
              "^java_lang_operatingsystem_freephysicalmemorysize$",
              "^catalina_manager_activesessions$",
              "^jvm_gc_collection_seconds_sum$",
              "^catalina_globalrequestprocessor_bytesreceived$"
            ]
          },
          {
            "source_labels": ["job"],
            "label_matcher": "^jmx$",
            "dimensions": [["area"]],
            "metric_selectors": [
              "^jvm_memory_bytes_used$"
            ]
          },
          {
            "source_labels": ["job"],
            "label_matcher": "^jmx$",
            "dimensions": [["pool"]],
            "metric_selectors": [
```

```
                "^jvm_memory_pool_bytes_used$"
              ]
            }
          ]
        }
      }
    },
    "force_flush_interval": 5
  }
}
```

5. Para reiniciar el agente de CloudWatch, introduzca uno de los siguientes comandos.

   En Linux ingrese el siguiente comando:

   ```
   sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-
   config -m ec2 -s -c file:/opt/aws/amazon-cloudwatch-agent/var/cwagent-config.json
   ```

   En Windows ingrese el siguiente comando:

   ```
   & "C:\Program Files\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent-ctl.ps1"
    -a fetch-config -m ec2 -s -c file:C:\ProgramData\Amazon\AmazonCloudWatchAgent
   \cwagent-config.json
   ```

Visualización de las métricas y los registros de Prometheus

Ahora puede ver las métricas de Java/JMX que se están recopilando.

Para visualizar las métricas de la carga de trabajo de ejemplo de Java/JMX

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En la región en la que se está ejecutando el clúster, elija Metrics (Métricas) en el panel de navegación izquierdo. Busque el espacio de nombres PrometheusTest para ver las métricas.
3. Para ver los eventos de CloudWatch Logs, elija Log groups (Grupos de registros) en el panel de navegación. Los eventos están en el grupo de registro prometheus-test.

Instalar el agente de CloudWatch mediante el complemento de EKS de observabilidad de Amazon CloudWatch

El complemento de observabilidad de EKS de Amazon CloudWatch instala el agente de CloudWatch y el agente Fluent-bit en un clúster de Amazon EKS, con la observabilidad mejorada de la Información de contenedores para Amazon EKS y CloudWatch Application Signals habilitados de forma predeterminada. Con el complemento, puede recopilar métricas de infraestructura, la telemetría de rendimiento de las aplicaciones y los registros de contenedores del clúster de Amazon EKS.

Con Información de contenedores, con una observabilidad mejorada para Amazon EKS, las métricas de Información de contenedores se cobran por observación en lugar de cobrarse por métrica almacenada o registro ingerido. En el caso de Application Signals, la facturación se basa en las solicitudes entrantes y salientes de las aplicaciones y en cada objetivo de nivel de servicio (SLO) configurado. Cada solicitud entrante recibida genera una señal de aplicación y cada solicitud saliente realizada genera una señal de aplicación. Cada SLO crea dos señales de aplicación por período de medición. Para obtener más información sobre los precios de CloudWatch, consulte Precios de Amazon CloudWatch.

El complemento de Amazon EKS habilita Información de contenedores en los nodos de trabajo de Linux y Windows del clúster de Amazon EKS. Para habilitar Información de contenedores en Windows, debe usar la versión 1.5.0 o posterior del complemento de Amazon EKS. Actualmente, Application Signals no es compatible con Windows en clústeres de Amazon EKS.

El complemento de observabilidad de EKS de Amazon CloudWatch se encuentra disponible en los clústeres de Amazon EKS que se ejecutan con la versión 1.23 o posterior de Kubernetes.

Cuando instala el complemento, debe conceder permisos de IAM para permitir que el agente de CloudWatch envíe métricas, registros y seguimientos a CloudWatch. Hay dos formas de hacer esto:

- Adjunte una política al rol de IAM de los nodos de trabajo. Esta opción concede permisos a los nodos de trabajo para enviar telemetría a CloudWatch.
- Utilice un rol de IAM para las cuentas de servicio para los pods del agente y asocie la política a este rol. Esto solo funciona para clústeres de Amazon EKS. Esta opción permite que CloudWatch acceda solo a los pods de los agentes correspondientes.

Opción 1: lleve a cabo la instalación con permisos de IAM en los nodos de trabajo

Para usar este método, primero adjunte la política de IAM CloudWatchAgentServerPolicy a sus nodos de trabajo introduciendo el siguiente comando. En este comando, sustituya *my-worker-node-role* por el rol de IAM que utilizan los nodos de trabajo de Kubernetes.

```
aws iam attach-role-policy \
--role-name my-worker-node-role \
--policy-arn arn:aws:iam::aws:policy/CloudWatchAgentServerPolicy
```

A continuación, instale el complemento de observabilidad de EKS de Amazon CloudWatch. Para instalar el complemento, puede usar la AWS CLI, la consola, AWS CloudFormation o Terraform.

AWS CLI

Cómo utilizar la AWS CLI para instalar el complemento de observabilidad de EKS de Amazon CloudWatch

Escriba el siguiente comando. Reemplace *my-cluster-name* por el nombre del clúster.

```
aws eks create-addon --addon-name amazon-cloudwatch-observability --cluster-name my-cluster-name
```

Amazon EKS console

Cómo utilizar la consola de Amazon EKS para agregar el complemento de observabilidad de EKS de Amazon CloudWatch

1. Abra la consola de Amazon EKS en https://console.aws.amazon.com/eks/home#/clusters.
2. En el panel de navegación izquierdo, elija Clusters (Clústeres).
3. Seleccione el nombre del clúster para el que desea configurar el complemento de observabilidad de EKS de Amazon CloudWatch.
4. Elija la pestaña Complementos.
5. Escoja Obtener más complementos.
6. En la página Seleccionar complementos, haga lo siguiente:

    a. En la sección Complementos de Amazon EKS, seleccione la casilla de verificación para Observabilidad de Amazon CloudWatch.

b. Elija Siguiente.

7. En la página Configurar las opciones de complementos seleccionados, haga lo siguiente:

   a. Seleccione la Version (Versión) que desea utilizar.

   b. En Seleccionar el rol de IAM, seleccione Heredar del nodo

   c. (Opcional) Puede ampliar las Valores de configuración opcionales. Si selecciona Anular en Método de resolución de conflictos, una o varias de las configuraciones del complemento existente pueden sobrescribirse con la configuración del complemento de Amazon EKS. Si no habilita esta opción y hay un conflicto con la configuración existente, la operación falla. Puede utilizar el mensaje de error resultante para solucionar el conflicto. Antes de seleccionar esta opción, asegúrese de que el complemento de Amazon EKS no administra las configuraciones que se necesitan autoadministrar.

   d. Elija Siguiente.

8. En la página Revisar y añadir, elija Crear. Una vez finalizada la instalación del complemento, verá el complemento instalado.

AWS CloudFormation

Cómo usar la AWS CloudFormation para instalar el complemento de observabilidad de EKS de Amazon CloudWatch

Reemplace *my-cluster-name* por el nombre del clúster. Para obtener más información, consulte AWS::EKS::Addon.

```
{
    "Resources": {
        "EKSAddOn": {
            "Type": "AWS::EKS::Addon",
            "Properties": {
                "AddonName": "amazon-cloudwatch-observability",
                "ClusterName": "my-cluster-name"
            }
        }
    }
}
```

Terraform

Cómo usar Terraform para instalar el complemento de observabilidad de EKS de Amazon CloudWatch

Reemplace *my-cluster-name* por el nombre del clúster. Para obtener más información, consulte Recurso: aws_eks_addon.

```
resource "aws_eks_addon" "example" {
  addon_name   = "amazon-cloudwatch-observability"
  cluster_name = "my-cluster-name"
}
```

Opción 2: llevar a cabo la instalación mediante el rol de cuenta de servicio de IAM

Antes de usar este método, verifique los siguientes requisitos previos:

- Dispone de un clúster de Amazon EKS funcional con nodos asociados en una de las Regiones de AWS que admiten la Información de contenedores. Para obtener las lista de las regiones admitidas, consulte Información de contenedores.
- Tiene instalada y configurada `kubectl` para el clúster. Para obtener más información, consulte Instalación del `kubectl` en la Guía del usuario de Amazon EKS.
- Tiene instalada `eksctl`. Para obtener más información, consulte Instalación o actualización de `eksctl` en la Guía del usuario de Amazon EKS.

Para instalar el complemento de observabilidad de EKS de Amazon CloudWatch mediante el rol de cuenta de servicio de IAM

1. Ingrese el siguiente comando para crear un proveedor de OpenID Connect (OIDC), si el clúster aún no tiene uno. Para obtener más información, consulte Configuración de una cuenta de servicio de Kubernetes para asumir un rol de IAM en la Guía del usuario de Amazon EKS.

   ```
   eksctl utils associate-iam-oidc-provider --cluster my-cluster-name --approve
   ```

2. Ingrese el siguiente comando para crear el rol de IAM con la política CloudWatchAgentServerPolicy adjunta y configure la cuenta de servicio del agente para que asuma ese rol mediante OIDC. Reemplace *my-cluster-name* por el nombre del clúster y

reemplace *my-service-account-role* por el nombre del rol al que desea asociar la cuenta de servicio. Si el rol no existe aún, `eksctl` lo crea por usted.

```
eksctl create iamserviceaccount \
  --name cloudwatch-agent \
  --namespace amazon-cloudwatch --cluster my-cluster-name \
  --role-name my-service-account-role \
  --attach-policy-arn arn:aws:iam::aws:policy/CloudWatchAgentServerPolicy \
  --role-only \
  --approve
```

3. Instale el complemento ingresando el siguiente comando: Reemplace *my-cluster-name* por el nombre de su clúster, reemplace *111122223333* por el identificador de su cuenta y reemplace *my-service-account-role* por el rol de IAM creado en el paso anterior.

```
aws eks create-addon --addon-name amazon-cloudwatch-observability --cluster-
name my-cluster-name --service-account-role-arn arn:aws:iam::111122223333:role/my-
service-account-role
```

Configuraciones adicionales (Opcional)

Desactivación de la recopilación de registros de contenedores

De forma predeterminada, el complemento usa Fluent Bit para recopilar los registros de contenedores de todos los pods y, a continuación, los envía a los registros de CloudWatch. Para obtener información sobre los registros que se recopilan, consulte Configuración de Fluent Bit.

Para dejar de recopilar registros de contenedores, utilice la siguiente opción cuando cree o actualice el complemento:

```
--configuration-values '{ "containerLogs": { "enabled": false } }'
```

Desactivación de la recopilación de métricas de GPU de NVIDIA

A partir de la versión 1.300034.0 del agente de CloudWatch, Información de contenedores recopila las métricas de GPU de NVIDIA de las cargas de trabajo de EKS de forma predeterminada. Estas métricas se muestran en la tabla de Métricas de GPU de NVIDIA.

Para dejar de recopilar métricas de GPU de NVIDIA, establezca la opción `accelerated_compute_metrics` del archivo de configuración del agente de CloudWatch como

`false`. Esta opción se encuentra en la sección `kubernetes` de la sección `metrics_collected` del archivo de configuración de CloudWatch. A continuación, se muestra un ejemplo de configuración de desactivación.

```
{
  "agent": {
    "region": "us-east-1"
  },
  "logs": {
    "metrics_collected": {
      "emf": {
      },
      "kubernetes": {
        "enhanced_container_insights": true,
        "accelerated_compute_metrics": false
      }
    },
    "force_flush_interval": 5,
  }
}
```

Uso de una configuración del agente de CloudWatch personalizada

Para recopilar otras métricas, registros o seguimientos con el agente de CloudWatch, puede especificar una configuración personalizada y, al mismo tiempo, mantener habilitados la Información de contenedores y CloudWatch Application Signals. Para ello, incorpore el archivo de configuración del agente de CloudWatch en la clave de configuración situada debajo de la clave de agente de la configuración avanzada, que puede utilizar al crear o actualizar el complemento EKS. Lo siguiente representa la configuración del agente predeterminada cuando no proporciona configuraciones adicionales.

> **Important**
>
> Cualquier configuración personalizada que proporcione mediante ajustes de configuración adicionales anula la configuración predeterminada que utiliza el agente. Tenga cuidado de no deshabilitar de forma involuntaria las funciones que están habilitadas de forma predeterminada, como Información de contenedores con observabilidad mejorada y CloudWatch Application Signals. En el caso de que tenga que proporcionar una configuración

de agente personalizada, le recomendamos que utilice la siguiente configuración predeterminada como referencia y, a continuación, la modifique en consecuencia.

```
--configuration-values '{
  "agent": {
    "config": {
      "logs": {
        "metrics_collected": {
          "app_signals": {},
          "kubernetes": {
            "enhanced_container_insights": true
          }
        }
      },
      "traces": {
        "traces_collected": {
          "app_signals": {}
        }
      }
    }

}'
```

El siguiente ejemplo muestra la configuración de agente predeterminada para el agente de CloudWatch en Windows. El agente de CloudWatch en Windows no admite la configuración personalizada.

```
{
  "logs": {
    "metrics_collected": {
      "kubernetes": {
        "enhanced_container_insights": true
      },
    }
  }
}
```

Administración de los certificados TLS de webhook de admisión

El complemento de observabilidad de EKS de Amazon CloudWatch utiliza los webhooks de admisión de Kubernetes para validar y mutar las solicitudes de recursos personalizados (CR) de `AmazonCloudWatchAgent` y `Instrumentation` y, como alternativa, las solicitudes de pod de Kubernetes en el clúster si CloudWatch Application Signals está habilitado. En Kubernetes, los webhooks requieren un certificado TLS en el que el servidor API se configura para confiar y así garantizar una comunicación segura.

De forma predeterminada, el complemento de observabilidad de EKS de Amazon CloudWatch genera de forma automática una CA autofirmada y un certificado TLS firmado por esta CA para proteger la comunicación entre el servidor de API y el servidor de webhook. Este certificado generado de forma automática caduca de manera predeterminada a los 10 años y no se renueva automáticamente al expirar. Además, el paquete de CA y el certificado se vuelven a generar cada vez que se actualiza o reinstala el complemento, lo que restablece la caducidad. Si desea cambiar la caducidad predeterminada del certificado generado de forma automática, puede usar las siguientes configuraciones adicionales al crear o actualizar el complemento. Sustituya la *`fecha de caducidad en días`* por la duración de caducidad deseada en días.

```
--configuration-values '{ "admissionWebhooks": { "autoGenerateCert":
 { "expiryDays": expiry-in-days } } }'
```

Si busca una solución de autoridad de certificación más segura y con más características, el complemento incluye asistencia opcional para cert-manager, una solución ampliamente adoptada para la administración de certificados TLS en Kubernetes, que simplifica el proceso de obtención, renovación, administración y uso de esos certificados. Asegura que los certificados sean válidos y estén actualizados, e intenta renovarlos en un momento configurado antes de que caduquen. Además, cert-manager también facilita la emisión de certificados desde una variedad de fuentes compatibles, que incluye a AWS Certificate Manager Private Certificate Authority.

Le aconsejamos que revise las prácticas recomendadas para la administración de los certificados TLS en sus clústeres y que opte por cert-manager para entornos de producción. Tenga en cuenta que si opta por habilitar cert-manager para administrar los certificados de TLS de webhook de admisión, deberá preinstalar cert-manager en su clúster de Amazon EKS antes de instalar el complemento de observabilidad de EKS de Amazon CloudWatch. Consulte la documentación de cert-manager para obtener más información sobre las opciones de instalación disponibles. Después de instalarlo, puede optar por utilizar cert-manager para administrar los certificados de TLS de webhook de admisión mediante la siguiente configuración adicional al crear o actualizar el complemento.

```
--configuration-values '{ "admissionWebhooks": { "certManager": { "enabled":
 true } } }'
```

La configuración avanzada que se describe en esta sección utilizará de forma predeterminada un emisor autofirmado.

Recopilación de los identificadores de volumen de Amazon EBS

Si desea que se recopilen los identificadores de volumen de Amazon EBS en los registros de rendimiento, debe agregar otra política al rol de IAM asociado a los nodos de trabajo o a la cuenta de servicio. Añada lo siguiente como política insertada. Para obtener más información, consulte Agregar y eliminar permisos de identidad de IAM.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "ec2:DescribeVolumes"
            ],
            "Resource": "*",
            "Effect": "Allow"
        }
    ]
}
```

Solución de problemas del complemento de observabilidad de EKS de Amazon CloudWatch

Utilice la siguiente información para solucionar problemas con el complemento de observabilidad de EKS de Amazon CloudWatch.

Actualización y eliminación del complemento de observabilidad de EKS de Amazon CloudWatch

Para obtener instrucciones sobre cómo actualizar o eliminar el complemento de observabilidad de EKS de Amazon CloudWatch, consulte Administración de complementos de Amazon EKS. Utilice `amazon-cloudwatch-observability` como nombre del complemento.

Verificación de la versión del agente de CloudWatch que utiliza el complemento de observabilidad de EKS de Amazon CloudWatch

El complemento de observabilidad de EKS de Amazon CloudWatch instala un recurso personalizado del tipo `AmazonCloudWatchAgent`, que controla el comportamiento del daemonset del agente de CloudWatch en el clúster, incluida la versión del agente de CloudWatch que se utiliza. Puede obtener una lista de todos los recursos personalizados del `AmazonCloudWatchAgent` instalados en su clúster al ingresar el siguiente comando:

```
kubectl get amazoncloudwatchagent -A
```

En la salida de este comando, se puede comprobar la versión del agente de CloudWatch. Como alternativa, también puede describir el recurso del `amazoncloudwatchagent` o uno de los pods del `cloudwatch-agent-*` que se ejecutan en el clúster para inspeccionar la imagen que se utiliza.

Control de un error de configuración al administrar el complemento

Al instalar o actualizar el complemento de observabilidad de EKS de Amazon CloudWatch, si observa un error provocado por `Health Issue` del tipo `ConfigurationConflict` con una descripción que comienza con `Conflicts found when trying to apply. Will not continue due to resolve conflicts mode`, es probable que se deba a que ya tiene el agente de CloudWatch y sus componentes asociados, como ServiceAccount, ClusterRole y ClusterRoleBinding instalados en el clúster. Cuando el complemento intenta instalar el agente de CloudWatch y sus componentes asociados, si detecta algún cambio en el contenido, por defecto no se realiza la instalación o la actualización para evitar sobrescribir el estado de los recursos del clúster.

Si está intentando incorporar al complemento de observabilidad de EKS de Amazon CloudWatch y observa este error, le recomendamos que elimine la configuración de agente de CloudWatch existente que haya instalado anteriormente en el clúster y, a continuación, instale el complemento de EKS. Asegúrese de realizar una copia de seguridad de las personalizaciones que haya realizado en la configuración original del agente de CloudWatch, como la configuración de agente personalizada, y envíelas al complemento de observabilidad de EKS de Amazon CloudWatch la próxima vez que lo instale o actualice. Si ya había instalado el agente de CloudWatch para incorporar Información de contenedores, consulte Eliminación del agente de CloudWatch y Fluen Bit para Información de contenedores para obtener más información.

Como alternativa, el complemento admite una opción de configuración de resolución de conflictos que puede especificar `OVERWRITE`. Puede usar esta opción para continuar con la instalación o

actualización del complemento sobrescribiendo los errores en el clúster. Si utiliza la consola de Amazon EKS, encontrará el método de resolución de errores al elegir los ajustes de configuración opcionales al crear o actualizar el complemento. Si está utilizando la AWS CLI, puede proporcionar `--resolve-conflicts OVERWRITE` a su comando para crear o actualizar el complemento.

Métricas que el agente de CloudWatch ha recopilado

Puede recopilar las métricas de los servidores con el agente de CloudWatch en el servidor. Puede instalar el agente tanto en las instancias de Amazon EC2 como en los servidores en las instalaciones, y también en los servidores con Linux o Windows Server o macOS. Si instala el agente en una instancia de Amazon EC2, las métricas que recopila se suman a las métricas habilitadas de forma predeterminada en las instancias de Amazon EC2.

Para obtener información acerca de la instalación del agente de CloudWatch en una instancia, consulte Recopile las métricas, registros y seguimientos con el agente de CloudWatch.

Todas las métricas que se analizan en esta sección las recopila directamente el agente de CloudWatch.

Métricas que el agente de CloudWatch recopila en instancias de Windows Server

En un servidor con Windows Server, la instalación del agente de CloudWatch le permite recopilar las métricas asociadas con los contadores en la supervisión de rendimiento de Windows. Los nombres de métrica de CloudWatch para estos contadores se crean con la inclusión de un espacio entre el nombre de objeto y el nombre de contador. Por ejemplo, al contador `% Interrupt Time` del objeto `Processor` se le ha asignado el nombre de métrica `Processor % Interrupt Time` en CloudWatch. Para obtener más información acerca de los contadores de Monitor de rendimiento de Windows, consulte la documentación de Microsoft Windows Server.

El espacio de nombres predeterminado para las métricas que el agente de CloudWatch recopila es `CWAgent`, aunque se puede especificar otro espacio de nombres al configurar el agente.

Métricas que el agente de CloudWatch recopila en instancias de Linux y de macOS

En la siguiente tabla se muestran las métricas que se pueden recopilar con el agente de CloudWatch en servidores Linux y en equipos macOS.

Métrica	Descripción
`cpu_time_active`	El periodo de tiempo que la CPU está activa en cualquier capacidad. Esta métrica se mide en centésimas de segundo. Unidad: ninguna
`cpu_time_guest`	El periodo de tiempo que la CPU está ejecutando una CPU virtual para un sistema operativo invitado. Esta métrica se mide en centésimas de segundo. Unidad: ninguna
`cpu_time_guest_nice`	El periodo de tiempo que la CPU está ejecutando una CPU virtual para un sistema operativo invitado que es de baja prioridad y otros procesos pueden interrumpirlo. Esta métrica se mide en centésimas de segundo. Unidad: ninguna
`cpu_time_idle`	El periodo de tiempo que la CPU está inactiva. Esta métrica se mide en centésimas de segundo. Unidad: ninguna
`cpu_time_iowait`	El periodo de tiempo que la CPU está a la espera de que se completen las operaciones de entrada/salida. Esta métrica se mide en centésimas de segundo. Unidad: ninguna
`cpu_time_irq`	El periodo de tiempo que la CPU está atendiendo interrupciones. Esta métrica se mide en centésimas de segundo. Unidad: ninguna

Métrica	Descripción
cpu_time_nice	El periodo de tiempo que la CPU está en modo de usuario con procesos de baja prioridad que otros procesos de mayor prioridad pueden interrumpir fácilmente. Esta métrica se mide en centésimas de segundo. Unidad: ninguna
cpu_time_softirq	El periodo de tiempo que la CPU está atendiendo interrupciones de software. Esta métrica se mide en centésimas de segundo. Unidad: ninguna
cpu_time_steal	El periodo de tiempo que la CPU se encuentra en tiempo descartado, que es el tiempo empleado en otros sistemas operativos en un entorno virtualizado. Esta métrica se mide en centésimas de segundo. Unidad: ninguna
cpu_time_system	El periodo de tiempo que la CPU está en modo de sistema. Esta métrica se mide en centésimas de segundo. Unidad: ninguna
cpu_time_user	El periodo de tiempo que la CPU está en modo de usuario. Esta métrica se mide en centésimas de segundo. Unidad: ninguna
cpu_usage_active	El porcentaje de tiempo que la CPU está activa en cualquier capacidad. Unidad: porcentaje

Métrica	Descripción
cpu_usage_guest	El porcentaje de tiempo que la CPU está ejecutando una CPU virtual para un sistema operativo invitado. Unidad: porcentaje
cpu_usage_guest_nice	El porcentaje de tiempo que la CPU está ejecutando una CPU virtual para un sistema operativo invitado que es de baja prioridad y otros procesos pueden interrumpirlo. Unidad: porcentaje
cpu_usage_idle	El porcentaje de tiempo que la CPU está inactiva. Unidad: porcentaje
cpu_usage_iowait	El porcentaje de tiempo que la CPU está a la espera de que se completen las operaciones de entrada/salida. Unidad: porcentaje
cpu_usage_irq	El porcentaje de tiempo que la CPU está atendiendo interrupciones. Unidad: porcentaje
cpu_usage_nice	El porcentaje de tiempo que la CPU está en modo de usuario con procesos de baja prioridad, que otros procesos de mayor prioridad pueden interrumpir fácilmente. Unidad: porcentaje
cpu_usage_softirq	El porcentaje de tiempo que la CPU está atendiendo interrupciones de software. Unidad: porcentaje

Métrica	Descripción
cpu_usage_steal	El porcentaje de tiempo que la CPU se encuentra en tiempo descartado, que es el tiempo empleado en otros sistemas operativos en un entorno virtualizado. Unidad: porcentaje
cpu_usage_system	El porcentaje de tiempo que la CPU está en modo de sistema. Unidad: porcentaje
cpu_usage_user	El porcentaje de tiempo que la CPU está en modo de usuario. Unidad: porcentaje
disk_free	Espacio libre en los discos. Unidades: bytes
disk_inodes_free	El número de nodos de índice disponibles en el disco. Unidad: recuento
disk_inodes_total	El número total de nodos de índice reservados en el disco. Unidad: recuento
disk_inodes_used	El número de nodos de índice usados en el disco. Unidad: recuento
disk_total	Espacio total en los discos, incluido el usado y el libre. Unidades: bytes

Métrica	Descripción
`disk_used`	Espacio usado en los discos. Unidades: bytes
`disk_used_percent`	El porcentaje de espacio total en disco que está utilizado. Unidad: porcentaje
`diskio_iops_in_progress`	El número de solicitudes de E/S que se han enviado al controlador de dispositivo, pero todavía no han completado. Unidad: recuento
`diskio_io_time`	El periodo de tiempo que el disco ha tenido las solicitudes de E/S en cola. Unidad: milisegundos La única estadística que debe utilizarse para esta métrica es `Sum`. No utilice `Average`.
`diskio_reads`	El número de operaciones de lectura de disco. Unidad: recuento La única estadística que debe utilizarse para esta métrica es `Sum`. No utilice `Average`.
`diskio_read_bytes`	El número de bytes que se leyeron de los discos. Unidades: bytes La única estadística que debe utilizarse para esta métrica es `Sum`. No utilice `Average`.

Métrica	Descripción
`diskio_read_time`	El periodo de tiempo que las solicitudes de lectura han esperado en los discos. Si hay varias solicitudes de lectura en espera simultáneamente, se aumentará el número. Por ejemplo, si hay cinco solicitudes que esperan un promedio de 100 milisegundos, el valor registrado es 500. Unidad: milisegundos La única estadística que debe utilizarse para esta métrica es `Sum`. No utilice `Average`.
`diskio_writes`	El número de operaciones de escritura de disco. Unidad: recuento La única estadística que debe utilizarse para esta métrica es `Sum`. No utilice `Average`.
`diskio_write_bytes`	El número de bytes escritos en los discos. Unidades: bytes La única estadística que debe utilizarse para esta métrica es `Sum`. No utilice `Average`.
`diskio_write_time`	El periodo de tiempo que las solicitudes de escritura han esperado en los discos. Si hay varias solicitud es de escritura en espera simultáneamente, se aumentará el número. Por ejemplo, si hay ocho solicitudes que esperan un promedio de 1000 milisegundos, el valor registrado es 8000. Unidad: milisegundos La única estadística que debe utilizarse para esta métrica es `Sum`. No utilice `Average`.

Métrica	Descripción
`ethtool_bw_in_allowance_exceeded`	El número de paquetes en cola o eliminados debido a que la banda ancha agregada de bajada superó el máximo de la instancia. Esta métrica sólo se recopila si la ha incluido en la subsección `ethtool` de la sección `metrics_collected` del archivo de configuración del agente de CloudWatch Para obtener más información, consulte Recopilación de las métricas de rendimiento de la red. Unidad: ninguna
`ethtool_bw_out_allowance_exceeded`	El número de paquetes en cola o eliminados debido a que la banda ancha agregada de subida superó el máximo de la instancia. Esta métrica sólo se recopila si la ha incluido en la subsección `ethtool` de la sección `metrics_collected` del archivo de configuración del agente de CloudWatch Para obtener más información, consulte Recopilación de las métricas de rendimiento de la red. Unidad: ninguna

Métrica	Descripción
`ethtool_conntrack_allowance_exceeded`	El número de paquetes eliminados porque el seguimiento de conexiones superó el máximo de la instancia y no se pudieron establecer nuevas conexiones. Esto puede provocar la pérdida de paquetes para el tráfico hacia o desde la instancia. Esta métrica sólo se recopila si la ha incluido en la subsección `ethtool` de la sección `metrics_collected` del archivo de configuración del agente de CloudWatch Para obtener más información, consulte Recopilación de las métricas de rendimiento de la red. Unidad: ninguna
`ethtool_linklocal_allowance_exceeded`	El número de paquetes eliminados porque el PPS del tráfico a los servicios proxy locales superó el máximo para la interfaz de red. Esto afecta al tráfico hacia el servicio de DNS, el servicio de metadatos de instancia y el Servicio de sincronización temporal de Amazon. Esta métrica sólo se recopila si la ha incluido en la subsección `ethtool` de la sección `metrics_collected` del archivo de configuración del agente de CloudWatch Para obtener más información, consulte Recopilación de las métricas de rendimiento de la red. Unidad: ninguna

Métrica	Descripción
`ethtool_pps_allowance_exceeded`	El número de paquetes en cola o eliminados debido a que el PPS bidireccional superó el máximo de la instancia. Esta métrica sólo se recopila si la ha incluido en la subsección `ethtool` de la sección `metrics_collected` del archivo de configuración del agente de CloudWatch Para obtener más información, consulte Recopilación de las métricas de rendimiento de la red. Unidad: ninguna
`mem_active`	La cantidad de memoria que se ha utilizado de alguna manera durante el último período de muestreo. Unidades: bytes
`mem_available`	La cantidad de memoria que está disponible y que se puede asignar de manera instantánea a los procesos. Unidades: bytes
`mem_available_percent`	El porcentaje de memoria que está disponible y que se puede asignar de manera instantánea a los procesos. Unidad: porcentaje
`mem_buffered`	La cantidad de memoria que se utiliza para los búferes. Unidades: bytes

Métrica	Descripción
`mem_cached`	La cantidad de memoria que se utiliza para la memoria caché de archivo. Unidades: bytes
`mem_free`	La cantidad de memoria que no se está utilizando. Unidades: bytes
`mem_inactive`	La cantidad de memoria que no se ha utilizado de alguna manera durante el último período de muestreo. Unidades: bytes
`mem_total`	La cantidad total de memoria. Unidades: bytes
`mem_used`	La cantidad de memoria en uso actualmente. Unidades: bytes
`mem_used_percent`	El porcentaje de memoria en uso actualmente. Unidad: porcentaje
`net_bytes_recv`	El número de bytes recibidos por la interfaz de red. Unidades: bytes La única estadística que debe utilizarse para esta métrica es `Sum`. No utilice `Average`.
`net_bytes_sent`	El número de bytes enviados por la interfaz de red. Unidades: bytes La única estadística que debe utilizarse para esta métrica es `Sum`. No utilice `Average`.

Métrica	Descripción
net_drop_in	El número de paquetes recibidos por esta interfaz de red que se han descartado. Unidad: recuento La única estadística que debe utilizarse para esta métrica es Sum. No utilice Average.
net_drop_out	El número de paquetes transmitidos por esta interfaz de red que se han descartado. Unidad: recuento La única estadística que debe utilizarse para esta métrica es Sum. No utilice Average.
net_err_in	El número de errores de recepción detectados por esta interfaz de red. Unidad: recuento La única estadística que debe utilizarse para esta métrica es Sum. No utilice Average.
net_err_out	El número de errores de transmisión detectados por esta interfaz de red. Unidad: recuento La única estadística que debe utilizarse para esta métrica es Sum. No utilice Average.
net_packets_sent	El número de paquetes enviados por esta interfaz de red. Unidad: recuento La única estadística que debe utilizarse para esta métrica es Sum. No utilice Average.

Métrica	Descripción
net_packets_recv	El número de paquetes recibidos por esta interfaz de red. Unidad: recuento La única estadística que debe utilizarse para esta métrica es Sum. No utilice Average.
netstat_tcp_close	El número de conexiones TCP sin estado. Unidad: recuento
netstat_tcp_close_wait	El número de conexiones TCP esperando una solicitud de finalización del cliente. Unidad: recuento
netstat_tcp_closing	El número de conexiones TCP que están esperando una solicitud de finalización con confirmación del cliente. Unidad: recuento
netstat_tcp_established	El número de conexiones TCP establecidas. Unidad: recuento
netstat_tcp_fin_wait1	El número de conexiones TCP en estado FIN_WAIT1 durante el proceso de cierre de una conexión. Unidad: recuento
netstat_tcp_fin_wait2	El número de conexiones TCP en estado FIN_WAIT2 durante el proceso de cierre de una conexión. Unidad: recuento

Métrica	Descripción
`netstat_tcp_last_ack`	El número de conexiones TCP esperando a que el cliente envíe la confirmación del mensaje de finalizac ión de conexión. Es el último estado antes de que se cierre la conexión. Unidad: recuento
`netstat_tcp_listen`	El número de puertos TCP que se escuchan actualmente para una solicitud de conexión. Unidad: recuento
`netstat_tcp_none`	El número de conexiones TCP con clientes inactivos. Unidad: recuento
`netstat_tcp_syn_sent`	El número de conexiones TCP que esperan una solicitud de conexión coincidente después de haber enviado una solicitud de conexión. Unidad: recuento
`netstat_tcp_syn_recv`	El número de conexiones TCP que esperan una confirmación de solicitud de conexión después de haber enviado y recibido una solicitud de conexión. Unidad: recuento
`netstat_tcp_time_wait`	El número de conexiones TCP en espera actualmen te para garantizar que el cliente ha recibido la confirmación de su solicitud de finalización de conexión. Unidad: recuento
`netstat_udp_socket`	El número de conexiones UDP actuales. Unidad: recuento

Métrica	Descripción
`processes_blocked`	El número de procesos que están bloqueados. Unidad: recuento
`processes_dead`	El número de procesos que están “muertos”, lo que se indica mediante el código de estado `X` en Linux. Esta métrica no se recopila en ordenadores con macOS. Unidad: recuento
`processes_idle`	El número de procesos que han estado inactivos (sin actividad durante más de 20 segundos). Disponible en instancias de FreeBSD. Unidad: recuento
`processes_paging`	El número de procesos que están paginando, lo que se indica mediante el código de estado `W` en Linux. Esta métrica no se recopila en ordenadores con macOS. Unidad: recuento
`processes_running`	El número de procesos que están en ejecución, lo que se indica mediante el código de estado `R`. Unidad: recuento
`processes_sleeping`	El número de procesos que están inactivos, lo que se indica mediante el código de estado `S`. Unidad: recuento

Métrica	Descripción
processes_stopped	El número de procesos que están detenidos, lo que se indica mediante el código de estado T. Unidad: recuento
processes_total	El número total de procesos en la instancia. Unidad: recuento
processes_total_threads	El número total de subprocesos que componen los procesos. Esta métrica solo está disponible en instancias Linux. Esta métrica no se recopila en ordenadores con macOS. Unidad: recuento
processes_wait	El número de procesos que están paginando, lo que se indica mediante el código de estado W en instancias de FreeBSD. Esta métrica sólo está disponible en instancias de FreeBSD y no está disponible en instancias de Linux, Windows Server o de macOS. Unidad: recuento
processes_zombies	El número de procesos zombis, lo que se indica mediante el código de estado Z. Unidad: recuento
swap_free	La cantidad de espacio de intercambio que no se está utilizando. Unidades: bytes

Métrica	Descripción
`swap_used`	La cantidad de espacio de intercambio en uso actualmente. Unidades: bytes
`swap_used_percent`	El porcentaje de espacio de intercambio en uso actualmente. Unidad: porcentaje

Definiciones de las métricas de memoria que el agente CloudWatch ha recopilado

Cuando el agente CloudWatch recopila métricas de memoria, la fuente es el subsistema de administración de memoria del host. Por ejemplo, el núcleo Linux expone los datos mantenidos por el sistema operativo en `/proc`. En el caso de la memoria, los datos están en `/proc/meminfo`.

Cada arquitectura y sistema operativo distintos tienen cálculos diferentes de los recursos que utilizan los procesos. Para obtener más información, consulte las siguientes secciones.

Durante cada intervalo de recopilación, el agente de CloudWatch de cada instancia recopila los recursos de la instancia y calcula los recursos que utilizan todos los procesos que se ejecutan en esa instancia. Esta información se remite a las métricas de CloudWatch. Puede configurar la duración del intervalo de recopilación en el archivo de configuración del agente de CloudWatch. Para obtener más información, consulte Archivo de configuración del agente de CloudWatch: sección del agente.

En la siguiente lista se explica cómo se definen las métricas de memoria que recopila el agente de CloudWatch.

- Memoria activa: memoria que utiliza un proceso. En otras palabras, la memoria que utilizan las aplicaciones que se están ejecutando.
- Memoria disponible: memoria que se puede asignar instantáneamente a los procesos sin que el sistema la intercambie (también conocida como memoria virtual).
- Memoria intermedia: área de datos que comparten los dispositivos de hardware o los procesos de programa que funcionan a diferentes velocidades y prioridades.

- Memoria en caché: almacena las instrucciones y los datos del programa que se utilizan repetidamente en el funcionamiento de los programas que probablemente la CPU necesite a continuación.
- Memoria libre: memoria que no se utiliza en absoluto y que está fácilmente disponible. Es completamente gratuito para que el sistema pueda usarse cuando sea necesario.
- Memoria inactiva: páginas a las que no se ha accedido “recientemente”.
- Memoria total: el tamaño de la memoria RAM física real.
- Memoria usada: memoria que los programas y procesos utilizan actualmente.

Temas

Linux: métricas recopiladas y cálculos utilizados

Métricas recopiladas y unidades:

- Activo (bytes)
- Disponible (bytes)
- Porcentaje disponible (porcentaje)
- Almacenado en búfer (bytes)
- En caché (bytes)
- Gratis (bytes)
- Inactivo (bytes)
- Total (Bytes)
- Usado (bytes)
- Porcentaje de uso (por ciento)

Memoria utilizada = Memoria total - Memoria libre - Memoria en caché - Memoria en búfer

Memoria total = Memoria utilizada + Memoria libre + Memoria caché + Memoria en búfer

macOS: métricas recopiladas y cálculos utilizados

Métricas recopiladas y unidades:

- Activo (bytes)
- Disponible (bytes)
- Porcentaje disponible (porcentaje)
- Gratis (bytes)
- Inactivo (bytes)
- Total (Bytes)
- Usado (bytes)
- Porcentaje de uso (por ciento)

Memoria disponible = memoria libre + memoria inactiva

Memoria usada = Memoria total - Memoria disponible

Memoria total = Memoria disponible + Memoria utilizada

Windows: métricas recopiladas

Las métricas recopiladas en los hosts de Windows se muestran a continuación. Todas estas métricas tienen `None` para `Unit`.

- Bytes disponibles
- Fallos de caché por segundo
- Errores de página por segundo
- Páginas por segundo

No se utilizan cálculos para las métricas de Windows porque el agente de CloudWatch analiza los eventos de los contadores de rendimiento.

Ejemplo: calcular las métricas de memoria en Linux

Como ejemplo, supongamos que al introducir el comando **`cat /proc/meminfo`** en un host Linux se obtienen los siguientes resultados:

```
MemTotal:       3824388 kB
MemFree:         462704 kB
MemAvailable:   2157328 kB
Buffers:         126268 kB
Cached:         1560520 kB
SReclaimable:    289080 kB>
```

En este ejemplo, el agente de CloudWatch recopilará los siguientes valores. Todos los valores que recopila e informa el agente de CloudWatch están en bytes.

- `mem_total`: 3 916 173 312 bytes
- `mem_available`: 2 209 103 872 bytes (MemFree + Cached)
- `mem_free`: 473 808 896 bytes
- `mem_cached`: 1 893 990 400 bytes (`cached` + `SReclaimable`
- `mem_used`: 1 419 075 584 bytes (`MemTotal` – (`MemFree` + `Buffers` + (`Cached` + `SReclaimable`)))
- `mem_buffered`: 129 667 072 bytes
- `mem_available_percent`: 56,41 %
- `mem_used_percent`: 36,24 % (`mem_used`/`mem_total`) * 100

Escenarios comunes con el agente de CloudWatch

En las siguientes secciones se describe cómo se completan algunas de las tareas comunes de configuración y personalización con el agente de CloudWatch.

Temas

- Diferencias de marcas de tiempo entre el agente unificado de CloudWatch y el agente de CloudWatch Logs anterior

Ejecución del agente de CloudWatch como otro usuario

En los servidores Linux, CloudWatch se ejecuta como el superusuario de forma predeterminada. Para que el agente se ejecute como otro usuario, utilice el parámetro `run_as_user` en la sección `agent` del archivo de configuración del agente de CloudWatch. Esta opción solo está disponible en los servidores Linux.

Si ya está ejecutando el agente con el usuario raíz y desea usar otro usuario, utilice uno de los siguientes procedimientos.

Para ejecutar el agente de CloudWatch como otro usuario en una instancia EC2 que ejecuta Linux

1. Descargue e instale un nuevo paquete del agente de CloudWatch. Para obtener más información, consulte Descargue del paquete de del agente de CloudWatch.
2. Cree un nuevo usuario de Linux o utilice el nombre de usuario de Linux predeterminado `cwagent` que creó el archivo RPM o DEB.
3. Proporcione credenciales para este usuario en una de estas formas:
   - Si el archivo `.aws/credentials` existe en el directorio inicial del superusuario, debe crear un archivo de credenciales para el usuario que va a utilizar para ejecutar el agente de CloudWatch. Este archivo de credenciales será /home/*username*/`.aws/credentials`. A continuación, establezca el valor del parámetro `shared_credential_file` en `common-config.toml` para el nombre de ruta del archivo de credenciales. Para obtener más información, consulte (Opcional) Modifique la configuración común para la información del proxy o de la región.
   - Si el archivo `.aws/credentials` no existe en el directorio inicial del superusuario, puede realizar una de las siguientes opciones:
     - Cree un archivo de credenciales para el usuario que va a utilizar para ejecutar el agente de CloudWatch. Este archivo de credenciales será /home/*username*/`.aws/credentials`. A continuación, establezca el valor del parámetro `shared_credential_file` en `common-config.toml` para el nombre de ruta del archivo de credenciales. Para obtener más información, consulte (Opcional) Modifique la configuración común para la información del proxy o de la región.

- En lugar de crear un archivo de credenciales, adjunte un rol de IAM a la instancia. El agente utiliza este rol como el proveedor de credenciales.

4. En el archivo de configuración del agente de CloudWatch, agregue la siguiente línea a la sección `agent`:

```
"run_as_user": "username"
```

   Realice otras modificaciones en el archivo de configuración según sea necesario. Para obtener más información, consulte Cree el archivo de configuración del agente de CloudWatch

5. Otorgar al usuario los permisos necesarios. El usuario debe tener permisos Read (r) para los archivos de registro que se van a recopilar y el permiso Execute (x) en cada directorio de la ruta de los archivos de registro.

6. Inicie el agente con el archivo de configuración que acaba de modificar.

```
sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-
config -m ec2 -s -c file:configuration-file-path
```

Para ejecutar el agente de CloudWatch como otro usuario en un servidor en las instalaciones que ejecuta Linux

1. Descargue e instale un paquete nuevo del agente de CloudWatch. Para obtener más información, consulte Descargue del paquete de del agente de CloudWatch.

2. Cree un nuevo usuario de Linux o utilice el nombre de usuario de Linux predeterminado `cwagent` que creó el archivo RPM o DEB.

3. Almacene las credenciales de este usuario en una ruta a la que el usuario pueda obtener acceso, como `/home/username/.aws/credentials`.

4. Establezca el valor del parámetro `shared_credential_file` en `common-config.toml` para nombre de ruta del archivo de credenciales. Para obtener más información, consulte (Opcional) Modifique la configuración común para la información del proxy o de la región.

5. En el archivo de configuración del agente de CloudWatch, agregue la siguiente línea a la sección `agent`:

```
"run_as_user": "username"
```

Realice otras modificaciones en el archivo de configuración según sea necesario. Para obtener más información, consulte Cree el archivo de configuración del agente de CloudWatch

6. Otorgar los permisos necesarios para el usuario. El usuario debe tener permisos Read (r) para los archivos de registro que se van a recopilar y el permiso Execute (x) en cada directorio de la ruta de los archivos de registro.
7. Inicie el agente con el archivo de configuración que acaba de modificar.

```
sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-
config -m ec2 -s -c file:configuration-file-path
```

Cómo el agente de CloudWatch maneja los archivos de registro dispersos

Los archivos dispersos son archivos con bloques vacíos y contenido real. Un archivo disperso utiliza el espacio en disco de manera más eficiente escribiendo información breve que representa los bloques vacíos en el disco en lugar de los bytes nulos reales que componen el bloque. Esto hace que el tamaño real de un archivo disperso sea generalmente mucho más pequeño que su tamaño aparente.

Sin embargo, el agente de CloudWatch no trata los archivos dispersos de manera distinta a los archivos normales. Cuando el agente lee un archivo disperso, los bloques vacíos se tratan como bloques «reales» rellenos de bytes nulos. Debido a esto, el agente de CloudWatch publica tantos bytes como el tamaño aparente de un archivo disperso en CloudWatch.

La configuración del agente de CloudWatch para publicar un archivo disperso puede causar costos de CloudWatch superiores a los esperados, por lo que se recomienda no configurarlo de esa manera. Por ejemplo, `/var/logs/lastlog` en Linux suele ser un archivo muy disperso y se recomienda que no lo publique en CloudWatch.

Adición de dimensiones personalizadas a métricas que el agente de CloudWatch recopila

Para añadir dimensiones personalizadas, como etiquetas, a métricas recopiladas por el agente, añada el campo `append_dimensions` a la sección del archivo de configuración del agente que enumera dichas métricas.

Por ejemplo, la siguiente sección de ejemplo del archivo de configuración añade una dimensión personalizada denominada `stackName` con un valor de `Prod` a las métricas `cpu` y `disk` recopiladas por el agente.

```
"cpu":{
  "resources":[
    "*"
  ],
  "measurement":[
    "cpu_usage_guest",
    "cpu_usage_nice",
    "cpu_usage_idle"
  ],
  "totalcpu":false,
  "append_dimensions":{
    "stackName":"Prod"
  }
},
"disk":{
  "resources":[
    "/",
    "/tmp"
  ],
  "measurement":[
    "total",
    "used"
  ],
  "append_dimensions":{
    "stackName":"Prod"
  }
}
```

Recuerde que cada vez que cambie el archivo de configuración del agente, debe reiniciar el agente para que los cambios surtan efecto.

Varios archivos de configuración del agente de CloudWatch

En los servidores Linux y Windows, puede configurar el agente de CloudWatch para que utilice varios archivos de configuración. Por ejemplo, puede utilizar un archivo de configuración común que recopile un conjunto de métricas, registros y seguimientos que siempre desea recopilar de todos los servidores de la infraestructura. Y también puede utilizar archivos de configuración adicionales que recopilen métricas de determinadas aplicaciones o en situaciones concretas.

Para realizar esta configuración, primero debe crear los archivos de configuración que desea utilizar. Los archivos de configuración que se vayan a utilizar conjuntamente en el mismo servidor deben tener nombres diferentes. Puede almacenar los archivos de configuración en los servidores o en el almacén de parámetros.

Inicie el agente de CloudWatch mediante la opción `fetch-config` y especifique el primer archivo de configuración. Para añadir el segundo archivo de configuración al agente en ejecución, utilice el mismo comando pero con la opción `append-config`. Se recopilan todas las métricas, registros y seguimientos indicados en los dos archivos de configuración. Los siguientes comandos de ejemplo ilustran este escenario mediante almacenes de configuración como archivos. La primera línea inicia el agente mediante el archivo de configuración `infrastructure.json` y la segunda añade el archivo de configuración `app.json`.

Los siguientes comandos de ejemplo son para Linux.

```
/opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-config -m ec2
 -s -c file:/tmp/infrastructure.json
```

```
/opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a append-config -m
 ec2 -s -c file:/tmp/app.json
```

Los siguientes comandos de ejemplo son para Windows Server.

```
& "C:\Program Files\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent-ctl.ps1"
 -a fetch-config -m ec2 -s -c file:"C:\Program Files\Amazon\AmazonCloudWatchAgent
\infrastructure.json"
```

```
& "C:\Program Files\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent-ctl.ps1"
 -a append-config -m ec2 -s -c file:"C:\Program Files\Amazon\AmazonCloudWatchAgent
\app.json"
```

Los siguientes archivos de configuración de ejemplo ilustran un uso de esta característica. El primer archivo de configuración se utiliza para todos los servidores de la infraestructura y el segundo solo recopila los registros de una determinada aplicación y se asocia a los servidores que ejecutan dicha aplicación.

infrastructure.json

```
{
```

```
  "metrics": {
    "metrics_collected": {
      "cpu": {
        "resources": [
          "*"
        ],
        "measurement": [
          "usage_active"
        ],
        "totalcpu": true
      },
      "mem": {
         "measurement": [
           "used_percent"
        ]
      }
    }
  },
  "logs": {
    "logs_collected": {
      "files": {
        "collect_list": [
          {
            "file_path": "/opt/aws/amazon-cloudwatch-agent/logs/amazon-cloudwatch-
agent.log",
            "log_group_name": "amazon-cloudwatch-agent.log"
          },
          {
            "file_path": "/var/log/messages",
            "log_group_name": "/var/log/messages"
          }
        ]
      }
    }
  }
}
```

app.json

```
{
    "logs": {
        "logs_collected": {
            "files": {
```

```
                "collect_list": [
                    {
                        "file_path": "/app/app.log*",
                        "log_group_name": "/app/app.log"
                    }
                ]
            }
        }
    }
}
```

Los archivos de configuración que se añadan a la configuración deben tener nombres diferentes y distintos del archivo de configuración inicial. Si utiliza `append-config` con un archivo de configuración que tiene el mismo nombre que otro que ya está utilizando el agente, el comando append sobrescribe la información del primer archivo de configuración en lugar de añadirle nuevo contenido. Esto es válido incluso si los dos archivos de configuración con el mismo nombre se encuentran en diferentes rutas de archivo.

El ejemplo anterior muestra el uso de dos archivos de configuración, pero no existe ningún límite respecto al número de archivos de configuración que se pueden añadir a la configuración del agente. También puede combinar el uso de archivos de configuración ubicados en servidores con las configuraciones ubicadas en el almacén de parámetros.

Adición o acumulación de las métricas que el agente de CloudWatch recopila

Para agregar o acumular las métricas recopiladas por el agente, añada un campo `aggregation_dimensions` a la sección de dicha métrica en el archivo de configuración del agente.

Por ejemplo, el siguiente fragmento del archivo de configuración acumula métricas en la dimensión `AutoScalingGroupName`. Se agregan las métricas de todas las instancias de cada grupo de Auto Scaling y se pueden ver como un conjunto.

```
"metrics": {
  "cpu":{...}
  "disk":{...}
  "aggregation_dimensions" : [["AutoScalingGroupName"]]
}
```

Para acumularlas en función de la combinación de cada una de las dimensiones `InstanceId` y `InstanceType`, además de acumularlas por el nombre de grupo de Auto Scaling, agregue lo siguiente.

```
"metrics": {
  "cpu":{...}
  "disk":{...}
  "aggregation_dimensions" : [["AutoScalingGroupName"], ["InstanceId", "InstanceType"]]
}
```

Para acumular las métricas en una colección en su lugar, utilice `[]`.

```
"metrics": {
  "cpu":{...}
  "disk":{...}
  "aggregation_dimensions" : [[]]
}
```

Recuerde que cada vez que cambie el archivo de configuración del agente, debe reiniciar el agente para que los cambios surtan efecto.

Recopilación de métricas de alta resolución con el agente de CloudWatch

El campo `metrics_collection_interval` especifica el intervalo de tiempo de las métricas recopiladas, en segundos. Al especificar un valor de menos de 60 para este campo, las métricas se recopilan como métricas de alta resolución.

Por ejemplo, si todas las métricas deben ser de alta resolución y se recopilan cada 10 segundos, especifique 10 como el valor de `metrics_collection_interval` en la sección `agent` como un intervalo de recopilación de métricas globales.

```
"agent": {
  "metrics_collection_interval": 10
}
```

De forma alternativa, el siguiente ejemplo establece las métricas de `cpu` que se van a recopilar cada segundo, mientras que las demás métricas se recopilan cada minuto.

```
"agent":{
  "metrics_collection_interval": 60
```

```
},
"metrics":{
  "metrics_collected":{
    "cpu":{
      "resources":[
        "*"
      ],
      "measurement":[
        "cpu_usage_guest"
      ],
      "totalcpu":false,
      "metrics_collection_interval": 1
    },
    "disk":{
      "resources":[
        "/",
        "/tmp"
      ],
      "measurement":[
        "total",
        "used"
      ]
    }
  }
}
```

Recuerde que cada vez que cambie el archivo de configuración del agente, debe reiniciar el agente para que los cambios surtan efecto.

Envío de métricas, registros y seguimientos a una cuenta diferente

Para que el agente de CloudWatch envíe las métricas, los registros o los seguimientos a una cuenta diferente, especifique un parámetro `role_arn` en el archivo de configuración del agente en el servidor de envío. El valor de `role_arn` especifica un rol de IAM en la cuenta de destino que el agente utiliza al enviar datos a dicha cuenta. Este rol permite que la cuenta de envío asuma un rol correspondiente en la cuenta de destino al enviar las métricas o los registros a la cuenta de destino.

También puede especificar cadenas `role_arn` distintas en el archivo de configuración del agente: una para utilizarla al enviar métricas, otra para enviar registros y otra para enviar seguimientos.

El siguiente ejemplo de parte de la sección `agent` del archivo de configuración establece que el agente utilice `CrossAccountAgentRole` al enviar datos a una cuenta diferente.

```
{
  "agent": {
    "credentials": {
      "role_arn": "arn:aws:iam::123456789012:role/CrossAccountAgentRole"
    }
  },
  .....
}
```

Por otro lado, el siguiente ejemplo establece diferentes funciones para la cuenta de envío cuando se envían métricas, registros y seguimientos:

```
"metrics": {
    "credentials": {
     "role_arn": "RoleToSendMetrics"
    },
    "metrics_collected": {....
```

```
"logs": {
    "credentials": {
    "role_arn": "RoleToSendLogs"
    },
    ....
```

Políticas necesarias

Cuando especifica un `role_arn` en el archivo de configuración del agente, también debe asegurarse de que los roles de IAM de las cuentas de envío y de destino cuenten con determinadas políticas. Los roles de las cuentas de envío y de destino deben tener la política `CloudWatchAgentServerPolicy`. Para obtener más información sobre cómo asignar esta política a un rol, consulte Cree roles de IAM para utilizarlos con el agente de CloudWatch en instancias de Amazon EC2.

El rol de la cuenta de envío también debe incluir la política siguiente. Esta política se añade en la pestaña Permissions (Permisos) de la consola de IAM al editar el rol.

```
{
    "Version": "2012-10-17",
```

```
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "sts:AssumeRole"
            ],
            "Resource": [
                "arn:aws:iam::target-account-ID:role/agent-role-in-target-account"
            ]
        }
    ]
}
```

El rol de la cuenta de destino debe incluir la siguiente política, de modo que reconozca el rol de IAM que la cuenta de envío utilice. Esta política se añade en la pestaña Trust relationships (Relaciones de confianza) de la consola de IAM al editar el rol. El rol de la cuenta de destino a la que se añade esta política es el rol que creó en Cree roles de IAM y usuarios para utilizarlos con el agente de CloudWatch. Este rol es el rol especificado en *agent-role-in-target-account* en la política utilizada por la cuenta de envío.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::sending-account-ID:role/role-in-sender-account"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Diferencias de marcas de tiempo entre el agente unificado de CloudWatch y el agente de CloudWatch Logs anterior

El agente de CloudWatch admite un conjunto diferente de símbolos para formatos de marca de tiempo, en comparación con el agente de CloudWatch Logs anterior. Estas diferencias se muestran en la siguiente tabla.

Símbolos que ambos agentes admiten	Símbolos que solo el agente unificado de CloudWatch admite	Símbolos que solo el agente de CloudWatch Logs anterior admite
%A, %a, %b, %B, %d, %f, %H, %I, %m, %M, %p, %S, %y, %Y, %Z, %z	%-d, %-I, %-m, %-M, %-S	%c,%j, %U, %W, %w

Para obtener más información sobre el significado de los símbolos que admite el nuevo agente de CloudWatch, consulte CloudWatch Agent Configuration File: Logs Section (Archivo de configuración del agente de CloudWatch: sección de registros) en la Guía del usuario de Amazon CloudWatch. Para obtener más información acerca de los símbolos que admite el agente de CloudWatch Logs, consulte Agent Configuration File (Archivo de configuración del agente) en la Guía del usuario de Amazon CloudWatch Logs.

Solución de problemas del agente de CloudWatch

Utilice la siguiente información para ayudarle a solucionar problemas con el agente de CloudWatch.

Temas

- Cierre y reinicio del agente de CloudWatch

Parámetros de la línea de comandos del agente de CloudWatch

Para ver la lista completa de los parámetros que el agente de CloudWatch admite, ingrese lo siguiente en la línea de comandos en un ordenador en el que lo tenga instalado:

```
amazon-cloudwatch-agent-ctl -help
```

Error al instalar el agente de CloudWatch mediante Run Command

Para instalar el agente de CloudWatch mediante Run Command de Systems Manager, el SSM Agent en el servidor de destino debe contar con la versión 2.2.93.0 o con una posterior. Si SSM Agent no cuenta con la versión correcta, es posible que vea errores que incluyen los siguientes mensajes:

```
no latest version found for package AmazonCloudWatchAgent on platform linux
```

```
failed to download installation package reliably
```

Para obtener información sobre la actualización de la versión de SSM Agent, consulte Installing and Configuring SSM Agent (Instalación y configuración de SSM Agent) en la Guía del usuario de AWS Systems Manager.

El agente de CloudWatch no se iniciará

Si el agente de CloudWatch no se inicia, podría haber un problema en la configuración. La información de configuración se registra en el archivo `configuration-validation.log`. Este archivo se encuentra en `/opt/aws/amazon-cloudwatch-agent/logs/configuration-validation.log` en los servidores Linux y en `$Env:ProgramData\Amazon\AmazonCloudWatchAgent\Logs\configuration-validation.log` en los servidores con Windows Server.

Verifique que el agente de CloudWatch esté en ejecución

Puede consultar el agente de CloudWatch para determinar si está en ejecución o detenido. Puede usar AWS Systems Manager, para hacerlo de forma remota. También puede utilizar la línea de comandos, pero solo para comprobar el servidor local.

Para consultar el estado del agente de CloudWatch mediante Run Command

1. Abra la consola de Administrador de sistemas en https://console.aws.amazon.com/systems-manager/.
2. En el panel de navegación, elija Ejecutar comando.

   -o bien-

   Si la página de inicio de AWS Systems Manager se abre, desplácese hacia abajo y elija Explore Run Command (Explorar Run Command).
3. Elija Run command (Ejecutar comando).
4. En la lista Documento de comando, elija el botón situado junto a AmazonCloudWatch-ManageAgent.
5. En la lista Action, elija status.
6. En Origen de configuración opcional elija predeterminado y mantenga la Ubicación de configuración opcional en blanco.
7. En el área Target, elija la instancia que comprobar.
8. Elija Ejecutar.

Si el agente está en ejecución, el resultado será similar al siguiente.

```
{
       "status": "running",
       "starttime": "2017-12-12T18:41:18",
       "version": "1.73.4"
}
```

Si el agente está detenido, el campo `"status"` muestra `"stopped"`.

Para consultar el estado del agente de CloudWatch localmente mediante la línea de comandos

- En un servidor Linux, escriba lo siguiente:

  ```
  sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -m ec2 -a
   status
  ```

  En un servidor con Windows Server, escriba lo siguiente en PowerShell como administrador:

```
& $Env:ProgramFiles\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent-ctl.ps1 -m
 ec2 -a status
```

El agente de CloudWatch no se iniciará y el error menciona la región de Amazon EC2

Si el agente no se inicia y el mensaje de error menciona un punto de enlace de una región de Amazon EC2, es posible que haya configurado el agente para que necesite acceso al punto de enlace de Amazon EC2, pero que no haya concedido ese tipo de acceso.

Por ejemplo, si especifica un valor para el parámetro `append_dimensions` en el archivo de configuración del agente que depende de los metadatos de Amazon EC2 y utiliza proxies, debe asegurarse de que el servidor pueda obtener acceso al punto de enlace de Amazon EC2. Para obtener más información sobre estos puntos de conexión, consulte Amazon Elastic Compute Cloud (Amazon EC2) en Referencia general de Amazon Web Services.

El agente de CloudWatch no se iniciará en Windows Server

Puede aparecer el siguiente error en Windows Server:

```
Start-Service : Service 'Amazon CloudWatch Agent (AmazonCloudWatchAgent)' cannot be
 started due to the following
error: Cannot start service AmazonCloudWatchAgent on computer '.'.
At C:\Program Files\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent-ctl.ps1:113
 char:12
+     $svc | Start-Service
+            ~~~~~~~~~~~~~
    + CategoryInfo          : OpenError:
 (System.ServiceProcess.ServiceController:ServiceController) [Start-Service],
   ServiceCommandException
    + FullyQualifiedErrorId :
 CouldNotStartService,Microsoft.PowerShell.Commands.StartServiceCommand
```

Para solucionar esto, asegúrese antes de que el servicio del servidor se esté ejecutando. Este error se puede ver si el agente intenta iniciarse cuando el servicio del servidor no se está ejecutando.

Si el servicio del servidor ya está en ejecución, puede que el siguiente sea el problema. En algunas instalaciones de Windows Server, el agente de CloudWatch tarda más de 30 segundos en iniciarse.

De forma predeterminada, Windows Server solo permite 30 segundos para que los servicios se inicien, lo que hace que el agente genere un error similar al siguiente:

Para solucionarlo, aumente el valor de tiempo de espera del servicio. Para obtener más información, vea Un servicio no se inicia y los eventos 7000 y 7011 se registran en el registro de eventos de Windows.

¿Dónde están las métricas?

Si el agente de CloudWatch se ha estado ejecutando, pero no encuentra las métricas que ha recopilado en AWS Management Console o AWS CLI, confirme que está usando el espacio de nombres correcto. De forma predeterminada, el espacio de nombres de las métricas recopiladas por el agente es `CWAgent`. Puede personalizar este espacio de nombres mediante el campo `namespace` de la sección `metrics` del archivo de configuración del agente. Si no se muestran las métricas que espera, compruebe el archivo de configuración para confirmar el espacio de nombres que está usando.

Al descargar por primera vez el paquete del agente de CloudWatch, el archivo de configuración del agente es `amazon-cloudwatch-agent.json`. Este archivo se encuentra en el directorio donde ha ejecutado el asistente de configuración o puede que lo haya movido a otro directorio. Si utiliza el asistente de configuración, el resultado del archivo de configuración del agente del asistente se denomina `config.json`. Para obtener más información sobre el archivo de configuración, incluido el campo `namespace`, consulte Archivo de configuración del agente de CloudWatch: sección de métricas.

El agente de CloudWatch tarda mucho en ejecutarse en un contenedor o registra un error de límite de saltos

Cuando ejecuta el agente de CloudWatch como un servicio de contenedores y desea agregar las dimensiones de métricas de Amazon EC2 a todas las métricas recopiladas por el agente, es posible que vea los siguientes errores en la versión v1.247354.0 del agente:

```
2022-06-07T03:36:11Z E! [processors.ec2tagger] ec2tagger: Unable to retrieve Instance
 Metadata Tags. This plugin must only be used on an EC2 instance.
2022-06-07T03:36:11Z E! [processors.ec2tagger] ec2tagger: Please increase hop limit
 to 2 by following this document https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/
configuring-instance-metadata-options.html#configuring-IMDS-existing-instances.
2022-06-07T03:36:11Z E! [telegraf] Error running agent: could not initialize processor
 ec2tagger: EC2MetadataRequestError: failed to get EC2 instance identity document
caused by: EC2MetadataError: failed to make EC2Metadata request
```

```
        status code: 401, request id:
caused by:
```

Es posible que aparezca este error si el agente intenta obtener metadatos de IMDSv2 dentro de un contenedor sin un límite de saltos adecuado. En las versiones del agente anteriores a la v1.247354.0, puede experimentar este problema sin ver el mensaje de registro.

Para solucionarlo, aumente el límite de saltos a 2 siguiendo las instrucciones que figuran en Configure the instance metadata options (Configurar las opciones de metadatos de instancia).

He actualizado la configuración del agente pero no puedo ver las métricas o los registros nuevos en la consola de CloudWatch

Si actualiza el archivo de configuración del agente de CloudWatch, la próxima vez que inicie el agente, deberá utilizar la opción **`fetch-config`**. Por ejemplo, si ha almacenado el archivo actualizado en el equipo local, escriba el siguiente comando:

```
sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-config -s -m ec2 -c file:configuration-file-path
```

Archivos y ubicaciones del agente de CloudWatch

En la siguiente tabla se muestran los archivos que ha instalado y que utiliza el agente de CloudWatch, junto con las ubicaciones en servidores que ejecutan Linux o Windows Server.

Archivos	Ubicación en Linux	Ubicación en Windows Server
El script de control que controla el inicio, la parada y el reinicio del agente.	`/opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl` o `/usr/bin/amazon-cloudwatch-agent-ctl`	`$Env:ProgramFiles\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent-ctl.ps1`
El archivo de log en el que escribe el agente. Es posible que tenga que adjuntar esta	`/opt/aws/amazon-cloudwatch-agent/logs/amazon-cloudwatch-agent.log` o `/var/`	`$Env:ProgramData\Amazon\AmazonCloudWatchAgent\Logs\ama`

Archivos	Ubicación en Linux	Ubicación en Windows Server
información cuando se ponga en contacto con AWS Support.	`log/amazon/amazon-cloudwatch-agent/amazon-cloudwatch-agent.log`	`zon-cloudwatch-agent.log`
Archivo de validación de la configuración del agente.	`/opt/aws/amazon-cloudwatch-agent/logs/configuration-validation.log` o `/var/log/amazon/amazon-cloudwatch-agent/configuration-validation.log`	`$Env:ProgramData\Amazon\AmazonCloudWatchAgent\Logs\configuration-validation.log`
El archivo JSON que se utiliza para configurar el agente, inmediatamente después de que lo cree el asistente. Para obtener más información, consulte Cree el archivo de configuración del agente de CloudWatch.	`/opt/aws/amazon-cloudwatch-agent/bin/config.json`	`$Env:ProgramFiles\Amazon\AmazonCloudWatchAgent\config.json`
El archivo JSON que se utiliza para configurar el agente si este archivo de configuración se ha descargado desde el almacén de parámetros.	`/opt/aws/amazon-cloudwatch-agent/etc/amazon-cloudwatch-agent.json` o `/etc/amazon/amazon-cloudwatch-agent/amazon-cloudwatch-agent.json`	`$Env:ProgramData\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent.json`

Archivos	Ubicación en Linux	Ubicación en Windows Server
El archivo TOML que se ha utilizado para especificar la información sobre la región y las credenciales que va a utilizar el agente, que anularán los valores predeterminados del sistema.	`/opt/aws/amazon-cloudwatch-agent/etc/common-config.toml` o `/etc/amazon/amazon-cloudwatch-agent/common-config.toml`	`$Env:ProgramData\Amazon\AmazonCloudWatchAgent\common-config.toml`
Archivo TOML que contiene el contenido convertido del archivo de configuración JSON. El script de `amazon-cloudwatch-agent-ctl` genera este archivo. Los usuarios no deben modificar este archivo directamente. Puede resultar útil para comprobar que la traducción de JSON a TOML se ha realizado correctamente.	`/opt/aws/amazon-cloudwatch-agent/etc/amazon-cloudwatch-agent.toml` o `/etc/amazon/amazon-cloudwatch-agent/amazon-cloudwatch-agent.toml`	`$Env:ProgramData\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent.toml`
Archivo YAML que contiene el contenido convertido del archivo de configuración JSON. El script de `amazon-cloudwatch-agent-ctl` genera este archivo. Los usuarios no deben modificar este archivo directamente. Puede resultar útil para comprobar que la traducción de JSON a TOML se ha realizado correctamente.	`/opt/aws/amazon-cloudwatch-agent/etc/amazon-cloudwatch-agent.yaml or /etc/amazon/amazon-cloudwatch-agent/amazon-cloudwatch-agent.yaml`	`$Env:ProgramData\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent.yaml`

Búsqueda de información sobre las versiones del agente de CloudWatch

Para encontrar el número de versión del agente de CloudWatch en un servidor Linux, ingrese el siguiente comando:

```
sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a status
```

Para encontrar el número de versión del agente de CloudWatch en Windows Server, ingrese el siguiente comando:

```
& $Env:ProgramFiles\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent-ctl.ps1 -m ec2
 -a status
```

> **Note**
>
> El uso de este comando es la forma correcta de encontrar la versión del agente de CloudWatch. Si utiliza Programas y características en el Panel de control, verá un número de versión incorrecto.

También puede descargar un archivo LÉAME sobre los últimos cambios que el agente ha realizado y un archivo que indique el número de versión que está disponible para la descarga. Estos archivos se encuentran en las siguientes ubicaciones:

- `https://amazoncloudwatch-agent.s3.amazonaws.com/info/latest/RELEASE_NOTES` o `https://amazoncloudwatch-agent-`*`region`*`.s3.`*`region`*`.amazonaws.com/info/latest/RELEASE_NOTES`
- `https://amazoncloudwatch-agent.s3.amazonaws.com/info/latest/CWAGENT_VERSION` o `https://amazoncloudwatch-agent-`*`region`*`.s3.`*`region`*`.amazonaws.com/info/latest/CWAGENT_VERSION`

Registros que el agente de CloudWatch ha generado

El agente genera un registro durante su ejecución. Este registro incluye la información de solución de errores. Este registro es el archivo `amazon-cloudwatch-agent.log`. Este archivo se encuentra en `/opt/aws/amazon-cloudwatch-agent/logs/amazon-cloudwatch-agent.log` en los

servidores Linux y en `$Env:ProgramData\Amazon\AmazonCloudWatchAgent\Logs\amazon-cloudwatch-agent.log` en los servidores con Windows Server.

Puede configurar el agente para que registre detalles adicionales en el archivo `amazon-cloudwatch-agent.log`. En el archivo de configuración del agente, en la sección `agent`, establezca el campo `debug` a `true`; a continuación, vuelva a configurar y reinicie el agente de CloudWatch. Para deshabilitar el registro de esta información adicional, establezca el campo `debug` en `false`. A continuación, vuelva a configurar y reinicie el agente. Para obtener más información, consulte Cree o edite de forma manual el archivo de configuración del agente de CloudWatch.

En las versiones 1.247350.0 y las posteriores del agente de CloudWatch, puede configurar opcionalmente el campo `aws_sdk_log_level` en la sección `agent` del archivo de configuración del agente a una o más de las siguientes opciones. Separe varias opciones con el character `|`.

- `LogDebug`
- `LogDebugWithSigning`
- `LogDebugWithHTTPBody`
- `LogDebugRequestRetries`
- `LogDebugWithEventStreamBody`

Para obtener más información sobre estas opciones, consulte LogLevelType.

Cierre y reinicio del agente de CloudWatch

Puede detener el agente de CloudWatch mediante AWS Systems Manager o la línea de comandos de forma manual.

Para detener el agente de CloudWatch mediante Run Command

1. Abra la consola de Administrador de sistemas en https://console.aws.amazon.com/systems-manager/.
2. En el panel de navegación, elija Ejecutar comando.

   -o bien-

   Si la página de inicio de AWS Systems Manager se abre, desplácese hacia abajo y elija Explore Run Command (Explorar Run Command).
3. Elija Run command (Ejecutar comando).

4. En la lista Command document, elija AmazonCloudWatch-ManageAgent.
5. En el área Targets (Destinos), elija la instancia donde ha instalado el agente de CloudWatch.
6. En la lista Action, elija stop.
7. Deje Optional Configuration Source (Origen de configuración opcional) y Optional Configuration Location (Ubicación de configuración opcional) en blanco.
8. Elija Ejecutar.

Para detener el agente de CloudWatch localmente mediante la línea de comandos

- En un servidor Linux, escriba lo siguiente:

```
sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -m ec2 -a
 stop
```

  En un servidor con Windows Server, escriba lo siguiente en PowerShell como administrador:

```
& $Env:ProgramFiles\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent-ctl.ps1 -m
 ec2 -a stop
```

Para reiniciar el agente, siga las instrucciones que se describen en Inicie el agente de CloudWatch.

Incrustar métricas en los registros

El formato de métrica integrado de CloudWatch le permite generar métricas personalizadas de forma asíncrona en forma de registros escritos en registros de CloudWatch. Puede integrar métricas personalizadas junto con datos de eventos de registro detallados, y CloudWatch automáticamente extrae las métricas personalizadas para que pueda visualizarlas y crear alarmas en función de ellas con el fin de detectar los incidentes en tiempo real. Además, los eventos de registro detallados que se asocian a las métricas extraídas se pueden consultar con CloudWatch Logs Insights para ofrecer información detallada acerca de las causas raíz de los eventos operativos.

El formato de métrica integrado le ayuda a generar métricas personalizadas accionables a partir de recursos efímeros como las funciones de Lambda y los contenedores. Ahora, mediante el uso del formato de métricas integradas para enviar registros desde estos recursos efímeros, puede crear métricas personalizadas con facilidad y sin tener que instrumentar o mantener un código independiente, a la vez que obtiene funciones analíticas potentes en sus datos de registro.

No es necesario realizar ninguna configuración para utilizar el formato de métrica integrado. Puede estructurar sus registros siguiendo la Especificación del formato de métrica integrado o generarlos mediante nuestras bibliotecas de cliente y enviarlos a registros de CloudWatch mediante la API PutLogEvents o el agente de CloudWatch.

Se incurre en cargos por la incorporación y el archivado de registros, así como por las métricas personalizadas que se generan. Para obtener más información, consulte Precios de Amazon CloudWatch.

> **Note**
>
> Tenga cuidado al configurar su extracción de métricas, ya que afecta a su uso de métricas personalizadas y a la factura correspondiente. Si crea métricas de forma involuntaria basadas en dimensiones de alta cardinalidad (como `requestId`), el formato de métricas integradas creará, por diseño, una métrica personalizada correspondiente a cada combinación de dimensiones única. Para obtener más información, consulte Dimensiones.

Temas

- Publicación de registros mediante el formato de métrica integrado
- Visualización de sus métricas y registros en la consola

- Configurar alarmas en las métricas creadas con el formato de métrica integrado

Publicación de registros mediante el formato de métrica integrado

Puede generar registros de formato de métricas integradas con los siguientes métodos:

- Generar y enviar los registros mediante las bibliotecas de cliente de código abierto.
- Genere manualmente los registros utilizando la especificación de formato de métrica integrado y, a continuación, utilice el agente de CloudWatch o la API PutLogEvents para enviar los registros.

Temas

- Crear registros en formato de métrica integrado utilizando las bibliotecas de cliente
- Especificación: Formato de métricas integradas
- Uso de la API PutLogEvents para enviar registros de formato de métricas integradas creados manualmente
- Uso del agente de CloudWatch para enviar registros de formato de métricas integradas
- Uso del formato de métricas integradas con Distro para OpenTelemetry de AWS.

Crear registros en formato de métrica integrado utilizando las bibliotecas de cliente

Amazon proporciona bibliotecas de cliente de código abierto que puede utilizar para crear registros de formato de métricas integradas. En la actualidad, esas bibliotecas se encuentran disponibles para los lenguajes de la siguiente lista. Puede encontrar ejemplos completos de diferentes configuraciones en nuestras bibliotecas de cliente, en /examples.

Las bibliotecas y las instrucciones sobre cómo usarlas se encuentran en GitHub. Utilice los siguientes enlaces.

- Node.Js

> **Note**
>
> Para Node.js, se requieren las versiones 4.1.1+, 3.0.2+ y 2.0.7+ para su uso con el formato de registro JSON de Lambda. El uso de versiones anteriores en estos entornos Lambda provocará la pérdida de las métricas.

Para obtener más información, consulte Acceso a los registros de Amazon CloudWatch para AWS Lambda.

- Python
- Java
- C#

Las bibliotecas cliente están diseñadas para funcionar de forma inmediata con el agente de CloudWatch. Los registros en formato de métrica integrado generados se envían al agente de CloudWatch, que, a continuación, los agrega y publica en registros de CloudWatch por usted.

Note

Cuando se utiliza Lambda, no se requiere ningún agente para enviar los registros a CloudWatch. Todo lo que se registre en STDOUT se envía a registros de CloudWatch a través del agente de registro de Lambda.

Especificación: Formato de métricas integradas

El formato de métricas integradas de CloudWatch es una especificación JSON que se utiliza para indicar a CloudWatch Logs que extraiga automáticamente los valores de métricas integradas en eventos de registro estructurados. Puede utilizar CloudWatch para graficar y crear alarmas en los valores de métricas extraídos.

Convenciones de especificación de formato de métricas integradas

Las palabras clave “DEBE”, “NO DEBE”, “OBLIGATORIO”, “DEBERÁ”, “NO DEBERÁ”, “DEBERÍA”, “NO DEBERÍA”, “RECOMENDADO”, “PUEDE QUE” y “OPCIONAL” de esta especificación de formato se interpretarán tal como se describe en Palabras clave RFC2119.

Los términos "JSON", "texto JSON", "valor JSON", "miembro", "elemento", "objeto", "matriz", "número", "cadena", "valor booleano", "true", "false" y "null" de esta especificación de formato se interpretarán tal como se define en Notación de objetos de JavaScript RFC8259.

> **Note**
>
> Si planea crear alarmas a partir de métricas creadas con un formato de métrica integrado, consulte Configurar alarmas en las métricas creadas con el formato de métrica integrado para obtener recomendaciones.

Estructura de los documentos con formato de métricas integradas

En esta sección se describe la estructura de un documento con formato de métricas integradas. Los documentos con formato de métricas integradas se definen en Notación de objetos de JavaScript RFC8259.

A menos que se indique lo contrario, los objetos definidos por esta especificación NO DEBEN contener miembros adicionales. Los miembros no reconocidos por esta especificación DEBEN pasarse por alto. Los miembros definidos en esta especificación distinguen entre mayúsculas y minúsculas.

El formato de métricas integradas está sujeto a los mismos límites que los eventos estándar de CloudWatch Logs y están limitados a un tamaño máximo de 256 KB.

Con el formato de métrica incorporado, puede hacer un seguimiento del procesamiento de sus registros de EMF por métricas publicadas en el espacio de nombres de `AWS/Logs` de su cuenta. Se pueden utilizar para hacer un seguimiento de la generación con errores de métricas a partir de EMF, así como para determinar si se producen errores debido al análisis o la validación. Para obtener más información, consulte Supervisión con métricas de CloudWatch.

Nodo Raíz

El mensaje LogEvent DEBE ser un objeto JSON válido sin datos adicionales al principio o al final de la cadena de mensajes LogEvent. Para obtener más información acerca de la estructura LogEvent, consulte InputLogEvent.

Los documentos con formato de métricas integradas DEBEN contener el siguiente miembro de nivel superior en el nodo raíz. Este es un objeto Objeto de metadatos .

```
{
 "_aws": {
    "CloudWatchMetrics": [ ... ]
  }
```

```
}
```

El nodo raíz DEBE contener todos los Miembros de destino miembros definidos por las referencias en el Objeto MetricDirective .

PUEDE QUE el nodo raíz contenga cualquier otro miembro que no esté incluido en los requisitos anteriores. Los valores de estos miembros DEBEN ser tipos JSON válidos.

Objeto de metadatos

El miembro `_aws` se puede utilizar para representar metadatos sobre la carga que informa a los servicios posteriores de cómo deben procesar el LogEvent. El valor DEBE ser un objeto y DEBE contener los siguientes miembros:

- CloudWatchMetrics: Matriz de Objeto MetricDirective que se utiliza para indicar a CloudWatch que extraiga métricas del nodo raíz del LogEvent.

```
{
  "_aws": {
    "CloudWatchMetrics": [ ... ]
  }
}
```

- Marca temporal: Número que representa la marca temporal que se utiliza para las métricas que se extraen del evento. Los valores DEBEN expresarse como el número de milisegundos después del 1 de enero de 1970 00:00:00 UTC.

```
{
  "_aws": {
    "Timestamp": 1559748430481
  }
}
```

Objeto MetricDirective

El objeto MetricDirective indica a los servicios posteriores que LogEvent contiene métricas que se extraerán y publicarán en CloudWatch. MetricDirectives DEBE contener los siguientes miembros:

- Espacio de nombres: cadena que representa el espacio de nombres de CloudWatch para la métrica.

- Dimensiones: Matriz DimensionSet .
- Métricas: matriz de objetos de MetricDefinition. Esta matriz NO DEBE contener más de 100 objetos MetricDefinition.

Matriz DimensionSet

Un DimensionSet es una matriz de cadenas que contiene las claves de dimensión que se aplicarán a todas las métricas del documento. Los valores de esta matriz también DEBEN ser miembros en el nodo raíz, denominado Miembros de destino

Un DimensionSet NO DEBE contener más de 30 claves de dimensión. Una DimensionSet PUEDE estar vacía.

El miembro de destino DEBE tener un valor de cadena. Este valor NO DEBE contener más de 1024 caracteres. El miembro de destino define una dimensión que se publicará como parte de la identidad de métrica. Cada DimensionSet que se utiliza crea una métrica nueva en CloudWatch. Para obtener más información acerca de las dimensiones, consulte Dimensión y Dimensiones.

```
{
 "_aws": {
   "CloudWatchMetrics": [
     {
       "Dimensions": [ [ "functionVersion" ] ],
       ...
     }
   ]
 },
 "functionVersion": "$LATEST"
}
```

> **Note**
>
> Tenga cuidado al configurar su extracción de métricas, ya que afecta a su uso de métricas personalizadas y a la factura correspondiente. Si crea métricas de forma involuntaria basadas en dimensiones de alta cardinalidad (como `requestId`), el formato de métricas integradas creará, por diseño, una métrica personalizada correspondiente a cada combinación de dimensiones única. Para obtener más información, consulte Dimensiones.

Objeto MetricDefinition

Una MetricDefinition es un objeto que DEBE contener el siguiente miembro:

- Nombre: cadena Valores de referencia a una métrica Miembros de destino. Los destinos de la métrica DEBEN ser un valor numérico o una matriz de valores numéricos.

PUEDE QUE un objeto MetricDefinition contenga el siguiente elemento:

- Unidad: valor de cadena OPCIONAL que representa la unidad de medida de la métrica correspondiente. Los valores DEBEN ser unidades métricas válidas de CloudWatch. Para obtener información acerca de las unidades válidas, consulte MetricDatum. Si no se proporciona un valor, el sistema presupone que se utiliza un valor predeterminado de NONE.
- StorageResolution: valor entero OPCIONAL que representa la resolución de almacenamiento de la métrica correspondiente. Si se establece en 1, se especifica esta métrica como una métrica de alta resolución, de modo que CloudWatch almacena la métrica con una resolución inferior a un minuto hasta un segundo. Si se establece en 60, se especifica esta métrica como resolución estándar, que CloudWatch almacena con una resolución de 1 minuto. Los valores DEBEN ser válidos para las resoluciones compatibles con CloudWatch, 1 o 60. Si no se proporciona un valor, el sistema supone que se utiliza un valor predeterminado de 60.

  Para obtener más información acerca de las métricas de alta resolución, consulte Métricas de alta resolución.

> **Note**
>
> Si planea crear alarmas a partir de métricas creadas con un formato de métrica integrado, consulte Configurar alarmas en las métricas creadas con el formato de métrica integrado para obtener recomendaciones.

```
{
  "_aws": {
    "CloudWatchMetrics": [
      {
        "Metrics": [
          {
            "Name": "Time",
```

```
            "Unit": "Milliseconds",
            "StorageResolution": 60
          }
        ],
        ...
      }
    ]
  },
  "Time": 1
}
```

Valores de referencia

Los valores de referencia son valores de cadena que hacen referencia a los miembros Miembros de destino en el nodo raíz. Estas referencias NO deberían confundirse con los punteros JSON descritos en RFC6901. Los valores de destino no se pueden anidar.

Miembros de destino

Los destinos válidos DEBEN ser miembros en el nodo raíz y no pueden ser objetos anidados. Por ejemplo, un valor _reference_ de `"A.a"` DEBE coincidir con el siguiente miembro:

```
{ "A.a" }
```

NO DEBE coincidir con el miembro anidado:

```
{ "A": { "a" } }
```

Los valores válidos de los miembros de destino dependen de lo que los hace referencia. El destino de la métrica DEBE ser un valor numérico o una matriz de valores numéricos. Los destinos de la métrica de matriz numérica NO DEBEN tener más de 100 miembros. Un destino de dimensión DEBE tener un valor de cadena.

Ejemplo de formato de métricas integradas y esquema JSON

A continuación, se muestra un ejemplo válido de formato de métricas integradas.

```
{
  "_aws": {
    "Timestamp": 1574109732004,
    "CloudWatchMetrics": [
```

```
      {
        "Namespace": "lambda-function-metrics",
        "Dimensions": [["functionVersion"]],
        "Metrics": [
          {
            "Name": "time",
            "Unit": "Milliseconds",
            "StorageResolution": 60
          }
        ]
      }
    ]
  },
  "functionVersion": "$LATEST",
  "time": 100,
  "requestId": "989ffbf8-9ace-4817-a57c-e4dd734019ee"
}
```

Puede utilizar el siguiente esquema para validar documentos con formato de métricas integradas.

```
{
    "type": "object",
    "title": "Root Node",
    "required": [
        "_aws"
    ],
    "properties": {
        "_aws": {
            "$id": "#/properties/_aws",
            "type": "object",
            "title": "Metadata",
            "required": [
                "Timestamp",
                "CloudWatchMetrics"
            ],
            "properties": {
                "Timestamp": {
                    "$id": "#/properties/_aws/properties/Timestamp",
                    "type": "integer",
                    "title": "The Timestamp Schema",
                    "examples": [
                        1565375354953
                    ]
```

```
                },
                "CloudWatchMetrics": {
                    "$id": "#/properties/_aws/properties/CloudWatchMetrics",
                    "type": "array",
                    "title": "MetricDirectives",
                    "items": {
                        "$id": "#/properties/_aws/properties/CloudWatchMetrics/items",
                        "type": "object",
                        "title": "MetricDirective",
                        "required": [
                            "Namespace",
                            "Dimensions",
                            "Metrics"
                        ],
                        "properties": {
                            "Namespace": {
                                "$id": "#/properties/_aws/properties/CloudWatchMetrics/
items/properties/Namespace",
                                "type": "string",
                                "title": "CloudWatch Metrics Namespace",
                                "examples": [
                                    "MyApp"
                                ],
                                "pattern": "^(.*)$",
                                "minLength": 1,
                                "maxLength": 1024
                            },
                            "Dimensions": {
                                "$id": "#/properties/_aws/properties/CloudWatchMetrics/
items/properties/Dimensions",
                                "type": "array",
                                "title": "The Dimensions Schema",
                                "minItems": 1,
                                "items": {
                                    "$id": "#/properties/_aws/properties/
CloudWatchMetrics/items/properties/Dimensions/items",
                                    "type": "array",
                                    "title": "DimensionSet",
                                    "minItems": 0,
                                    "maxItems": 30,
                                    "items": {
                                        "$id": "#/properties/_aws/properties/
CloudWatchMetrics/items/properties/Dimensions/items/items",
                                        "type": "string",
```

```
                                        "title": "DimensionReference",
                                        "examples": [
                                            "Operation"
                                        ],
                                        "pattern": "^(.*)$",
                                        "minLength": 1,
                                        "maxLength": 250
}
                                    }
                                },
                                "Metrics": {
                                    "$id": "#/properties/_aws/properties/CloudWatchMetrics/
items/properties/Metrics",
                                    "type": "array",
                                    "title": "MetricDefinitions",
                                    "items": {
                                        "$id": "#/properties/_aws/properties/
CloudWatchMetrics/items/properties/Metrics/items",
                                        "type": "object",
                                        "title": "MetricDefinition",
                                        "required": [
                                            "Name"
                                        ],
                                        "properties": {
                                            "Name": {
                                                "$id": "#/properties/_aws/properties/
CloudWatchMetrics/items/properties/Metrics/items/properties/Name",
                                                "type": "string",
                                                "title": "MetricName",
                                                "examples": [
                                                    "ProcessingLatency"
                                                ],
                                                "pattern": "^(.*)$",
                                                "minLength": 1,
                                                "maxLength": 1024
                                            },
                                            "Unit": {
                                                "$id": "#/properties/_aws/properties/
CloudWatchMetrics/items/properties/Metrics/items/properties/Unit",
                                                "type": "string",
                                                "title": "MetricUnit",
                                                "examples": [
                                                    "Milliseconds"
                                                ],
```

```
                                                "pattern": "^(Seconds|Microseconds|
Milliseconds|Bytes|Kilobytes|Megabytes|Gigabytes|Terabytes|Bits|Kilobits|Megabits|
Gigabits|Terabits|Percent|Count|Bytes\\/Second|Kilobytes\\/Second|Megabytes\\/Second|
Gigabytes\\/Second|Terabytes\\/Second|Bits\\/Second|Kilobits\\/Second|Megabits\\/
Second|Gigabits\\/Second|Terabits\\/Second|Count\\/Second|None)$"
                                            },
                                            "StorageResolution": {
                                                "$id": "#/properties/_aws/properties/
CloudWatchMetrics/items/properties/Metrics/items/properties/StorageResolution",
                                                "type": "integer",
                                                "title": "StorageResolution",
                                                "examples": [
                                                    60
                                                ]
                                            }
                                        }
                                    }
                                }
                            }
                        }
                    }
                }
            }
        }
    }
}
```

Uso de la API PutLogEvents para enviar registros de formato de métricas integradas creados manualmente

Puede enviar registros de formato de métricas integradas a CloudWatch Logs mediante la API PutLogEvents de CloudWatch Logs. Al llamar a PutLogEvents, debe incluir la siguiente cabecera HTTP para indicarle a CloudWatch Logs que se deberían extraer las métricas, pero que ya no es necesario.

```
x-amzn-logs-format: json/emf
```

A continuación, se muestra un ejemplo completo del uso de SDK para Java 2.x de AWS:

```
package org.example.basicapp;

import software.amazon.awssdk.regions.Region;
```

```
import software.amazon.awssdk.services.cloudwatchlogs.CloudWatchLogsClient;
import software.amazon.awssdk.services.cloudwatchlogs.model.DescribeLogStreamsRequest;
import software.amazon.awssdk.services.cloudwatchlogs.model.DescribeLogStreamsResponse;
import software.amazon.awssdk.services.cloudwatchlogs.model.InputLogEvent;
import software.amazon.awssdk.services.cloudwatchlogs.model.PutLogEventsRequest;

import java.util.Collections;

public class EmbeddedMetricsExample {
        public static void main(String[] args) {

                final String usage = "To run this example, supply a Region code (eg.
 us-east-1), log group, and stream name as command line arguments"
                                + "Ex: PutLogEvents <region-id> <log-group-name>
 <stream-name>";

                if (args.length != 3) {
                        System.out.println(usage);
                        System.exit(1);
                }

                String regionId = args[0];
                String logGroupName = args[1];
                String logStreamName = args[2];

                CloudWatchLogsClient logsClient =
 CloudWatchLogsClient.builder().region(Region.of(regionId)).build();

                // Build a JSON log using the EmbeddedMetricFormat.
                long timestamp = System.currentTimeMillis();
                String message = "{" +
                                "  \"_aws\": {" +
                                "    \"Timestamp\": " + timestamp  + "," +
                                "    \"CloudWatchMetrics\": [" +
                                "      {" +
                                "        \"Namespace\": \"MyApp\"," +
                                "        \"Dimensions\": [[\"Operation\"], [\"Operation
\", \"Cell\"]]," +
                                "        \"Metrics\": [{ \"Name\": \"ProcessingLatency
\", \"Unit\": \"Milliseconds\", \"StorageResolution\": 60 }]" +
                                "      }" +
                                "    ]" +
                                "  }," +
                                "  \"Operation\": \"Aggregator\"," +
```

```
                                "  \"Cell\": \"001\"," +
                                "  \"ProcessingLatency\": 100" +
                                "}";
                InputLogEvent inputLogEvent = InputLogEvent.builder()
                        .message(message)
                        .timestamp(timestamp)
                        .build();

                // Specify the request parameters.
                PutLogEventsRequest putLogEventsRequest = PutLogEventsRequest.builder()
                        .logEvents(Collections.singletonList(inputLogEvent))
                        .logGroupName(logGroupName)
                        .logStreamName(logStreamName)
                        .build();

                logsClient.putLogEvents(putLogEventsRequest);

                System.out.println("Successfully put CloudWatch log event");
        }

}
```

> **Note**
>
> Con el formato de métrica incorporado, puede hacer un seguimiento del procesamiento de sus registros de EMF por métricas publicadas en el espacio de nombres de `AWS/Logs` de su cuenta. Se pueden utilizar para hacer un seguimiento de la generación con errores de métricas a partir de EMF, así como para determinar si se producen errores debido al análisis o la validación. Para obtener más información, consulte Supervisión con métricas de CloudWatch.

Uso del agente de CloudWatch para enviar registros de formato de métricas integradas

Para utilizar este método, instale primero el agente de CloudWatch para los servicios desde los que desea enviar registros con formato de métricas integradas y, a continuación, puede comenzar a enviar los eventos.

La versión del agente de CloudWatch debe ser 1.230621.0 o una posterior.

> **Note**
>
> No es necesario que se instale el agente de CloudWatch para enviar registros desde las funciones de Lambda.
> Los tiempos de espera de las funciones Lambda no se administran automáticamente. Esto significa que si el tiempo de espera de su función se agota antes de que las métricas se vacíen, las métricas de esa invocación no se capturarán.

Installing the CloudWatch agent (Instalación del agente de CloudWatch)

Instale el agente de CloudWatch para cada servicio que deba enviar registros con formato de métricas integradas.

Instalación del agente de CloudWatch en EC2

Primero, instale el agente de CloudWatch en la instancia. Para obtener más información, consulte Instalación del agente de CloudWatch.

Una vez que haya instalado el agente, configúrelo para que escuche un puerto UDP o TCP para los registros de formato de métricas integradas. A continuación, se muestra un ejemplo de esta configuración que escucha el socket predeterminado `tcp:25888`. Para obtener más información acerca de la configuración del agente, consulte Cree o edite de forma manual el archivo de configuración del agente de CloudWatch.

```
{
  "logs": {
    "metrics_collected": {
      "emf": { }
    }
  }
}
```

Instalación del agente de CloudWatch en Amazon ECS

La manera más fácil de implementar el agente de CloudWatch en Amazon ECS es ejecutarlo como sidecar al definirlo en la misma definición de tarea que su aplicación.

Creación de un archivo de configuración del agente

Cree el archivo de configuración del agente de CloudWatch localmente. En este ejemplo, la ruta relativa del archivo será `amazon-cloudwatch-agent.json`.

Para obtener más información acerca de la configuración del agente, consulte Cree o edite de forma manual el archivo de configuración del agente de CloudWatch.

```
{
  "logs": {
    "metrics_collected": {
      "emf": { }
    }
  }
}
```

Envío de la configuración al almacén de parámetros SSM

Escriba el siguiente comando para publicar el archivo de configuración del agente de CloudWatch al almacén de parámetros de AWS Systems Manager (SSM).

```
aws ssm put-parameter \
    --name "cwagentconfig" \
    --type "String" \
    --value "`cat amazon-cloudwatch-agent.json`" \
    --region "{{region}}"
```

Configuración de la definición de la tarea

Configure la definición de tarea para utilizar el agente de CloudWatch y exponer el puerto TCP o UDP. La definición de tarea de ejemplo que debe utilizar depende del modo de red.

Observe que `webapp` especifica la variable de entorno `AWS_EMF_AGENT_ENDPOINT`. La biblioteca la utiliza y debe apuntar al punto de enlace que escucha el agente. Además, el `cwagent` especifica el `CW_CONFIG_CONTENT` como parámetro “valueFrom” que apunta a la configuración de SSM que creó en el paso anterior.

Esta sección contiene un ejemplo para el modo puente y un ejemplo para el modo host o awsvpc. Para obtener más ejemplos de cómo se puede configurar el agente de CloudWatch en Amazon ECS, consulte el Github samples repository (Repositorio de muestras de Github)

A continuación, se muestra un ejemplo del modo puente. Cuando se habilitan redes en modo puente, el agente debe enlazarse con la aplicación mediante el parámetro `links` y debe hacerse referencia al mismo con el nombre del contenedor.

```
{
  "containerDefinitions": [
          {
              "name": "webapp",
              "links": [ "cwagent" ],
              "image": "my-org/web-app:latest",
              "memory": 256,
              "cpu": 256,
              "environment": [{
                "name": "AWS_EMF_AGENT_ENDPOINT",
                "value": "tcp://cwagent:25888"
              }],
          },
          {
              "name": "cwagent",
              "mountPoints": [],
              "image": "public.ecr.aws/cloudwatch-agent/cloudwatch-agent:latest",
              "memory": 256,
              "cpu": 256,
              "portMappings": [{
                  "protocol": "tcp",
                  "containerPort": 25888
              }],
              "environment": [{
                "name": "CW_CONFIG_CONTENT",
                "valueFrom": "cwagentconfig"
              }],
          }
      ],
}
```

A continuación, se muestra un ejemplo del modo de host o el modo awsvpc. Al ejecutarse en estos modos de red, se puede hacer referencia al agente a través de `localhost`.

```
{
  "containerDefinitions": [
          {
              "name": "webapp",
```

```
            "image": "my-org/web-app:latest",
            "memory": 256,
            "cpu": 256,
            "environment": [{
              "name": "AWS_EMF_AGENT_ENDPOINT",
              "value": "tcp://127.0.0.1:25888"
            }],
        },
        {
            "name": "cwagent",
            "mountPoints": [],
            "image": "public.ecr.aws/cloudwatch-agent/cloudwatch-agent:latest",
            "memory": 256,
            "cpu": 256,
            "portMappings": [{
                "protocol": "tcp",
                "containerPort": 25888
            }],
            "environment": [{
              "name": "CW_CONFIG_CONTENT",
              "valueFrom": "cwagentconfig"
            }],
        }
    ],
}
```

Note

En el modo awsvpc, debe proporcionar una dirección IP pública a la VPC (solo para Fargate), configurar una gateway NAT o configurar un punto de enlace de la VPC de CloudWatch Logs. Para obtener más información acerca de cómo configurar una NAT, consulte Gateways NAT. Para obtener más información acerca de cómo se configura un punto de enlace de la VPC de CloudWatch Logs, consulte Using CloudWatch Logs with Interface VPC Endpoints (Uso de CloudWatch Logs con los puntos de enlace de la VPC de tipo interfaz).
A continuación, se muestra un ejemplo de cómo asignar una dirección IP pública a una tarea que utiliza el tipo de lanzamiento de Fargate.

```
aws ecs run-task \
--cluster {{cluster-name}} \
--task-definition cwagent-fargate \
--region {{region}} \
--launch-type FARGATE \
```

```
--network-configuration
 "awsvpcConfiguration={subnets=[{{subnetId}}],securityGroups=[{{sgId}}],assignPublicIp=ENA
```

Seguro de permisos

Asegúrese de que el rol de IAM que ejecuta las tareas tiene permiso para leer datos del almacén de parámetros SSM. Puede añadir este permiso asociando la política AmazonSSMReadOnlyAccess. Para ello, introduzca el siguiente comando.

```
aws iam attach-role-policy --policy-arn arn:aws:iam::aws:policy/AmazonSSMReadOnlyAccess
 \
--role-name CWAgentECSExecutionRole
```

Instalación del agente de CloudWatch en Amazon EKS

Algunas partes de este proceso se pueden omitir si ya ha instalado CloudWatch Container Insights en este clúster.

Permisos

Si aún no ha instalado Container Insights, asegúrese primero de que los nodos de Amazon EKS tienen los permisos de IAM adecuados. Deben tener la CloudWatchAgentServerPolicy asociada. Para obtener más información, consulte Verificación de los requisitos previos de .

Creación de ConfigMap

Cree un ConfigMap para el agente. El ConfigMap también indica al agente que escuche un puerto TCP o UDP. Utilice el siguiente ConfigMap.

```
# cwagent-emf-configmap.yaml
apiVersion: v1
data:
  # Any changes here must not break the JSON format
  cwagentconfig.json: |
    {
      "agent": {
        "omit_hostname": true
      },
      "logs": {
        "metrics_collected": {
          "emf": { }
```

```
        }
      }
    }
kind: ConfigMap
metadata:
  name: cwagentemfconfig
  namespace: default
```

Si ya ha instalado Container Insights, añada la siguiente línea `"emf": { }` a su ConfigMap existente.

Aplicación del ConfigMap

Escriba el siguiente comando para aplicar el ConfigMap.

```
kubectl apply -f cwagent-emf-configmap.yaml
```

Implementación del agente

Para implementar el agente de CloudWatch como un sidecar, agregue el agente a la definición del pod, como en el ejemplo siguiente.

```
apiVersion: v1
kind: Pod
metadata:
  name: myapp
  namespace: default
spec:
  containers:
    # Your container definitions go here
    - name: web-app
      image: my-org/web-app:latest
    # CloudWatch Agent configuration
    - name: cloudwatch-agent
      image: public.ecr.aws/cloudwatch-agent/cloudwatch-agent:latest
      imagePullPolicy: Always
      resources:
        limits:
          cpu: 200m
          memory: 100Mi
        requests:
          cpu: 200m
          memory: 100Mi
```

```
      volumeMounts:
        - name: cwagentconfig
          mountPath: /etc/cwagentconfig
      ports:
  # this should match the port configured in the ConfigMap
        - protocol: TCP
          hostPort: 25888
          containerPort: 25888
  volumes:
    - name: cwagentconfig
      configMap:
        name: cwagentemfconfig
```

Uso del agente de CloudWatch para enviar registros de formato de métricas integradas

Una vez que el agente de CloudWatch esté instalado y en ejecución, podrá enviar los registros con formato de métricas integradas a través de TCP o UDP. Hay dos requisitos al enviar los registros a través del agente:

- Los registros deben contener una `LogGroupName` clave que indique al agente qué grupo de registros utilizar.
- Cada evento de registro debe estar en una sola línea. En otras palabras, un evento de registro no puede contener el carácter de nueva línea (\n).

Los eventos de registro también deben seguir la especificación de formato de métricas integradas. Para obtener más información, consulte Especificación: Formato de métricas integradas .

Si planea crear alarmas a partir de métricas creadas con un formato de métrica integrado, consulte Configurar alarmas en las métricas creadas con el formato de métrica integrado para obtener recomendaciones.

A continuación, se muestra un ejemplo de envío manual de eventos de registro desde un shell Bash de Linux. En su lugar, puede utilizar las interfaces de socket UDP proporcionadas por el lenguaje de programación que elija.

```
echo '{"_aws":{"Timestamp":1574109732004,"LogGroupName":"Foo","CloudWatchMetrics":
[{"Namespace":"MyApp","Dimensions":[["Operation"]],"Metrics":
[{"Name":"ProcessingLatency","Unit":"Milliseconds","StorageResolution":60}]}]},"Operation":"Agg
 \
```

```
> /dev/udp/0.0.0.0/25888
```

> **Note**
>
> Con el formato de métrica incorporado, puede hacer un seguimiento del procesamiento de sus registros de EMF por métricas publicadas en el espacio de nombres de `AWS/Logs` de su cuenta. Se pueden utilizar para hacer un seguimiento de la generación con errores de métricas a partir de EMF, así como para determinar si se producen errores debido al análisis o la validación. Para obtener más información, consulte Supervisión con métricas de CloudWatch.

Uso del formato de métricas integradas con Distro para OpenTelemetry de AWS.

Puede utilizar el formato de métricas integradas como parte del proyecto OpenTelemetry. OpenTelemetry es una iniciativa de código abierto que elimina los límites y restricciones entre los formatos específicos del proveedor para el seguimiento, los registros y las métricas al ofrecer un único conjunto de especificaciones y de las API. Para obtener más información, consulte OpenTelemetry.

El uso de formato de métricas integradas con OpenTelemetry requiere dos componentes: un origen de datos compatible con OpenTelemetry y el recopilador de Distro for OpenTelemetry de AWS habilitado para su uso con registros de formato de métricas integradas de CloudWatch.

Tenemos redistribuciones preconfiguradas de los componentes de OpenTelemetry, que AWS mantiene para que la incorporación sea lo más fácil posible. Para obtener más información sobre el uso de OpenTelemetry con formato de métricas integradas, además de otros servicios de AWS, consulte AWS Distro for OpenTelemetry.

Para obtener información adicional sobre la compatibilidad y el uso de idiomas, consulte Observabilidad de AWS en Github.

Visualización de sus métricas y registros en la consola

Después de generar registros con formato de métricas integradas que extraen métricas, puede utilizar la consola de CloudWatch para ver las métricas. Las métricas integradas tienen las

dimensiones que especificó al generar los registros. Además, las métricas integradas que generó con las bibliotecas de cliente tienen las siguientes dimensiones predeterminadas:

- ServiceType
- ServiceName
- LogGroup

Para ver las métricas generadas a partir de los registros de formato de métricas integradas

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, seleccione Metrics (Métricas).
3. Seleccione un espacio de nombres que especificó para las métricas integradas cuando las generó. Si utilizó las bibliotecas de cliente para generar las métricas y no especificó un espacio de nombres, seleccione aws-embedded-metrics. Este es el espacio de nombres predeterminado de las métricas integradas generadas mediante las bibliotecas de cliente.
4. Seleccione una dimensión de métrica (por ejemplo, ServiceName).
5. La pestaña All metrics muestra todas las métricas para dicha dimensión en el espacio de nombres. Puede hacer lo siguiente:

    a. Para ordenar la tabla, utilice el encabezado de columna.

    b. Para representar gráficamente una métrica, active la casilla de verificación situada junto a ella. Para seleccionar todas las métricas, seleccione la casilla de verificación en la fila de encabezado de la tabla.

    c. Para filtrar por recurso, seleccione el ID de recurso y, a continuación, elija Add to search (Añadir a la búsqueda).

    d. Para filtrar por métrica, elija el nombre de la métrica y, a continuación, seleccione Add to search (Añadir a búsqueda).

Consulta de registros que utilizan CloudWatch Logs Insights

Puede consultar los eventos de registro detallados que están asociados a las métricas extraídas mediante CloudWatch Logs Insights para ofrecer información detallada acerca de las causas raíz de los eventos operativos. Una de las ventajas de extraer métricas de los registros es que puede filtrar estos más adelante por la métrica única (nombre de métrica más conjunto de dimensiones único) y

los valores de métrica, para obtener el contexto de los eventos que contribuyeron al valor de métrica agregado

Por ejemplo, para obtener un ID de rastro de X-Ray o un ID de solicitud integrada, podría ejecutar la siguiente consulta en CloudWatch Logs Insights.

```
filter Latency > 1000 and Operation = "Aggregator"
| fields RequestId, TraceId
```

También puede realizar la agregación en tiempo de consulta en claves de alta cardinalidad, como la búsqueda de los clientes afectados por un evento. En el siguiente ejemplo, se ilustra este caso.

```
filter Latency > 1000 and Operation = "Aggregator"
| stats count() by CustomerId
```

Para obtener más información, consulte Analizar datos de registro con CloudWatch Logs Insights

Configurar alarmas en las métricas creadas con el formato de métrica integrado

En general, la creación de alarmas en las métricas generadas mediante el formato de métrica integrado sigue el mismo patrón que la creación de alarmas en cualquier otra métrica. Para obtener más información, consulte Uso de las alarmas de Amazon CloudWatch.

La generación de métricas en formato de métrica integrado depende del flujo de publicación de registros, ya que registros de CloudWatch debe procesar los registros para transformarlos en métricas. Por eso, es importante que publique los registros de manera oportuna para que los puntos de datos de las métricas se creen dentro del período de tiempo en el que se evalúan las alarmas.

Si planea utilizar el formato de métrica integrado para enviar métricas de alta resolución y crear alarmas en estas métricas, le recomendamos que vacíe los registros en registros de CloudWatch en un intervalo de 5 segundos o menos para evitar introducir demoras adicionales que puedan provocar alarmas en datos parciales o faltantes. Si utiliza el agente de CloudWatch, puede ajustar el intervalo de descarga configurando el parámetro `force_flush_interval` en el archivo de configuración del agente de CloudWatch. Este valor se establece de forma predeterminada en 5 segundos.

Si utiliza Lambda en otras plataformas en las que no puede controlar el intervalo de vaciado de registros, considere la posibilidad de utilizar alarmas “M de N” para controlar la cantidad de puntos

de datos que se utilizan para emitir la alarma. Para obtener más información, consulte Evaluación de una alarma.

Servicios de AWS que publican métricas de CloudWatch

Los siguientes servicios de AWS publican métricas en CloudWatch. Para obtener más información acerca de estas métricas y dimensiones, consulte la documentación especificada.

Servicio	Espacio de nombres	Documentación
AWS Amplify	`AWS/AmplifyHosting`	Supervisión
Amazon API Gateway	`AWS/ApiGateway`	Monitorizar la ejecución de la API con Amazon CloudWatch
Amazon AppFlow	`AWS/AppFlow`	Supervisión de Amazon AppFlow con Amazon CloudWatch
AWS Application Migration Service	`AWS/MGN`	Monitorización de Application Migration Service con Amazon CloudWatch
AWS App Runner	`AWS/AppRunner`	Visualización de métricas de servicio de App Runner que se notifican a CloudWatch
AppStream 2.0	`AWS/AppStream`	Monitoreo de recursos de Amazon AppStream 2.0
AWS AppSync	`AWS/AppSync`	Métricas de CloudWatch
Amazon Athena	`AWS/Athena`	Supervisión de las consultas de Athena con métricas de CloudWatch
Amazon Aurora	`AWS/RDS`	Métricas de Amazon Aurora
AWS Backup	`AWS/Backup`	Monitorización de métricas de copia de seguridad de AWS con CloudWatch
Amazon Bedrock	`AWS/Bedrock`	Monitorización de Amazon Bedrock con Amazon CloudWatch

Servicio	Espacio de nombres	Documentación
AWS Billing and Cost Management	`AWS/Billing`	Monitorización de cargos con alertas y notificaciones
Amazon Braket	`AWS/Braket/By Device`	Monitorización de Amazon Braket con Amazon CloudWatch
AWS Certificate Manager	`AWS/CertificateManager`	Métricas de CloudWatch compatibles
Autoridad de certificación privada de AWS	`AWS/ACMPrivateCA`	Métricas de CloudWatch compatibles
AWS Chatbot	`AWS/Chatbot`	Monitorización de AWS Chatbot con Amazon CloudWatch
Amazon Chime	`AWS/ChimeVoiceConnector`	Monitorización de Amazon Chime con Amazon CloudWatch
Amazon Chime SDK	`AWS/ChimeSDK`	Métricas de servicios
AWS Client VPN	`AWS/ClientVPN`	Monitorización con Amazon CloudWatch
Amazon CloudFront	`AWS/CloudFront`	Monitorizar la actividad de CloudFront en CloudWatch
AWS CloudHSM	`AWS/CloudHSM`	Obtención de métricas de CloudWatch
Amazon CloudSearch	`AWS/CloudSearch`	Monitorización de un dominio Amazon CloudSearch con Amazon CloudWatch

Servicio	Espacio de nombres	Documentación
AWS CloudTrail	`AWS/CloudTrail`	Métricas de CloudWatch compatibles
Agente de CloudWatch	Un `CWAgent` o un espacio de nombres personalizado	Métricas que el agente de CloudWatch ha recopilado
Flujo métrico de CloudWatch	`AWS/CloudWatch/MetricStreams`	Supervisión del flujo métrico con las métricas de CloudWatch
CloudWatch RUM	`AWS/RUM`	Métricas de CloudWatch que puede recopilar con CloudWatch RUM
CloudWatch Synthetics	`CloudWatchSynthetics`	Métricas de CloudWatch que los valores controlados publican
Registros de Amazon CloudWatch	`AWS/Logs`	Monitorización del uso con métricas de CloudWatch
AWS CodeBuild	`AWS/CodeBuild`	Supervisar AWS CodeBuild
Revisor de Amazon CodeGuru		Monitorización de CodeGuru Reviewer con Amazon CloudWatch
Amazon Kendra		Monitorización de Amazon Kendra con Amazon CloudWatch
Amazon CodeWhisperer	`AWS/CodeWhisperer`	Monitorización de Amazon CodeWhisperer con Amazon CloudWatch

Servicio	Espacio de nombres	Documentación
Amazon Cognito	`AWS/Cognito`	Monitoring Amazon Cognito (Supervisión de Amazon Cognito)
Amazon Comprehend	`AWS/Comprehend`	Monitorización de Amazon Comprehendpuntos de enlace
AWS Config	`AWS/Config`	Métricas de uso y éxito de AWS Config
Amazon Connect	`AWS/Connect`	Monitorización de Amazon Connect en las métricas de Amazon CloudWatch
Amazon Data Lifecycle Manager	`AWS/DataLifecycleManager`	Monitoree sus políticas mediante Amazon CloudWatch
AWS DataSync	`AWS/DataSync`	Monitorización de la tarea
Amazon DataZone		Monitorización de Amazon DataZone con Amazon CloudWatch
Amazon DevOps Guru	`AWS/DevOps-Guru`	Monitorización de Amazon DevOps Guru con Amazon CloudWatch
AWS Database Migration Service	`AWS/DMS`	Monitoreo de tareas de AWS DMS
AWS Direct Connect	`AWS/DX`	Monitorización con Amazon CloudWatch
AWS Directory Service	`AWS/DirectoryService`	Utilice las métricas de Amazon CloudWatch para determinar cuándo añadir controladores de dominio

Servicio	Espacio de nombres	Documentación
Amazon DocumentDB	`AWS/DocDB`	Métricas de Amazon DocumentDB
Amazon DynamoDB	`AWS/DynamoDB`	Dimensiones y métricas de DynamoDB
DynamoDB Accelerator (DAX)	`AWS/DAX`	Visualización de dimensiones y métricas DAX
Amazon EC2	`AWS/EC2`	Monitorización de las instancias con CloudWatch
Amazon EC2 Elastic Graphics	`AWS/ElasticGPUs`	Uso de las métricas CloudWatch para monitoreo de Elastic Graphics
Flota de spot de Amazon EC2	`AWS/EC2Spot`	Métricas de CloudWatch para las flotas de spot
Amazon EC2 Auto Scaling	`AWS/AutoScaling`	Supervisión de los grupos de Auto Scaling y las instancias mediante CloudWatch
AWS Elastic Beanstalk	`AWS/ElasticBeanstalk`	Publicación de métricas personalizadas de un entorno en Amazon CloudWatch
Amazon Elastic Block Store	`AWS/EBS`	Métricas de Amazon CloudWatch para Amazon EBS
Amazon Elastic Container Registry	`AWS/ECR`	Métricas de repositorios de Amazon ECR
Amazon Elastic Container Service	`AWS/ECS`	Métricas de Amazon ECS CloudWatch

Servicio	Espacio de nombres	Documentación
Amazon ECS a través de Información de contenedores de CloudWatch	`ECS/ContainerInsights`	Métricas de Información de contenedores de Amazon ECS
Escalado automático de clústeres de Amazon ECS	`AWS/ECS/ManagedScaling`	Escalado automático de clústeres de Amazon ECS
AWS Elastic Disaster Recovery		Métricas de CloudWatch para DRS
Amazon Elastic File System	`AWS/EFS`	Supervisión con CloudWatch
Amazon Elastic Inference	`AWS/ElasticInference`	Uso de métricas de CloudWatch para supervisar Amazon Elastic Inference
Amazon EKS a través de Información de contenedores de CloudWatch	`ContainerInsights`	Métricas de Información de contenedores de Kubernetes y de Amazon EKS
Elastic Load Balancing	`AWS/ApplicationELB`	Métricas de CloudWatch para el Application Load Balancer
Elastic Load Balancing	`AWS/NetworkELB`	Métricas de CloudWatch para el Network Load Balancer

Servicio	Espacio de nombres	Documentación
Elastic Load Balancing	`AWS/GatewayELB`	Métricas de CloudWatch para el balanceador de carga de su puerta de enlace
Elastic Load Balancing	`AWS/ELB`	Métricas de CloudWatch para el Classic Load Balancer
Amazon Elastic Transcoder	`AWS/ElasticTranscoder`	Monitorización con Amazon CloudWatch
Amazon ElastiCache for Memcached	`AWS/ElastiCache`	Monitorización del uso con métricas de CloudWatch
Amazon ElastiCache for Redis	`AWS/ElastiCache`	Monitorización del uso con métricas de CloudWatch
Amazon OpenSearch Service	`AWS/ES`	Supervisión de métricas del clúster de OpenSearch con Amazon CloudWatch
Amazon EMR	`AWS/ElasticMapReduce`	Monitorización de métricas con CloudWatch
AWS Elemental MediaConnect	`AWS/MediaConnect`	Supervisión de MediaConnect con Amazon CloudWatch
AWS Elemental MediaConvert	`AWS/MediaConvert`	Uso de métricas de CloudWatch para ver métricas de recursos de AWS Elemental MediaConvert
AWS Elemental MediaLive	`AWS/MediaLive`	Monitoreo de actividad mediante métricas de Amazon CloudWatch

Servicio	Espacio de nombres	Documentación
AWS Elemental MediaPackage	`AWS/MediaPackage`	Monitoreo de AWS Elemental MediaPackage con métricas de Amazon CloudWatch
AWS Elemental MediaStore	`AWS/MediaStore`	Monitoreo de AWS Elemental MediaStore con métricas de Amazon CloudWatch
AWS Elemental MediaTailor	`AWS/MediaTailor`	Monitorización de AWS Elemental MediaTailor con Amazon CloudWatch
Amazon EventBridge	`AWS/Events`	Monitoreo de Amazon EventBridge
Amazon FinSpace		Registro y monitorización
Amazon Forecast		Métricas de CloudWatch para Amazon Forecast
Amazon Fraud Detector		Monitorización de Amazon Fraud Detector con Amazon CloudWatch
Amazon FSx for Lustre	`AWS/FSx`	Supervisión de Amazon FSx for Lustre
Amazon FSx for OpenZFS	`AWS/FSx`	Supervisión con Amazon CloudWatch
Amazon FSx for Windows File Server	`AWS/FSx`	Supervisión de Amazon FSx for Windows File Server
Amazon FSx for ONTAP de NetApp	`AWS/FSx`	Supervisión con Amazon CloudWatch

Servicio	Espacio de nombres	Documentación
Amazon FSx for OpenZFS	`AWS/FSx`	Supervisión con Amazon CloudWatch
Amazon GameLift	`AWS/GameLift`	Supervise Amazon GameLift con CloudWatch
AWS Global Accelerator	`AWS/GlobalAccelerator`	Uso de Amazon CloudWatch con AWS Global Accelerator
AWS Glue	`Glue`	Monitoreo de AWS Glue con CloudWatch Metrics
AWS Ground Station	`AWS/GroundStation`	Métricas que utilizan Amazon CloudWatch
AWS HealthLake	`AWS/HealthLake`	Monitoreo de HealthLake con CloudWatch
Amazon Inspector	`AWS/Inspector`	Supervisión de Amazon Inspector mediante CloudWatch
Amazon Interactive Video Service	`AWS/IVS`	Supervisión de Amazon IVS con Amazon CloudWatch
Chat de Amazon Interactive Video Service	`AWS/IVSChat`	Supervisión de Amazon IVS con Amazon CloudWatch
AWS IoT	`AWS/IoT`	Dimensiones y métricas de AWS IoT
AWS IoT Analytics	`AWS/IoTAnalytics`	Espacios de nombres, métricas y dimensiones
AWS IoT FleetWise	`AWS/IoTFleetWise`	Supervisión de AWS IoT FleetWise con Amazon CloudWatch

Servicio	Espacio de nombres	Documentación
AWS IoT SiteWise	`AWS/IoTSiteWise`	Monitoreo de AWS IoT SiteWise con métricas de Amazon CloudWatch
AWS IoT TwinMaker	`AWS/IoTTwinMaker`	Supervisión de AWS IoT TwinMaker con métricas de Amazon CloudWatch
AWS IoT 1-Click		Monitorización de AWS IoT en un clic con Amazon CloudWatch
AWS Key Management Service	`AWS/KMS`	Supervisión con CloudWatch
Amazon Keyspaces (para Apache Cassandra)	`AWS/Cassandra`	Dimensiones y métricas de Amazon Keyspaces
Amazon Kendra		Monitorización de Amazon Kendra con Amazon CloudWatch
Amazon Managed Service para Apache Flink	`AWS/KinesisAnalytics`	Servicio gestionado para aplicaciones Apache Flink para SQL: supervisión con CloudWatch Managed Service para Apache Flink: consulte las métricas y dimensiones de Amazon Managed Service para Apache Flink
Amazon Data Firehose	`AWS/Firehose`	Supervisión de Firehose mediante métricas de CloudWatch
Amazon Kinesis Data Streams	`AWS/Kinesis`	Supervisión de Amazon Kinesis Data Streams con Amazon CloudWatch
Amazon Kinesis Video Streams	`AWS/KinesisVideo`	Supervisión de métricas de Kinesis Video Streams con CloudWatch

Servicio	Espacio de nombres	Documentación
AWS Lambda	`AWS/Lambda`	Métricas de AWS Lambda
Amazon Lex	`AWS/Lex`	Monitoreo de Amazon Lex con Amazon CloudWatch

AWS License Manager	`AWSLicenseManager/licenseUsage`
	
	`AWS/LicenseManager/LinuxSubscriptions`
		Monitoreo del uso de licencias con Amazon CloudWatch
		
		Métricas de uso y alarmas de Amazon CloudWatch para suscripciones de Linux

Amazon Location Service	`AWS/Location`	Métricas de Amazon Location Service exportadas a Amazon CloudWatch
Amazon Lookout for Equipment	`AWS/lookoutequipment`	Monitorización de Lookout for Equipment con Amazon CloudWatch
Amazon Lookout for Metrics	`AWS/LookoutMetrics`	Monitoreo de Lookout for Metrics con Amazon CloudWatch
Amazon Lookout for Vision	`AWS/LookoutVision`	Monitorización de Lookout for Metrics con Amazon CloudWatch
AWS Mainframe Modernization		Monitorización de AWS Mainframe Modernization con Amazon CloudWatch
Amazon Machine Learning	`AWS/ML`	Supervisión de Amazon ML con las métricas de Amazon CloudWatch

Servicio	Espacio de nombres	Documentación
Amazon Managed Blockchain	`AWS/managedblockchain`	Utilice las métricas de nodos pares de Hyperledger Fabric en Amazon Managed Blockchain
Amazon Managed Service para Prometheus	`AWS/Prometheus`	Métricas de Amazon CloudWatch
Amazon Managed Streaming for Apache Kafka	`AWS/Kafka`	Supervisión de Amazon MSK con Amazon CloudWatch
Amazon Managed Streaming for Apache Kafka	`AWS/KafkaConnect`	Monitorización de MSK Connect
Amazon Managed Workflows para Apache Airflow	`AWS/MWAA`	Métricas de contenedores, colas y bases de datos para Amazon MWAA
Amazon MemoryDB for Redis	`AWS/MemoryDB`	Monitorización de métricas de CloudWatch
Amazon MQ	`AWS/AmazonMQ`	Supervisión de los agentes de Amazon MQ mediante Amazon CloudWatch
Amazon Neptune	`AWS/Neptune`	Monitoreo de Neptune con CloudWatch
AWS Network Firewall	`AWS/NetworkFirewall`	Métricas de AWS Network Firewall en Amazon CloudWatch

Servicio	Espacio de nombres	Documentación
Administrador de red de AWS	`AWS/NetworkManager`	Métricas de CloudWatch para recursos locales
Amazon Nimble Studio	`AWS/NimbleStudio`	Supervisión de Nimble Studio con Amazon CloudWatch
AWS HealthOmics	`AWS/Omics`	Monitorización de AWS HealthOmics con Amazon CloudWatch
AWS OpsWorks	`AWS/OpsWorks`	Supervisión de pilas mediante Amazon CloudWatch
AWS Outposts	`AWS/Outposts`	Métricas de CloudWatch para AWS Outposts
AWS Panorama	`AWS/PanoramaDeviceMetrics`	Supervisión de dispositivos y aplicaciones con Amazon CloudWatch
Amazon Personalize	`AWS/Personalize`	Métricas de CloudWatch para Amazon Personalize
Amazon Pinpoint	`AWS/Pinpoint`	Visualización de métricas de Amazon Pinpoint en CloudWatch
Amazon Polly	`AWS/Polly`	Integración de CloudWatch con Amazon Polly
AWS PrivateLink	`AWS/PrivateLinkEndpoints`	Métricas de CloudWatch para AWS PrivateLink
AWS PrivateLink	`AWS/PrivateLinkServices`	Métricas de CloudWatch para AWS PrivateLink

Servicio	Espacio de nombres	Documentación
AWS Private 5G	`AWS/Private5G`	Métricas de Amazon CloudWatch
Amazon QLDB	`AWS/QLDB`	Supervisión de datos en Amazon QuickSight
Amazon QuickSight	`AWS/QuickSight`	Supervisión con Amazon CloudWatch
Amazon Redshift	`AWS/Redshift`	Datos de rendimiento de Amazon Redshift
Amazon Relational Database Service	`AWS/RDS`	Monitorización de métricas de Amazon RDS con Amazon CloudWatch
Amazon Rekognition	`AWS/Rekognition`	Monitorización de Rekognition con Amazon CloudWatch
AWS re:Post Private	`AWS/rePostPrivate`	Monitoreo de AWS re:Post Private con Amazon CloudWatch
AWS RoboMaker	`AWS/Robomaker`	Supervisión de RoboMaker de AWS con Amazon CloudWatch
Amazon Route 53	`AWS/Route53`	Supervisión de Amazon Route 53
Controlador de recuperación de aplicaciones Route 53	`AWS/Route53RecoveryReadiness`	Uso de Amazon CloudWatch con el Controlador de recuperación de aplicaciones
Amazon SageMaker	`AWS/SageMaker`	Supervisión de SageMaker con CloudWatch

Servicio	Espacio de nombres	Documentación
Canalizaciones de creación de modelos de Amazon SageMaker	`AWS/SageMaker/ModelBuildingPipeline`	Métricas de canalizaciones de SageMaker
AWS Secrets Manager	`AWS/SecretsManager`	Supervisión de Secrets Manager con Amazon CloudWatch
Amazon Security Lake	`AWS/SecurityLake`	Métricas de CloudWatch para Amazon Security Lake
Service Catalog	`AWS/ServiceCatalog`	Catálogo de servicio de CloudWatch Metrics
AWS Shield Advanced	`AWS/DDoSProtection`	Supervisión con CloudWatch
Amazon Simple Email Service	`AWS/SES`	Recuperación de datos de eventos de Amazon SES desde CloudWatch
AWS SimSpace Weaver	`AWS/simspaceweaver`	Monitorización de AWS SimSpace Weaver con Amazon CloudWatch
Amazon Simple Notification Service	`AWS/SNS`	Monitorización de Amazon SNS con CloudWatch
Amazon Simple Queue Service	`AWS/SQS`	Monitoreo de colas de Amazon SQS con CloudWatch
Amazon S3	`AWS/S3`	Monitorización de métricas con Amazon CloudWatch
Almacenamiento de lente de S3	`AWS/S3/Storage-Lens`	Monitoreo de métricas de S3 Storage Lens en CloudWatch

Servicio	Espacio de nombres	Documentación
Amazon Simple Workflow Service	`AWS/SWF`	Métricas de Amazon SWF para CloudWatch
AWS Step Functions	`AWS/States`	Monitoreo de Step Functions con CloudWatch
AWS Storage Gateway	`AWS/StorageGateway`	Uso de métricas de Amazon CloudWatch
AWS Systems Manager Run Command	`AWS/SSM-RunCommand`	Monitorización de métricas de Run Command mediante CloudWatch
Amazon Textract	`AWS/Textract`	Métricas de CloudWatch para Amazon Textract
Amazon Timestream	`AWS/Timestream`	Dimensiones y métricas de Timestream
AWS Transfer for SFTP	`AWS/Transfer`	Métricas de CloudWatch de AWS SFTP
Amazon Transcribe	`AWS/Transcribe`	Monitorización de Amazon Transcribe con Amazon CloudWatch
Amazon Translate	`AWS/Translate`	Dimensiones y métricas de CloudWatch para Amazon Translate
AWS Trusted Advisor	`AWS/TrustedAdvisor`	Creación de alarmas de Trusted Advisor mediante CloudWatch
Amazon VPC	`AWS/NATGateway`	Supervisión de la puerta de enlace NAT con CloudWatch

Servicio	Espacio de nombres	Documentación
Amazon VPC	`AWS/TransitGateway`	Métricas de CloudWatch para Transit Gateways
Amazon VPC	`AWS/VPN`	Supervisión con CloudWatch
IP Address Manager (IPAM) de Amazon VPC	`AWS/IPAM`	Crear alarmas con Amazon CloudWatch
AWS WAF	`AWS/WAFV2` para recursos de AWS WAF `WAF` para recursos clásicos de AWS WAF	Supervisión con CloudWatch
Amazon WorkMail	`AWS/WorkMail`	Monitorización de Amazon WorkMail con Amazon CloudWatch
Amazon WorkSpaces	`AWS/WorkSpaces`	Monitorizar sus WorkSpaces mediante métricas de CloudWatch
Amazon WorkSpaces Web	`AWS/WorkSpacesWeb`	Monitoreo de Amazon WorkSpaces Web con Amazon CloudWatch

Métricas de uso de AWS

CloudWatch recopila métricas que realizan un rastreo del uso de algunos recursos de AWS y de las API. Estas métricas se publican en el espacio de nombres de `AWS/Usage`. Las métricas de uso de CloudWatch le permiten administrar el uso de forma proactiva mediante la visualización de métricas en la consola de CloudWatch, la creación de paneles personalizados, la detección de cambios en la actividad con la detección de anomalías de CloudWatch y la configuración de alarmas que le avisan cuando el uso se acerca a un umbral.

Algunos servicios de AWS integran estas métricas de uso con Service Quotas. Para estos servicios, puede usar CloudWatch para administrar el uso de las cuotas de servicio de la cuenta. Para obtener más información, consulte Visualización de las cuotas de servicio y configuración de alarmas.

Temas

Visualización de las cuotas de servicio y configuración de alarmas

Para algunos servicios de AWS, puede utilizar estas métricas para visualizar el uso actual del servicio en paneles y en gráficos de CloudWatch. Puede utilizar una función de cálculo de métricas de CloudWatch para mostrar las cuotas de servicio de esos recursos en los gráficos. También puede configurar alarmas que le avisen cuando su uso se acerque a una cuota de servicio. Para obtener más información acerca de las cuotas de servicio, consulte el artículo donde se explica qué son las cuotas de servicio en la Guía del usuario de Service Quotas.

Si ha iniciado sesión en una cuenta que está configurada como cuenta de monitorización en la observabilidad entre cuentas de CloudWatch, puede utilizar esa cuenta de monitorización para visualizar las Service Quotas y establecer alarmas para las métricas en las cuentas de origen que están vinculadas a esa cuenta de monitorización. Para obtener más información, consulte Observabilidad entre cuentas de CloudWatch.

actualmente, los siguientes servicios integran las métricas de uso con Service Quotas:

- AWS CloudHSM
- SDK de Amazon Chime

- Amazon CloudWatch
- Registros de Amazon CloudWatch
- Amazon DynamoDB
- Amazon EC2
- Amazon Elastic Container Registry
- Elastic Load Balancing
- AWS Fargate
- AWS Fault Injection Service
- AWS Interactive Video Service
- AWS Key Management Service
- Amazon Data Firehose
- Amazon Location Service
- Consulta de Amazon Managed Blockchain (AMB)
- AWS RoboMaker
- Amazon SageMaker

Para visualizar una cuota de servicio y, opcionalmente, configurar una alarma

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, seleccione Métricas.
3. En la pestaña All metrics (Todas las métricas), elija Usage (Uso) y, a continuación, By AWS Resource (Por recurso de AWS).

   Aparecerá la lista de métricas de uso de cuotas de servicio.
4. Active la casilla situada junto a una de las métricas.

   En el gráfico se muestra su uso actual de ese recurso de AWS.
5. Para añadir su cuota de servicio al gráfico, haga lo siguiente:

   a. Elija la pestaña Métricas diagramadas.

   b. Elija Math expression (Expresión matemática) y Start with an empty expression (Comenzar con una expresión vacía). En la nueva fila, bajo el título Details (Detalles), ingrese **`SERVICE_QUOTA(m1)`**.

Se añade una nueva línea al gráfico, mostrando la cuota de servicio del recurso representado en la métrica.

6. Para ver su uso actual como porcentaje de la cuota, añada una nueva expresión o cambie la expresión SERVICE_QUOTA actual. La nueva expresión que se va a utilizar es **`m1/SERVICE_QUOTA(m1)*100`**.
7. (Opcional) Para configurar una alarma que le notifique si se acerca a la cuota de servicio, haga lo siguiente:

   a. En la fila con **`m1/SERVICE_QUOTA(m1)*100`**, en Actions (Acciones), elija el ícono de alarma. Se parece a una campana.

      Aparecerá la página de creación de alarmas.

   b. En Conditions (Condiciones), asegúrese de que Threshold type (Tipo de umbral) es Static (Estático) y Whenever Expression1 is (Siempre que Expression1 sea) se establece en Greater (Mayor). En than (que), escriba **`80`**. Esto crea una alarma que pasa al estado ALARM cuando su uso supera el 80 % de la cuota.

   c. Seleccione Siguiente.

   d. En la página siguiente, seleccione un tema de Amazon SNS o cree uno nuevo y, a continuación, elija Next (Siguiente). El tema que seleccione se notifica cuando la alarma pasa al estado de ALARMA.

   e. En la página siguiente, escriba un nombre y una descripción para la alarma y, a continuación, elija Next (Siguiente).

   f. Elija Crear alarma.

Métricas de uso de las API de AWS

La mayoría de las API compatibles con registros de AWS CloudTrail también informan métricas de uso a CloudWatch. Las métricas de uso de las API en CloudWatch le permiten administrar proactivamente el uso de las API mediante la visualización de métricas en la consola de CloudWatch, la creación de paneles personalizados, la detección de cambios en la actividad con la detección de anomalías de CloudWatch y la configuración de alarmas que alertan cuando el uso se acerca a un umbral.

En la siguiente tabla se enumeran los servicios que informan de las métricas de uso de las API a CloudWatch y el valor que se va a utilizar para la dimensión del `Service` para ver las métricas de uso de ese servicio.

Servicio	El valor de la dimensión del **`Service`**
AWS Identity and Access Management Access Analyzer	`Access Analyzer`
AWS Account Management	`Account Management`
Alexa for Business	`A4B`
Amazon API Gateway	`API Gateway`
AWS App Mesh	`App Mesh`
AWS AppConfig	`AWS AppConfig`
Amazon AppFlow	`AppFlow`
Application Auto Scaling	`Application Auto Scaling`
Application Discovery Service	`Application Discovery Service`
Amazon AppStream	`AppStream`
AppStream 2.0 Image Builder	`Image Builder`
Amazon Athena	`Athena`
AWS Audit Manager	`Audit Manager`
AWS Backup	`Backup`
AWS Batch	`Batch`
Amazon Braket	`Braket`
AWS Budgets	`Budgets`

Servicio	El valor de la dimensión del **`Service`**
AWS Certificate Manager	`Certificate Manager`
SDK de Amazon Chime	`ChimeSDK`
Amazon Cloud Directory	`Cloud Directory`
AWS Cloud Map	`Cloud Map`
AWS CloudFormation	`CloudFormation`
AWS CloudHSM	`CloudHSM`
Amazon CloudSearch	`CloudSearch`
AWS CloudShell	`CloudShell`
AWS CloudTrail	`CloudTrail`
Amazon CloudWatch	`CloudWatch`
Registros de Amazon CloudWatch	`Logs`
Información de aplicaciones de Amazon CloudWatch	`CloudWatch Application Insights`
AWS CodeBuild	`CodeBuild`
AWS CodeCommit	`CodeCommit`
Generador de perfiles de Amazon CodeGuru	`CodeGuru Profiler`
AWS CodePipeline	`CodePipeline`
AWS CodeStar	`CodeStar`
Notificaciones de AWS CodeStar	`CodeStar Notifications`
Conexiones de AWS CodeStar	`CodeStar Connections`
Grupo de identidades de Amazon Cognito	`Cognito Identity Pools`

Servicio	El valor de la dimensión del **`Service`**
Amazon Cognito Sync	`Cognito Sync`
Amazon Comprehend	`Comprehend`
Amazon Comprehend Medical	`Comprehend Medical`
AWS Compute Optimizer	`ComputeOptimzier`
Amazon Connect	`Connect`
Perfiles de clientes de Amazon Connect	`Customer Profiles`
Informes de uso y costo de AWS	`Cost and Usage Report`
AWS Cost Explorer	`Cost Explorer`
AWS Data Exchange	`Data Exchange`
Administrador del ciclo de vida de los datos de AWS	`Data Lifecycle Manager`
AWS Database Migration Service	`Database Migration Service`
AWS DataSync	`DataSync`
AWS DeepLens	`AWS DeepLens`
Amazon Detective	`Detective`
Asesor de dispositivos	`Device Advisor`
AWS Direct Connect	`Direct Connect`
AWS Directory Service	`Directory Service`
DynamoDB Accelerator	`DynamoDBAccelerator`
Amazon EC2	`EC2`
Escalado automático de EC2	`EC2 Auto Scaling`

Servicio	El valor de la dimensión del **`Service`**
Amazon Elastic Container Registry	`ECR Public`
Amazon Elastic Container Service	`ECS`
Amazon Elastic File System	`EFS`
Amazon Elastic Kubernetes Service	`EKS`
AWS Elastic Beanstalk	`Elastic Beanstalk`
Amazon Elastic Inference	`Elastic Inference`
Elastic Load Balancing	`Elastic Load Balancing`
Amazon EMR	`EMR Containers`
AWS Firewall Manager	`Firewall Manager`
Amazon FSx	`FSx`
Amazon GameLift	`GameLift`
AWS Glue DataBrew	`DataBrew`
Amazon Managed Grafana	`Grafana`
AWS IoT Greengrass	`Greengrass`
AWS Ground Station	`Ground Station`
API y notificaciones de AWS Health	`AWS Health APIs And Notifications`
Amazon Interactive Video Service	`IVS`
AWS IoT Core	`IoT`
AWS IoT 1-Click	`IoT 1-Click`
AWS IoT Events	`IoT Events`

Servicio	El valor de la dimensión del **`Service`**
AWS IoT RoboRunner	`IoT RoboRunner`
AWS IoT SiteWise	`IoT Sitewise`
AWS IoT Wireless	`IoT Wireless`
Amazon Kendra	`Kendra`
Amazon Keyspaces (para Apache Cassandra)	`Keyspaces`
Amazon Managed Service para Apache Flink	`Kinesis Analytics`
Amazon Data Firehose	`Firehose`
Kinesis Video Streams	`Kinesis Video Streams`
AWS Key Management Service	`KMS`
AWS Lambda	`Lambda`
AWS Launch Wizard	`Launch Wizard`
Amazon Lex	`Amazon Lex`
Amazon Lightsail	`Lightsail`
Amazon Location Service	`Location`
Amazon Lookout for Vision	`Lookout for Vision`
Amazon Machine Learning	`Amazon Machine Learning`
Amazon Macie	`Macie`
Consulta de Amazon Managed Blockchain (AMB)	`Amazon Managed Blockchain Query`
AWS Managed Services	`AWS Managed Services`
AWS Marketplace Commerce Analytics	`Marketplace Analytics Service`

Servicio	El valor de la dimensión del **`Service`**
AWS Elemental MediaConnect	`MediaConnect`
AWS Elemental MediaConvert	`MediaConvert`
AWS Elemental MediaLive	`MediaLive`
AWS Elemental MediaStore	`Mediastore`
AWS Elemental MediaTailor	`MediaTailor`
AWS Mobile Hub	`Mobile Hub`
AWS Network Firewall	`Network Firewall`
AWS OpsWorks	`OpsWorks`
AWS OpsWorks para la administración de la configuración	`OPsWorks CM`
AWS Outposts	`Outposts`
AWS Organizations	`Organizations`
Amazon RDS Performance Insights	`Performance Insights`
Amazon Pinpoint	`Pinpoint`
AWS Private Certificate Authority	`Private Certificate Authority`
Amazon Managed Service para Prometheus	`Prometheus`
AWS Proton	`Proton`
Amazon Quantum Ledger Database (Amazon QLDB)	`QLDB`
Amazon RDS	`RDS`
Amazon Redshift	`Redshift Data API`

Servicio	El valor de la dimensión del **`Service`**
Amazon Rekognition	`Rekognition`
AWS Resource Access Manager	`Resource Access Manager`
AWS Resource Groups	`Resource Groups`
AWS Resource Groups Tagging API	`Resource Groups Tagging API`
AWS RoboMaker	`RoboMaker`
Amazon Route 53 Dominios	`Route 53 Domains`
Amazon Route 53 Resolver	`Route 53 Resolver`
Amazon S3	`S3`
Amazon S3 Glacier	`Amazon S3 Glacier`
Tiempo de ejecución de Amazon SageMaker	`Sagemaker`
Savings Plans	`Savings Plans`
AWS Secrets Manager	`Secrets Manager`
AWS Security Hub	`Security Hub`
AWS Server Migration Service	`AWS Server Migration Service`
AWS Service Catalog AppRegistry	`Service Catalog AppRegistry`
Service Quotas	`Service Quotas`
AWS Shield	`Shield`
AWS Signer	`Signer`
Amazon Simple Notification Service	`SNS`
Amazon Simple Email Service	`SES`

Servicio	El valor de la dimensión del **`Service`**
Amazon Simple Queue Service	`SQS`
Almacén de identidades	`Identity Store`
Storage Gateway	`Storage Gateway`
AWS Support	`Support`
Amazon Simple Workflow Service	`SWF`
Amazon Textract	`Textract`
AWS IoT Things Graph	`ThingsGraph`
Amazon Timestream	`Timestream`
Amazon Transcribe	`Transcribe`
Amazon Translate	`Translate`
Transcripción de streaming de Amazon Transcribe	`Transcribe Streaming`
AWS Transfer Family	`Transfer`
AWS WAF	`WAF`
Amazon WorkDocs	`Amazon WorkDocs`
Amazon WorkLink	`WorkLink`
Amazon WorkMail	`Amazon WorkMail`
Amazon WorkSpaces	`Workspaces`
AWS X-Ray	`X-Ray`

Algunos servicios también informan sobre métricas de uso para las API adicionales. Para ver si una API informa de las métricas de uso a CloudWatch, utilice la consola de CloudWatch para ver las métricas que ese servicio en el espacio de nombres de `AWS/Usage` notifica.

Para ver la lista de las API de un servicio que informan de métricas de uso a CloudWatch

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, seleccione Métricas.
3. En la pestaña All metrics (Todas las métricas), elija Usage (Uso) y, a continuación, By AWS Resource (Por recurso de AWS).
4. En el cuadro de búsqueda situado cerca de la lista de las métricas, introduzca el nombre del servicio. Las métricas se filtran por el servicio que ha introducido.

Métricas de uso de CloudWatch

CloudWatch recopila métricas que realizan un seguimiento del uso de algunos recursos de AWS. Estas métricas corresponden a las cuotas de servicio de AWS. El seguimiento de estas métricas puede ayudarlo a administrar sus cuotas de forma proactiva. Para obtener más información, consulte Visualización de las cuotas de servicio y configuración de alarmas.

Las métricas de uso de cuotas de servicio se encuentran en el espacio de nombres `AWS/Usage` y se recopilan cada minuto.

Actualmente, el único nombre de métrica de este espacio de nombres que CloudWatch publica es `CallCount`. Esta métrica se publica con las dimensiones `Resource`, `Service` y `Type`. La dimensión `Resource` especifica el nombre de la operación de API de la que se realiza el seguimiento. Por ejemplo, la métrica `CallCount` con las dimensiones `"Service": "CloudWatch"`, `"Type": "API"` y `"Resource": "PutMetricData"` indica el número de veces que se ha llamado a la operación de la API de CloudWatch en la cuenta.`PutMetricData`

La métrica `CallCount` no tiene una unidad especificada. La estadística más útil para la métrica es `SUM`, que representa el recuento total de operaciones para el periodo de 1 minuto.

Métricas

Métrica	Descripción
`CallCount`	El número de operaciones especificadas realizadas en su cuenta.

Dimensiones

Dimensión	Descripción
`Service`	El nombre del servicio de AWS que contiene el recurso. Para las métricas de uso de CloudWatch, el valor de esta dimensión es `CloudWatch` .
`Class`	La clase de recurso a la que se realiza el seguimiento. Las métricas de uso de la API de CloudWatch utilizan esta dimensión con un valor de `None`.
`Type`	El tipo de recurso al que se realiza el seguimiento. Actualmente, cuando la dimensión `Service` es `CloudWatch` , el único valor válido para `Type` es `API`.
`Resource`	El nombre de la operación de la API. Entre los valores válidos se incluyen los siguientes: `DeleteAlarms` , `DeleteDashboards` , `DescribeAlarmHistory` , `DescribeAlarms` , `GetDashboard` , `GetMetricData` , `GetMetricStatistics` , `ListMetrics` , `PutDashboard` y `PutMetricData`

Tutoriales de CloudWatch

Las siguientes situaciones ilustran los usos de Amazon CloudWatch. En la primera situación, utilice la consola de CloudWatch para crear una alarma de facturación que realiza un seguimiento del consumo de AWS y que le permite saber si ha superado un determinado umbral de gasto. En la segunda situación más avanzada, utilizará AWS Command Line Interface (AWS CLI) para publicar una métrica única para una aplicación hipotética denominada GetStarted.

Escenarios

- Monitorear los cargos estimados
- Publicar las métricas

Situación: Monitoreo de los cargos estimados utilizando CloudWatch

En esta situación, se crea una alarma de Amazon CloudWatch para monitorear los cargos estimados. Al habilitar el monitoreo de los cargos estimados para su cuenta de AWS, los cargos estimados se calculan y se envían varias veces al día a CloudWatch como datos de métricas.

Los datos de métricas de facturación se almacenan en la región de EE. UU. Este (Norte de Virginia) y reflejan cargos en todo el mundo. Estos datos incluyen los cargos estimados para todos los servicios de AWS que utilice, así como el total estimado de los cargos de AWS.

Puede optar por recibir alertas mediante email cuando los cargos superen un umbral determinado. CloudWatch activa las alertas, y los mensajes se envían mediante Amazon Simple Notification Service (Amazon SNS).

> **Note**
>
> Para obtener información sobre cómo analizar los cargos de CloudWatch que ya se le han facturado, consulte Facturación y costo de CloudWatch.

Tareas

- Paso 1: Habilite las alertas de facturación

Paso 1: Habilite las alertas de facturación

Antes de crear una alarma para los cargos estimados, debe habilitar las alertas de facturación, a fin de que pueda supervisar los cargos estimados de AWS y crear una alarma a través de los datos de las métricas de facturación. Después de habilitar las alertas de facturación, no puede deshabilitar la recopilación de datos, pero puede eliminar las alarmas de facturación que ha creado.

Después de habilitar las alertas de facturación por primera vez, se tardan unos 15 minutos antes de poder ver los datos de facturación y definir alertas de facturación.

Requisitos

- Debe iniciar sesión con las credenciales de usuario raíz o como usuario con permiso para ver la información de facturación.
- En el caso de las cuentas de facturación consolidada, los datos de facturación de cada cuenta vinculada pueden encontrarse iniciando sesión en la cuenta de pago. Puede consultar los datos de facturación de los cargos totales estimados y de los cargos estimados por servicio de cada cuenta vinculada, además de la cuenta consolidada.
- En una cuenta de facturación unificada, las métricas de la cuenta vinculada al miembro solo se capturan si la cuenta del pagador habilita la preferencia Recibir alertas de facturación. Si cambia qué cuenta es su cuenta de administración/pagador, debe activar las alertas de facturación en la nueva cuenta de administración/pagador.
- La cuenta no debe formar parte de la Red de socios de Amazon (APN) debido a que las métricas de facturación no se publican en CloudWatch para cuentas APN. Para obtener más información, consulte Red de socios de AWS.

Para habilitar la monitorización de los cargos estimados

1. Abra la consola de AWS Billing en https://console.aws.amazon.com/billing/.
2. En el panel de navegación, elija Billing preferences (Preferencias de facturación).
3. En Preferencias de alertas, seleccione Editar.

4. Elija Recibir alertas de facturación de CloudWatch.
5. Elija Save preferences.

Paso 2: Cree una alarma de facturación

> **Important**
>
> Antes de crear una alarma de facturación, debe configurar su región en Este de EE. UU. (Norte de Virginia). Los datos de métricas de facturación se almacenan en esta región y representan cargos en todo el mundo. También debe habilitar las alertas de facturación en su cuenta o en la cuenta de administración o pagador (si utiliza la facturación unificada). Para obtener más información, consulte Paso 1: Habilite las alertas de facturación.

En este procedimiento, se crea una alarma que envía una notificación cuando los cargos estimados de AWS superan un umbral definido.

Para crear una alarma mediante la consola de CloudWatch

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Alarms (Alarmas) y, a continuación, elija All Alarms (Todas las alarmas).
3. Elija Create alarm (Crear alarma).
4. Elija Select Metric (Seleccionar métrica). En Browse (Navegar), elija Billing (Facturación) y, a continuación, elija Total Estimated Charge (Cargo total estimado).

   > **Note**
   >
   > Si no ve la métrica Facturación o Cargo total estimado, habilite las alertas de facturación y cambie su región a Este de EE. UU. (Norte de Virginia). Para obtener más información, consulte Habilitación de alertas de facturación.

5. Seleccione la casilla de la métrica EstimatedCharges y, a continuación, elija Select metric (Seleccionar métrica).
6. En Statistic (Estadística), elija Maximum (Máximo).
7. En Period (Período), seleccione 6 hours (6 horas).

8. En Threshold type (Tipo de umbral), elija Static (Estático).
9. En Whenever EstimatedCharges is . . . (Siempre que EstimatedCharges sea...), elija Greater (Mayor).
10. Para entonces..., defina el valor que desea que haga activar la alarma. Por ejemplo, **200** USD.

    Los valores métricos de EstimatedCharges están expresados únicamente en dólares estadounidenses (USD) y Amazon Services LLC se encarga de la conversión de divisas. Para obtener más información, consulte ¿Qué es AWS Billing?.

    > **Note**
    >
    > Después de definir un valor de umbral, el gráfico de vista previa muestra los cargos estimados para el mes actual.

11. Seleccione Configuración adicional y haga lo siguiente:

    - En Datapoints to alarm (Puntos de datos para alarma), especifique 1 out of 1 (1 de 1).
    - En Missing data treatment (Tratamiento de datos faltantes), elija Treat missing data as missing (Tratar los datos que faltan como faltantes).

12. Elija Next (Siguiente).
13. En Notificación, asegúrese de seleccionar En alarma. A continuación, especifique el tema de Amazon SNS con el que se le notificará cuando su alarma se encuentre en el estado `ALARM`. El tema de Amazon SNS puede incluir su dirección de correo electrónico para que reciba un email cuando el importe de facturación supere el umbral que haya especificado.

    Puede seleccionar un tema de Amazon SNS existente, crear un tema de Amazon SNS nuevo o elegir un ARN de tema para notificar a otra cuenta. Si quiere que la alarma envíe varias notificaciones para el mismo estado de alarma o para estados de alarma diferentes, elija Add notification (Agregar notificación).

14. Elija Next (Siguiente).
15. En Name and description (Nombre y descripción), ingrese un nombre para su alarma.

    - (Opcional) Ingrese una descripción de la alarma.

16. Elija Next (Siguiente).
17. En Preview and create (Ver la vista previa y crear), asegúrese de que su configuración sea correcta y, a continuación, elija Create alarm (Crear alarma).

Paso 3: Compruebe el estado de la alarma

Ahora, compruebe el estado de la alarma de facturación que acaba de crear.

Para comprobar el estado de alarma

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. Si es necesario, cambie la región a EE. UU. Este (Norte de Virginia). Los datos de métricas de facturación se almacenan en esta región y reflejan cargos en todo el mundo.
3. En el panel de navegación, elija Alarms.
4. Active la casilla que hay junto a la alarma. Hasta que se confirme la suscripción, se muestra como "Pendiente de confirmación". Cuando se confirme la suscripción, actualice la consola para mostrar el estado actualizado.

Paso 4: Edite una alarma de facturación

Por ejemplo, quizás desee aumentar de 200 a 400 USD la cantidad de dinero que gasta en AWS cada mes. Puede editar la alarma de facturación existente y aumentar el importe monetario que se debe superar antes de activar la alarma.

Para editar una alarma de facturación

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. Si es necesario, cambie la región a EE. UU. Este (Norte de Virginia). Los datos de métricas de facturación se almacenan en esta región y reflejan cargos en todo el mundo.
3. En el panel de navegación, elija Alarms.
4. Seleccione la casilla de verificación junto a la alarma y elija Actions (Acciones), Modify (Modificar).
5. En Siempre que los cargos AWS totales del mes excedan. especifique el nuevo importe que debe superarse para activar la alarma y enviar una notificación por correo electrónico.
6. Elija Guardar cambios.

Paso 5: Elimine una alarma de facturación

Si ya no necesita la alarma de facturación, puede eliminarla.

Para eliminar una alarma de facturación

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. Si es necesario, cambie la región a EE. UU. Este (Norte de Virginia). Los datos de métricas de facturación se almacenan en esta región y reflejan cargos en todo el mundo.
3. En el panel de navegación, elija Alarms.
4. Seleccione la casilla de verificación junto a la alarma y elija Actions (Acciones), Delete (Eliminar).
5. Cuando se le indique que confirme, seleccione Yes, Delete (Sí, borrar).

Situación: Publicación de métricas en CloudWatch

En esta situación, utiliza AWS Command Line Interface (AWS CLI) para publicar una métrica única para una aplicación hipotética denominada GetStarted. Si no ha instalado y configurado todavía la AWS CLI, consulte Getting Set Up with the AWS Command Line Interface (Configuración inicial del AWS Command Line Interface) en la Guía del usuario.

Tareas

- Paso 1: Defina la configuración de datos
- Paso 2: Agregue métricas a CloudWatch
- Paso 3: Obtenga estadísticas de CloudWatch
- Paso 4: Visualice gráficos con la consola

Paso 1: Defina la configuración de datos

En esta situación, publica puntos de datos que realizan un seguimiento de la latencia de solicitudes para la aplicación. Elija nombres para su métrica y el espacio de nombres que tenga sentido. En este ejemplo, nombre la métrica RequestLatency y coloque todos los puntos de datos en el espacio de nombres GetStarted.

Puede publicar varios puntos de datos que representan colectivamente los datos de latencia de tres horas. Los datos sin procesar engloban 15 lecturas de latencia de solicitud distribuidas en tres horas. Cada lectura está en milisegundos:

- Hora uno: 87, 51, 125, 235
- Hora dos: 121, 113, 189, 65, 89

- Hora tres: 100, 47, 133, 98, 100, 328

Puede publicar datos en CloudWatch como puntos de datos únicos o como un conjunto de puntos de datos acumulados que se conoce como conjunto estadístico. Puede acumular métricas con una granularidad de tan solo un minuto. Puede publicar los puntos de datos acumulados en CloudWatch como un conjunto de estadísticas con cuatro claves predefinidas: `Sum`, `Minimum`, `Maximum` y `SampleCount`.

Publica los puntos de datos desde la hora uno como puntos de datos únicos. Para los datos de las horas dos y tres, acumulará los puntos de datos y publica un conjunto estadístico para cada hora. Los valores principales se muestran en la siguiente tabla.

Hora	Datos sin procesar	Sum	Mínimo	Máximo	Recuento de ejemplo
1	87				
1	51				
1	125				
1	235				
2	121, 113, 189, 65, 89	577	65	189	5
3	100, 47, 133, 98, 100, 328	806	47	328	6

Paso 2: Agregue métricas a CloudWatch

Después de haber definido la configuración de datos, está listo para agregar datos.

Para publicar puntos de datos en CloudWatch

1. En el símbolo del sistema, ejecute los siguientes comandos put-metric-data para agregar datos para la primera hora. Sustituya la marca temporal de ejemplo por una marca temporal que se encuentre dos horas en el pasado, en tiempo universal coordinado (UTC).

```
aws cloudwatch put-metric-data --metric-name RequestLatency --namespace GetStarted
 \
--timestamp 2016-10-14T20:30:00Z --value 87 --unit Milliseconds
aws cloudwatch put-metric-data --metric-name RequestLatency --namespace GetStarted
 \
--timestamp 2016-10-14T20:30:00Z --value 51 --unit Milliseconds
aws cloudwatch put-metric-data --metric-name RequestLatency --namespace GetStarted
 \
--timestamp 2016-10-14T20:30:00Z --value 125 --unit Milliseconds
aws cloudwatch put-metric-data --metric-name RequestLatency --namespace GetStarted
 \
--timestamp 2016-10-14T20:30:00Z --value 235 --unit Milliseconds
```

2. Agregue los datos para la segunda hora, utilizando una marca temporal que sea una hora después de la primera hora.

```
aws cloudwatch put-metric-data --metric-name RequestLatency --namespace GetStarted
 \
--timestamp 2016-10-14T21:30:00Z --statistic-values
 Sum=577,Minimum=65,Maximum=189,SampleCount=5 --unit Milliseconds
```

3. Agregue datos para la tercera hora, omitiendo la marca temporal al valor predeterminado para la hora actual.

```
aws cloudwatch put-metric-data --metric-name RequestLatency --namespace GetStarted
 \
--statistic-values Sum=806,Minimum=47,Maximum=328,SampleCount=6 --unit Milliseconds
```

Paso 3: Obtenga estadísticas de CloudWatch

Ahora que ha publicado métricas en CloudWatch, puede recuperar las estadísticas basadas en dichas métricas mediante el comando get-metric-statistics como se indica a continuación. Asegúrese de especificar `--start-time` y `--end-time` suficientemente lejos en el pasado para cubrir la primera marca temporal que ha publicado.

```
aws cloudwatch get-metric-statistics --namespace GetStarted --metric-name
 RequestLatency --statistics Average \
--start-time 2016-10-14T00:00:00Z --end-time 2016-10-15T00:00:00Z --period 60
```

A continuación, se muestra un ejemplo de la salida:

```
{
 "Datapoints": [],
 "Label": "Request:Latency"
}
```

Paso 4: Visualice gráficos con la consola

Una vez que haya publicado métricas en CloudWatch, puede utilizar la consola de CloudWatch para ver los gráficos estadísticos.

Para ver gráficos de sus estadísticas en la consola

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel Navigation, seleccione Metrics.
3. En la pestaña All metrics (Todas las métricas), en el campo de búsqueda, escriba RequestLatency y pulse Intro.
4. Seleccione la casilla de verificación de la métrica RequestLatency. Se muestra un gráfico de los datos de métricas en el panel superior.

Para obtener más información, consulte Representación gráfica de las métricas.

Uso de CloudWatch con SDK de AWS

Los kits de desarrollo de software (SDK) de AWS se encuentran disponibles en muchos lenguajes de programación populares. Cada SDK proporciona una API, ejemplos de código y documentación que facilitan a los desarrolladores la creación de aplicaciones en su lenguaje preferido.

Documentación de SDK	Ejemplos de código
AWS SDK for C++	Ejemplos de código de AWS SDK for C++
AWS CLI	Ejemplos de código de AWS CLI
AWS SDK for Go	Ejemplos de código de AWS SDK for Go
AWS SDK for Java	Ejemplos de código de AWS SDK for Java
AWS SDK for JavaScript	Ejemplos de código de AWS SDK for JavaScript
AWS SDK para Kotlin	Ejemplos de código de AWS SDK para Kotlin
AWS SDK for .NET	Ejemplos de código de AWS SDK for .NET
AWS SDK for PHP	Ejemplos de código de AWS SDK for PHP
AWS Tools for PowerShell	Ejemplos de código de Herramientas para PowerShell
AWS SDK for Python (Boto3)	Ejemplos de código de AWS SDK for Python (Boto3)
AWS SDK for Ruby	Ejemplos de código de AWS SDK for Ruby
AWS SDK para Rust	Ejemplos de código de AWS SDK para Rust
AWS SDK para SAP ABAP	Ejemplos de código de AWS SDK para SAP ABAP
AWS SDK para Swift	Ejemplos de código de AWS SDK para Swift

Para ver ejemplos específicos de CloudWatch, consulte Ejemplos de código para CloudWatch usando los AWS SDK.

Ejemplo de disponibilidad

¿No encuentra lo que necesita? Solicite un ejemplo de código a través del enlace de Enviar comentarios que se encuentra al final de esta página.

Ejemplos de código para CloudWatch usando los AWS SDK

Los siguientes ejemplos de código indican cómo utilizar CloudWatch con un kit de desarrollo de software (SDK) de AWS.

Las acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Mientras las acciones muestran cómo llamar a las funciones de servicio individuales, es posible ver las acciones en contexto en los escenarios relacionados y en los ejemplos entre servicios.

Los escenarios son ejemplos de código que muestran cómo llevar a cabo una tarea específica llamando a varias funciones dentro del mismo servicio.

Los ejemplos con varios servicios son aplicaciones de muestra que funcionan con varios Servicios de AWS.

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de CloudWatch con SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Introducción

Introducción a CloudWatch

En los siguientes ejemplos de código se muestra cómo empezar a utilizar CloudWatch.

.NET

AWS SDK for .NET

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
using Amazon.CloudWatch;
using Amazon.CloudWatch.Model;
using Microsoft.Extensions.DependencyInjection;
using Microsoft.Extensions.Hosting;
```

```
namespace CloudWatchActions;

public static class HelloCloudWatch
{
    static async Task Main(string[] args)
    {
        // Use the AWS .NET Core Setup package to set up dependency injection for
 the Amazon CloudWatch service.
        // Use your AWS profile name, or leave it blank to use the default
 profile.
        using var host = Host.CreateDefaultBuilder(args)
            .ConfigureServices((_, services) =>
                services.AddAWSService<IAmazonCloudWatch>()
            ).Build();

        // Now the client is available for injection.
        var cloudWatchClient =
 host.Services.GetRequiredService<IAmazonCloudWatch>();

        // You can use await and any of the async methods to get a response.
        var metricNamespace = "AWS/Billing";
        var response = await cloudWatchClient.ListMetricsAsync(new
 ListMetricsRequest
        {
            Namespace = metricNamespace
        });
        Console.WriteLine($"Hello Amazon CloudWatch! Following are some metrics
 available in the {metricNamespace} namespace:");
        Console.WriteLine();
        foreach (var metric in response.Metrics.Take(5))
        {
            Console.WriteLine($"\tMetric: {metric.MetricName}");
            Console.WriteLine($"\tNamespace: {metric.Namespace}");
            Console.WriteLine($"\tDimensions: {string.Join(", ",
 metric.Dimensions.Select(m => $"{m.Name}:{m.Value}"))}");
            Console.WriteLine();
        }
    }
}
```

- Para obtener información sobre las API, consulte ListMetrics en la Referencia de la API de AWS SDK for .NET.

Java

SDK para Java 2.x

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.cloudwatch.CloudWatchClient;
import software.amazon.awssdk.services.cloudwatch.model.CloudWatchException;
import software.amazon.awssdk.services.cloudwatch.model.ListMetricsRequest;
import software.amazon.awssdk.services.cloudwatch.paginators.ListMetricsIterable;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class HelloService {
    public static void main(String[] args) {
        final String usage = """

                Usage:
                  <namespace>\s

                Where:
                  namespace - The namespace to filter against (for example, AWS/
EC2).\s
                """;

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }
```

```
        String namespace = args[0];
        Region region = Region.US_EAST_1;
        CloudWatchClient cw = CloudWatchClient.builder()
                .region(region)
                .build();

        listMets(cw, namespace);
        cw.close();
    }

    public static void listMets(CloudWatchClient cw, String namespace) {
        try {
            ListMetricsRequest request = ListMetricsRequest.builder()
                    .namespace(namespace)
                    .build();

            ListMetricsIterable listRes = cw.listMetricsPaginator(request);
            listRes.stream()
                    .flatMap(r -> r.metrics().stream())
                    .forEach(metrics -> System.out.println(" Retrieved metric is:
 " + metrics.metricName()));

        } catch (CloudWatchException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- Para obtener información sobre las API, consulte ListMetrics en la Referencia de la API de AWS SDK for Java 2.x.

Kotlin

SDK para Kotlin

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
/**
Before running this Kotlin code example, set up your development environment,
including your credentials.

For more information, see the following documentation topic:
https://docs.aws.amazon.com/sdk-for-kotlin/latest/developer-guide/setup.html
 */
suspend fun main(args: Array<String>) {
    val usage = """
        Usage:
           <namespace>
        Where:
           namespace - The namespace to filter against (for example, AWS/EC2).
    """

    if (args.size != 1) {
        println(usage)
        exitProcess(0)
    }

    val namespace = args[0]
    listAllMets(namespace)
}

suspend fun listAllMets(namespaceVal: String?) {
    val request = ListMetricsRequest {
        namespace = namespaceVal
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        cwClient.listMetricsPaginated(request)
            .transform { it.metrics?.forEach { obj -> emit(obj) } }
            .collect { obj ->
                println("Name is ${obj.metricName}")
                println("Namespace is ${obj.namespace}")
            }
    }
}
```

- Para obtener información sobre la API, consulte ListMetrics en la Referencia de la API deAWS SDK para Kotlin.

Ejemplos de código

- Acciones para CloudWatch usando los AWS SDK
  - Uso de DeleteAlarms con un AWS SDK o la CLI
  - Uso de DeleteAnomalyDetector con un AWS SDK o la CLI
  - Uso de DeleteDashboards con un AWS SDK o la CLI
  - Uso de DescribeAlarmHistory con un AWS SDK o la CLI
  - Uso de DescribeAlarms con un AWS SDK o la CLI
  - Uso de DescribeAlarmsForMetric con un AWS SDK o la CLI
  - Uso de DescribeAnomalyDetectors con un AWS SDK o la CLI
  - Uso de DisableAlarmActions con un AWS SDK o la CLI
  - Uso de EnableAlarmActions con un AWS SDK o la CLI
  - Uso de GetDashboard con un AWS SDK o la CLI
  - Uso de GetMetricData con un AWS SDK o la CLI
  - Uso de GetMetricStatistics con un AWS SDK o la CLI
  - Uso de GetMetricWidgetImage con un AWS SDK o la CLI
  - Uso de ListDashboards con un AWS SDK o la CLI
  - Uso de ListMetrics con un AWS SDK o la CLI
  - Uso de PutAnomalyDetector con un AWS SDK o la CLI
  - Uso de PutDashboard con un AWS SDK o la CLI
  - Uso de PutMetricAlarm con un AWS SDK o la CLI
  - Uso de PutMetricData con un AWS SDK o la CLI
- Situaciones de CloudWatch usando los AWS SDK
  - Primeros pasos para usar las alarmas de CloudWatch mediante un SDK de AWS
  - Primeros pasos para usar las métricas de CloudWatch mediante un SDK de AWS
  - Administración de métricas y alarmas de CloudWatch mediante un SDK de AWS
- Ejemplos de servicios combinados para CloudWatch con AWS SDK
  - Supervisión del rendimiento de Amazon DynamoDB mediante AWS SDK

Acciones para CloudWatch usando los AWS SDK

Los siguientes ejemplos de código muestran cómo realizar acciones de CloudWatch individuales con SDK de AWS. Estos fragmentos llaman a la API de CloudWatch y son partes de código de programas más grandes que deben ejecutarse en contexto. En cada ejemplo se incluye un enlace a GitHub, con instrucciones de configuración y ejecución del código.

Los siguientes ejemplos incluyen solo las acciones que se utilizan con mayor frecuencia. Para ver una lista completa, consulte la Referencia de la API de Amazon CloudWatch.

Ejemplos

- Uso de DeleteAlarms con un AWS SDK o la CLI
- Uso de DeleteAnomalyDetector con un AWS SDK o la CLI
- Uso de DeleteDashboards con un AWS SDK o la CLI
- Uso de DescribeAlarmHistory con un AWS SDK o la CLI
- Uso de DescribeAlarms con un AWS SDK o la CLI
- Uso de DescribeAlarmsForMetric con un AWS SDK o la CLI
- Uso de DescribeAnomalyDetectors con un AWS SDK o la CLI
- Uso de DisableAlarmActions con un AWS SDK o la CLI
- Uso de EnableAlarmActions con un AWS SDK o la CLI
- Uso de GetDashboard con un AWS SDK o la CLI
- Uso de GetMetricData con un AWS SDK o la CLI
- Uso de GetMetricStatistics con un AWS SDK o la CLI
- Uso de GetMetricWidgetImage con un AWS SDK o la CLI
- Uso de ListDashboards con un AWS SDK o la CLI
- Uso de ListMetrics con un AWS SDK o la CLI
- Uso de PutAnomalyDetector con un AWS SDK o la CLI
- Uso de PutDashboard con un AWS SDK o la CLI
- Uso de PutMetricAlarm con un AWS SDK o la CLI
- Uso de PutMetricData con un AWS SDK o la CLI

Uso de **`DeleteAlarms`** con un AWS SDK o la CLI

Los siguientes ejemplos de código muestran cómo utilizar `DeleteAlarms`.

Los ejemplos de acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Puede ver esta acción en su contexto en los siguientes ejemplos de código:

- Primeros pasos para usar alarmas
- Primeros pasos para usar las métricas, los paneles y las alarmas
- Gestionar métricas y alarmas

.NET

AWS SDK for .NET

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
/// <summary>
/// Delete a list of alarms from CloudWatch.
/// </summary>
/// <param name="alarmNames">A list of names of alarms to delete.</param>
/// <returns>True if successful.</returns>
public async Task<bool> DeleteAlarms(List<string> alarmNames)
{
    var deleteAlarmsResult = await _amazonCloudWatch.DeleteAlarmsAsync(
        new DeleteAlarmsRequest()
        {
            AlarmNames = alarmNames
        });

    return deleteAlarmsResult.HttpStatusCode == HttpStatusCode.OK;
}
```

- Para obtener información sobre las API, consulte DeleteAlarms en la Referencia de la API de AWS SDK for .NET.

C++

SDK para C++

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Incluir los archivos requeridos.

```
#include <aws/core/Aws.h>
#include <aws/monitoring/CloudWatchClient.h>
#include <aws/monitoring/model/DeleteAlarmsRequest.h>
#include <iostream>
```

Elimine la alarma.

```
        Aws::CloudWatch::CloudWatchClient cw;
        Aws::CloudWatch::Model::DeleteAlarmsRequest request;
        request.AddAlarmNames(alarm_name);

        auto outcome = cw.DeleteAlarms(request);
        if (!outcome.IsSuccess())
        {
            std::cout << "Failed to delete CloudWatch alarm:" <<
                outcome.GetError().GetMessage() << std::endl;
        }
        else
        {
            std::cout << "Successfully deleted CloudWatch alarm " << alarm_name
                << std::endl;
        }
```

- Para obtener información sobre las API, consulte DeleteAlarms en la Referencia de la API de AWS SDK for C++.

CLI

AWS CLI

Eliminación de una alarma

En el siguiente ejemplo, se utiliza el comando `delete-alarms` para eliminar la alarma de Amazon CloudWatch denominada “mialarma”:

```
aws cloudwatch delete-alarms --alarm-names myalarm
```

Salida:

```
This command returns to the prompt if successful.
```

- Para obtener información de la API, consulte DeleteAlarms en la Referencia de comandos de AWS CLI.

Java

SDK para Java 2.x

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.cloudwatch.CloudWatchClient;
import software.amazon.awssdk.services.cloudwatch.model.CloudWatchException;
import software.amazon.awssdk.services.cloudwatch.model.DeleteAlarmsRequest;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
```

```
 */

public class DeleteAlarm {
    public static void main(String[] args) {
        final String usage = """

                Usage:
                  <alarmName>

                Where:
                  alarmName - An alarm name to delete (for example, MyAlarm).
                """;

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }

        String alarmName = args[0];
        Region region = Region.US_EAST_2;
        CloudWatchClient cw = CloudWatchClient.builder()
                .region(region)
                .build();

        deleteCWAlarm(cw, alarmName);
        cw.close();
    }

    public static void deleteCWAlarm(CloudWatchClient cw, String alarmName) {
        try {
            DeleteAlarmsRequest request = DeleteAlarmsRequest.builder()
                    .alarmNames(alarmName)
                    .build();

            cw.deleteAlarms(request);
            System.out.printf("Successfully deleted alarm %s", alarmName);

        } catch (CloudWatchException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- Para obtener información sobre las API, consulte DeleteAlarms en la Referencia de la API de AWS SDK for Java 2.x.

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Importar el SDK y los módulos de cliente, y llamar a la API.

```
import { DeleteAlarmsCommand } from "@aws-sdk/client-cloudwatch";
import { client } from "../libs/client.js";

const run = async () => {
  const command = new DeleteAlarmsCommand({
    AlarmNames: [process.env.CLOUDWATCH_ALARM_NAME], // Set the value of
 CLOUDWATCH_ALARM_NAME to the name of an existing alarm.
  });

  try {
    return await client.send(command);
  } catch (err) {
    console.error(err);
  }
};

export default run();
```

Cree el cliente en un módulo separado y expórtelo.

```
import { CloudWatchClient } from "@aws-sdk/client-cloudwatch";
```

```
export const client = new CloudWatchClient({});
```

- Para obtener información, consulte la Guía para desarrolladores de AWS SDK for JavaScript.
- Para obtener información sobre las API, consulte DeleteAlarms en la Referencia de la API de AWS SDK for JavaScript.

SDK para JavaScript (v2)

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Importar el SDK y los módulos de cliente, y llamar a la API.

```
// Load the AWS SDK for Node.js
var AWS = require("aws-sdk");
// Set the region
AWS.config.update({ region: "REGION" });

// Create CloudWatch service object
var cw = new AWS.CloudWatch({ apiVersion: "2010-08-01" });

var params = {
  AlarmNames: ["Web_Server_CPU_Utilization"],
};

cw.deleteAlarms(params, function (err, data) {
  if (err) {
    console.log("Error", err);
  } else {
    console.log("Success", data);
  }
});
```

- Para obtener información, consulte la Guía para desarrolladores de AWS SDK for JavaScript.

- Para obtener información sobre las API, consulte DeleteAlarms en la Referencia de la API de AWS SDK for JavaScript.

Kotlin

SDK para Kotlin

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
suspend fun deleteAlarm(alarmNameVal: String) {
    val request = DeleteAlarmsRequest {
        alarmNames = listOf(alarmNameVal)
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        cwClient.deleteAlarms(request)
        println("Successfully deleted alarm $alarmNameVal")
    }
}
```

- Para obtener información sobre las API, consulte DeleteAlarms en la Referencia de la API de AWS SDK for Kotlin.

Python

SDK para Python (Boto3)

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
class CloudWatchWrapper:
```

```
    """Encapsulates Amazon CloudWatch functions."""

    def __init__(self, cloudwatch_resource):
        """
        :param cloudwatch_resource: A Boto3 CloudWatch resource.
        """
        self.cloudwatch_resource = cloudwatch_resource


    def delete_metric_alarms(self, metric_namespace, metric_name):
        """
        Deletes all of the alarms that are currently watching the specified
metric.

        :param metric_namespace: The namespace of the metric.
        :param metric_name: The name of the metric.
        """
        try:
            metric = self.cloudwatch_resource.Metric(metric_namespace,
metric_name)
            metric.alarms.delete()
            logger.info(
                "Deleted alarms for metric %s.%s.", metric_namespace, metric_name
            )
        except ClientError:
            logger.exception(
                "Couldn't delete alarms for metric %s.%s.",
                metric_namespace,
                metric_name,
            )
            raise
```

- Para obtener información sobre las API, consulte DeleteAlarms en la Referencia de la API de AWS SDK para Python (Boto3).

SAP ABAP

SDK de SAP ABAP

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
TRY.
    lo_cwt->deletealarms(
      it_alarmnames = it_alarm_names
    ).
    MESSAGE 'Alarms deleted.' TYPE 'I'.
  CATCH /aws1/cx_cwtresourcenotfound .
    MESSAGE 'Resource being accessed is not found.' TYPE 'E'.
ENDTRY.
```

- Para obtener información acerca de las API, consulte DeleteAlarms en la Referencia de API del SDK AWS para SAP ABAP.

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de CloudWatch con SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Uso de **`DeleteAnomalyDetector`** con un AWS SDK o la CLI

Los siguientes ejemplos de código muestran cómo utilizar `DeleteAnomalyDetector`.

Los ejemplos de acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Puede ver esta acción en contexto en el siguiente ejemplo de código:

- Primeros pasos para usar las métricas, los paneles y las alarmas

.NET

AWS SDK for .NET

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
   /// <summary>
   /// Delete a single metric anomaly detector.
   /// </summary>
   /// <param name="anomalyDetector">The anomaly detector to delete.</param>
   /// <returns>True if successful.</returns>
   public async Task<bool> DeleteAnomalyDetector(SingleMetricAnomalyDetector
anomalyDetector)
   {
       var deleteAnomalyDetectorResponse = await
_amazonCloudWatch.DeleteAnomalyDetectorAsync(
           new DeleteAnomalyDetectorRequest()
           {
               SingleMetricAnomalyDetector = anomalyDetector
           });

       return deleteAnomalyDetectorResponse.HttpStatusCode == HttpStatusCode.OK;
   }
```

- Para obtener información sobre la API, consulte DeleteAnomalyDetector en la Referencia de la API de AWS SDK for .NET.

Java

SDK para Java 2.x

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
   public static void deleteAnomalyDetector(CloudWatchClient cw, String
fileName) {
       try {
           // Read values from the JSON file.
           JsonParser parser = new JsonFactory().createParser(new
File(fileName));
           com.fasterxml.jackson.databind.JsonNode rootNode = new
ObjectMapper().readTree(parser);
           String customMetricNamespace =
rootNode.findValue("customMetricNamespace").asText();
           String customMetricName =
rootNode.findValue("customMetricName").asText();

           SingleMetricAnomalyDetector singleMetricAnomalyDetector =
SingleMetricAnomalyDetector.builder()
                   .metricName(customMetricName)
                   .namespace(customMetricNamespace)
                   .stat("Maximum")
                   .build();

           DeleteAnomalyDetectorRequest request =
DeleteAnomalyDetectorRequest.builder()
                   .singleMetricAnomalyDetector(singleMetricAnomalyDetector)
                   .build();

           cw.deleteAnomalyDetector(request);
           System.out.println("Successfully deleted the Anomaly Detector.");

       } catch (CloudWatchException e) {
           System.err.println(e.awsErrorDetails().errorMessage());
           System.exit(1);
       } catch (IOException e) {
           e.printStackTrace();
       }
   }
```

- Para obtener información sobre la API, consulte DeleteAnomalyDetector en la Referencia de la API de AWS SDK for Java 2.x.

Kotlin

SDK para Kotlin

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
suspend fun deleteAnomalyDetector(fileName: String) {
    // Read values from the JSON file.
    val parser = JsonFactory().createParser(File(fileName))
    val rootNode = ObjectMapper().readTree<JsonNode>(parser)
    val customMetricNamespace =
 rootNode.findValue("customMetricNamespace").asText()
    val customMetricName = rootNode.findValue("customMetricName").asText()

    val singleMetricAnomalyDetectorVal = SingleMetricAnomalyDetector {
        metricName = customMetricName
        namespace = customMetricNamespace
        stat = "Maximum"
    }

    val request = DeleteAnomalyDetectorRequest {
        singleMetricAnomalyDetector = singleMetricAnomalyDetectorVal
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        cwClient.deleteAnomalyDetector(request)
        println("Successfully deleted the Anomaly Detector.")
    }
}
```

- Para obtener detalles sobre la API, consulte DeleteAnomalyDetector en la Referencia de la API del AWS SDK para Kotlin.

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de CloudWatch con SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Uso de **`DeleteDashboards`** con un AWS SDK o la CLI

Los siguientes ejemplos de código muestran cómo utilizar `DeleteDashboards`.

Los ejemplos de acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Puede ver esta acción en contexto en el siguiente ejemplo de código:

- Primeros pasos para usar las métricas, los paneles y las alarmas

.NET

AWS SDK for .NET

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    /// <summary>
    /// Delete a list of CloudWatch dashboards.
    /// </summary>
    /// <param name="dashboardNames">List of dashboard names to delete.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> DeleteDashboards(List<string> dashboardNames)
    {
        var deleteDashboardsResponse = await
_amazonCloudWatch.DeleteDashboardsAsync(
            new DeleteDashboardsRequest()
            {
                DashboardNames = dashboardNames
            });

        return deleteDashboardsResponse.HttpStatusCode == HttpStatusCode.OK;
    }
```

- Para obtener información sobre la API, consulte DeleteDashboards en la Referencia de la API de AWS SDK for .NET.

Java

SDK para Java 2.x

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
   public static void deleteDashboard(CloudWatchClient cw, String dashboardName)
{
       try {
           DeleteDashboardsRequest dashboardsRequest =
DeleteDashboardsRequest.builder()
                   .dashboardNames(dashboardName)
                   .build();
           cw.deleteDashboards(dashboardsRequest);
           System.out.println(dashboardName + " was successfully deleted.");

       } catch (CloudWatchException e) {
           System.err.println(e.getMessage());
           System.exit(1);
       }
   }
```

- Para obtener información sobre la API, consulte DeleteDashboards en la Referencia de la API de AWS SDK for Java 2.x.

Kotlin

SDK para Kotlin

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
suspend fun deleteDashboard(dashboardName: String) {
    val dashboardsRequest = DeleteDashboardsRequest {
        dashboardNames = listOf(dashboardName)
    }
    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        cwClient.deleteDashboards(dashboardsRequest)
        println("$dashboardName was successfully deleted.")
    }
}
```

- Para obtener información sobre la API, consulte DeleteDashboards en la Referencia de la API de AWS SDK para Kotlin.

PowerShell

Herramientas para PowerShell

Ejemplo 1: elimina el panel especificado y lo confirma antes de continuar. Para omitir la confirmación, agregue el modificador -Force al comando.

```
Remove-CWDashboard -DashboardName Dashboard1
```

- Para obtener información sobre la API, consulte DeleteDashboards en la Referencia de Cmdlet de AWS Tools for PowerShell.

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de CloudWatch con SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Uso de **`DescribeAlarmHistory`** con un AWS SDK o la CLI

Los siguientes ejemplos de código muestran cómo utilizar `DescribeAlarmHistory`.

Los ejemplos de acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Puede ver esta acción en contexto en el siguiente ejemplo de código:

- Primeros pasos para usar las métricas, los paneles y las alarmas

.NET

AWS SDK for .NET

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    /// <summary>
    /// Describe the history of an alarm for a number of days in the past.
    /// </summary>
    /// <param name="alarmName">The name of the alarm.</param>
    /// <param name="historyDays">The number of days in the past.</param>
    /// <returns>The list of alarm history data.</returns>
    public async Task<List<AlarmHistoryItem>> DescribeAlarmHistory(string
alarmName, int historyDays)
    {
        List<AlarmHistoryItem> alarmHistory = new List<AlarmHistoryItem>();
        var paginatedAlarmHistory =
_amazonCloudWatch.Paginators.DescribeAlarmHistory(
            new DescribeAlarmHistoryRequest()
            {
                AlarmName = alarmName,
                EndDateUtc = DateTime.UtcNow,
                HistoryItemType = HistoryItemType.StateUpdate,
                StartDateUtc = DateTime.UtcNow.AddDays(-historyDays)
            });

        await foreach (var data in paginatedAlarmHistory.AlarmHistoryItems)
        {
            alarmHistory.Add(data);
        }
        return alarmHistory;
    }
```

- Para obtener información sobre las API, consulte DescribeAlarmHistory en la Referencia de la API de AWS SDK for .NET.

CLI

AWS CLI

Recuperación del historial de una alarma

En el siguiente ejemplo, se utiliza el comando `describe-alarm-history` para recuperar el historial de la alarma de Amazon CloudWatch denominada “mialarma”:

```
aws cloudwatch describe-alarm-history --alarm-name "myalarm" --history-item-type
 StateUpdate
```

Salida:

```
{
    "AlarmHistoryItems": [
        {
            "Timestamp": "2014-04-09T18:59:06.442Z",
            "HistoryItemType": "StateUpdate",
            "AlarmName": "myalarm",
            "HistoryData": "{\"version\":\"1.0\",\"oldState\":{\"stateValue
\":\"ALARM\",\"stateReason\":\"testing purposes\"},\"newState\":{\"stateValue
\":\"OK\",\"stateReason\":\"Threshold Crossed: 2 datapoints were not
 greater than the threshold (70.0). The most recent datapoints: [38.958,
 40.292].\",\"stateReasonData\":{\"version\":\"1.0\",\"queryDate\":
\"2014-04-09T18:59:06.419+0000\",\"startDate\":\"2014-04-09T18:44:00.000+0000\",
\"statistic\":\"Average\",\"period\":300,\"recentDatapoints\":[38.958,40.292],
\"threshold\":70.0}}}",
            "HistorySummary": "Alarm updated from ALARM to OK"
        },
        {
            "Timestamp": "2014-04-09T18:59:05.805Z",
            "HistoryItemType": "StateUpdate",
            "AlarmName": "myalarm",
            "HistoryData": "{\"version\":\"1.0\",\"oldState\":{\"stateValue
\":\"OK\",\"stateReason\":\"Threshold Crossed: 2 datapoints were
 not greater than the threshold (70.0). The most recent datapoints:
 [38.839999999999996, 39.714].\",\"stateReasonData\":{\"version\":
\"1.0\",\"queryDate\":\"2014-03-11T22:45:41.569+0000\",\"startDate\":
\"2014-03-11T22:30:00.000+0000\",\"statistic\":\"Average\",\"period\":300,
\"recentDatapoints\":[38.839999999999996,39.714],\"threshold\":70.0}},\"newState
\":{\"stateValue\":\"ALARM\",\"stateReason\":\"testing purposes\"}}",
            "HistorySummary": "Alarm updated from OK to ALARM"
```

```
        }
    ]
}
```

- Para obtener información sobre la API, consulte DescribeAlarmHistory en la Referencia de comandos de la AWS CLI.

Java

SDK para Java 2.x

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
   public static void getAlarmHistory(CloudWatchClient cw, String fileName,
String date) {
       try {
           // Read values from the JSON file.
           JsonParser parser = new JsonFactory().createParser(new
File(fileName));
           com.fasterxml.jackson.databind.JsonNode rootNode = new
ObjectMapper().readTree(parser);
           String alarmName = rootNode.findValue("exampleAlarmName").asText();

           Instant start = Instant.parse(date);
           Instant endDate = Instant.now();
           DescribeAlarmHistoryRequest historyRequest =
DescribeAlarmHistoryRequest.builder()
                   .startDate(start)
                   .endDate(endDate)
                   .alarmName(alarmName)
                   .historyItemType(HistoryItemType.ACTION)
                   .build();

           DescribeAlarmHistoryResponse response =
cw.describeAlarmHistory(historyRequest);
           List<AlarmHistoryItem> historyItems = response.alarmHistoryItems();
           if (historyItems.isEmpty()) {
```

```
                System.out.println("No alarm history data found for " + alarmName
 + ".");
            } else {
                for (AlarmHistoryItem item : historyItems) {
                    System.out.println("History summary: " +
 item.historySummary());
                    System.out.println("Time stamp: " + item.timestamp());
                }
            }

        } catch (CloudWatchException | IOException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }
```

- Para obtener información sobre las API, consulte DescribeAlarmHistory en la Referencia de la API de AWS SDK for Java 2.x.

Kotlin

SDK para Kotlin

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
suspend fun getAlarmHistory(fileName: String, date: String) {
    // Read values from the JSON file.
    val parser = JsonFactory().createParser(File(fileName))
    val rootNode = ObjectMapper().readTree<JsonNode>(parser)
    val alarmNameVal = rootNode.findValue("exampleAlarmName").asText()
    val start = Instant.parse(date)
    val endDateVal = Instant.now()

    val historyRequest = DescribeAlarmHistoryRequest {
        startDate = aws.smithy.kotlin.runtime.time.Instant(start)
        endDate = aws.smithy.kotlin.runtime.time.Instant(endDateVal)
```

```
        alarmName = alarmNameVal
        historyItemType = HistoryItemType.Action
    }

    CloudWatchClient { credentialsProvider = EnvironmentCredentialsProvider();
 region = "us-east-1" }.use { cwClient ->
        val response = cwClient.describeAlarmHistory(historyRequest)
        val historyItems = response.alarmHistoryItems
        if (historyItems != null) {
            if (historyItems.isEmpty()) {
                println("No alarm history data found for $alarmNameVal.")
            } else {
                for (item in historyItems) {
                    println("History summary ${item.historySummary}")
                    println("Time stamp: ${item.timestamp}")
                }
            }
        }
    }
}
```

- Para obtener información acerca de la API, consulte DescribeAlarmHistory en la Referencia de la API del SDK de AWS para Kotlin.

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de CloudWatch con SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Uso de **`DescribeAlarms`** con un AWS SDK o la CLI

Los siguientes ejemplos de código muestran cómo utilizar `DescribeAlarms`.

Los ejemplos de acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Puede ver esta acción en su contexto en los siguientes ejemplos de código:

- Primeros pasos para usar alarmas
- Primeros pasos para usar las métricas, los paneles y las alarmas

.NET

AWS SDK for .NET

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    /// <summary>
    /// Describe the current alarms, optionally filtered by state.
    /// </summary>
    /// <param name="stateValue">Optional filter for alarm state.</param>
    /// <returns>The list of alarm data.</returns>
    public async Task<List<MetricAlarm>> DescribeAlarms(StateValue? stateValue =
null)
    {
        List<MetricAlarm> alarms = new List<MetricAlarm>();
        var paginatedDescribeAlarms =
_amazonCloudWatch.Paginators.DescribeAlarms(
            new DescribeAlarmsRequest()
            {
                StateValue = stateValue
            });

        await foreach (var data in paginatedDescribeAlarms.MetricAlarms)
        {
            alarms.Add(data);
        }
        return alarms;
    }
```

- Para obtener información sobre la API, consulte DescribeAlarms en la Referencia de la API de AWS SDK for .NET.

CLI

AWS CLI

Visualización de información acerca de una alarma

En el siguiente ejemplo, se utiliza el comando `describe-alarms` para proporcionar información sobre la alarma denominada “mialarma”:

```
aws cloudwatch describe-alarms --alarm-names "myalarm"
```

Salida:

```
{
    "MetricAlarms": [
        {
            "EvaluationPeriods": 2,
            "AlarmArn": "arn:aws:cloudwatch:us-
east-1:123456789012:alarm:myalarm",
            "StateUpdatedTimestamp": "2014-04-09T18:59:06.442Z",
            "AlarmConfigurationUpdatedTimestamp": "2012-12-27T00:49:54.032Z",
            "ComparisonOperator": "GreaterThanThreshold",
            "AlarmActions": [
                "arn:aws:sns:us-east-1:123456789012:myHighCpuAlarm"
            ],
            "Namespace": "AWS/EC2",
            "AlarmDescription": "CPU usage exceeds 70 percent",
            "StateReasonData": "{\"version\":\"1.0\",\"queryDate\":
\"2014-04-09T18:59:06.419+0000\",\"startDate\":\"2014-04-09T18:44:00.000+0000\",
\"statistic\":\"Average\",\"period\":300,\"recentDatapoints\":[38.958,40.292],
\"threshold\":70.0}",
            "Period": 300,
            "StateValue": "OK",
            "Threshold": 70.0,
            "AlarmName": "myalarm",
            "Dimensions": [
                {
                    "Name": "InstanceId",
                    "Value": "i-0c986c72"
                }
            ],
            "Statistic": "Average",
```

```
            "StateReason": "Threshold Crossed: 2 datapoints were not greater than
 the threshold (70.0). The most recent datapoints: [38.958, 40.292].",
            "InsufficientDataActions": [],
            "OKActions": [],
            "ActionsEnabled": true,
            "MetricName": "CPUUtilization"
        }
    ]
}
```

- Para obtener información sobre la API, consulte DescribeAlarms en la Referencia de comandos de la AWS CLI.

Java

SDK para Java 2.x

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    public static void describeAlarms(CloudWatchClient cw) {
        try {
            List<AlarmType> typeList = new ArrayList<>();
            typeList.add(AlarmType.METRIC_ALARM);

            DescribeAlarmsRequest alarmsRequest = DescribeAlarmsRequest.builder()
                    .alarmTypes(typeList)
                    .maxRecords(10)
                    .build();

            DescribeAlarmsResponse response = cw.describeAlarms(alarmsRequest);
            List<MetricAlarm> alarmList = response.metricAlarms();
            for (MetricAlarm alarm : alarmList) {
                System.out.println("Alarm name: " + alarm.alarmName());
                System.out.println("Alarm description: " +
 alarm.alarmDescription());
            }
        } catch (CloudWatchException e) {
```

```
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
```

- Para obtener información sobre la API, consulte DescribeAlarms en la Referencia de la API de AWS SDK for Java 2.x.

Kotlin

SDK para Kotlin

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
suspend fun describeAlarms() {
    val typeList = ArrayList<AlarmType>()
    typeList.add(AlarmType.MetricAlarm)
    val alarmsRequest = DescribeAlarmsRequest {
        alarmTypes = typeList
        maxRecords = 10
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        val response = cwClient.describeAlarms(alarmsRequest)
        response.metricAlarms?.forEach { alarm ->
            println("Alarm name: ${alarm.alarmName}")
            println("Alarm description: ${alarm.alarmDescription}")
        }
    }
}
```

- Para obtener información sobre la API, consulte DescribeAlarms en la Referencia de la API de AWS SDK para Kotlin.

Ruby

SDK para Ruby

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
require "aws-sdk-cloudwatch"

# Lists the names of available Amazon CloudWatch alarms.
#
# @param cloudwatch_client [Aws::CloudWatch::Client]
#   An initialized CloudWatch client.
# @example
#   list_alarms(Aws::CloudWatch::Client.new(region: 'us-east-1'))
def list_alarms(cloudwatch_client)
  response = cloudwatch_client.describe_alarms
  if response.metric_alarms.count.positive?
    response.metric_alarms.each do |alarm|
      puts alarm.alarm_name
    end
  else
    puts "No alarms found."
  end
rescue StandardError => e
  puts "Error getting information about alarms: #{e.message}"
end
```

- Para obtener información sobre la API, consulte DescribeAlarms en la Referencia de la API de AWS SDK for Ruby.

SAP ABAP

SDK de SAP ABAP

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    TRY.
        oo_result = lo_cwt->describealarms(                " oo_result is
 returned for testing purposes. "
          it_alarmnames = it_alarm_names
        ).
        MESSAGE 'Alarms retrieved.' TYPE 'I'.
      CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
        DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
        MESSAGE lv_error TYPE 'E'.
    ENDTRY.
```

- Para más información sobre la API, consulte DescribeAlarms en la Referencia de API del SDK AWS para SAP ABAP.

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de CloudWatch con SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Uso de **`DescribeAlarmsForMetric`** con un AWS SDK o la CLI

Los siguientes ejemplos de código muestran cómo utilizar `DescribeAlarmsForMetric`.

Los ejemplos de acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Puede ver esta acción en su contexto en los siguientes ejemplos de código:

- Primeros pasos para usar las métricas, los paneles y las alarmas
- Gestionar métricas y alarmas

.NET

AWS SDK for .NET

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    /// <summary>
    /// Describe the current alarms for a specific metric.
    /// </summary>
    /// <param name="metricNamespace">The namespace of the metric.</param>
    /// <param name="metricName">The name of the metric.</param>
    /// <returns>The list of alarm data.</returns>
    public async Task<List<MetricAlarm>> DescribeAlarmsForMetric(string
metricNamespace, string metricName)
    {
        var alarmsResult = await _amazonCloudWatch.DescribeAlarmsForMetricAsync(
            new DescribeAlarmsForMetricRequest()
            {
                Namespace = metricNamespace,
                MetricName = metricName
            });

        return alarmsResult.MetricAlarms;
    }
```

- Para obtener información sobre las API, consulte DescribeAlarmsForMetric en la Referencia de la API de AWS SDK for .NET.

C++

SDK para C++

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Incluir los archivos requeridos.

```
#include <aws/core/Aws.h>
#include <aws/monitoring/CloudWatchClient.h>
#include <aws/monitoring/model/DescribeAlarmsRequest.h>
#include <aws/monitoring/model/DescribeAlarmsResult.h>
#include <iomanip>
#include <iostream>
```

Describa las alarmas

```
        Aws::CloudWatch::CloudWatchClient cw;
        Aws::CloudWatch::Model::DescribeAlarmsRequest request;
        request.SetMaxRecords(1);

        bool done = false;
        bool header = false;
        while (!done)
        {
            auto outcome = cw.DescribeAlarms(request);
            if (!outcome.IsSuccess())
            {
                std::cout << "Failed to describe CloudWatch alarms:" <<
                    outcome.GetError().GetMessage() << std::endl;
                break;
            }

            if (!header)
            {
                std::cout << std::left <<
                    std::setw(32) << "Name" <<
```

```
                    std::setw(64) << "Arn" <<
                    std::setw(64) << "Description" <<
                    std::setw(20) << "LastUpdated" <<
                    std::endl;
                header = true;
            }

            const auto &alarms = outcome.GetResult().GetMetricAlarms();
            for (const auto &alarm : alarms)
            {
                std::cout << std::left <<
                    std::setw(32) << alarm.GetAlarmName() <<
                    std::setw(64) << alarm.GetAlarmArn() <<
                    std::setw(64) << alarm.GetAlarmDescription() <<
                    std::setw(20) <<
                    alarm.GetAlarmConfigurationUpdatedTimestamp().ToGmtString(
                        SIMPLE_DATE_FORMAT_STR) <<
                    std::endl;
            }

            const auto &next_token = outcome.GetResult().GetNextToken();
            request.SetNextToken(next_token);
            done = next_token.empty();
        }
```

- Para obtener información sobre las API, consulte DescribeAlarmsForMetric en la Referencia de la API de AWS SDK for C++.

CLI

AWS CLI

Visualización de información sobre las alarmas asociadas a una métrica

En el siguiente ejemplo, se utiliza el comando `describe-alarms-for-metric` para mostrar información sobre las alarmas asociadas a la métrica CPUUtilization de Amazon EC2 y a la instancia con el ID i-0c986c72.:

```
aws cloudwatch describe-alarms-for-metric --metric-name CPUUtilization --
namespace AWS/EC2 --dimensions Name=InstanceId,Value=i-0c986c72
```

Salida:


```
{
    "MetricAlarms": [
        {
            "EvaluationPeriods": 10,
            "AlarmArn": "arn:aws:cloudwatch:us-
east-1:111122223333:alarm:myHighCpuAlarm2",
            "StateUpdatedTimestamp": "2013-10-30T03:03:51.479Z",
            "AlarmConfigurationUpdatedTimestamp": "2013-10-30T03:03:50.865Z",
            "ComparisonOperator": "GreaterThanOrEqualToThreshold",
            "AlarmActions": [
                "arn:aws:sns:us-east-1:111122223333:NotifyMe"
            ],
            "Namespace": "AWS/EC2",
            "AlarmDescription": "CPU usage exceeds 70 percent",
            "StateReasonData": "{\"version\":\"1.0\",\"queryDate\":
\"2013-10-30T03:03:51.479+0000\",\"startDate\":\"2013-10-30T02:08:00.000+0000\",
\"statistic\":\"Average\",\"period\":300,\"recentDatapoints\":
[40.698,39.612,42.432,39.796,38.816,42.28,42.854,40.088,40.760000000000005,41.316],
\"threshold\":70.0}",
            "Period": 300,
            "StateValue": "OK",
            "Threshold": 70.0,
            "AlarmName": "myHighCpuAlarm2",
            "Dimensions": [
                {
                    "Name": "InstanceId",
                    "Value": "i-0c986c72"
                }
            ],
            "Statistic": "Average",
            "StateReason": "Threshold Crossed: 10 datapoints were not
 greater than or equal to the threshold (70.0). The most recent datapoints:
 [40.760000000000005, 41.316].",
            "InsufficientDataActions": [],
            "OKActions": [],
            "ActionsEnabled": true,
            "MetricName": "CPUUtilization"
        },
        {
            "EvaluationPeriods": 2,
            "AlarmArn": "arn:aws:cloudwatch:us-
east-1:111122223333:alarm:myHighCpuAlarm",
```

```
            "StateUpdatedTimestamp": "2014-04-09T18:59:06.442Z",
            "AlarmConfigurationUpdatedTimestamp": "2014-04-09T22:26:05.958Z",
            "ComparisonOperator": "GreaterThanThreshold",
            "AlarmActions": [
                "arn:aws:sns:us-east-1:111122223333:HighCPUAlarm"
            ],
            "Namespace": "AWS/EC2",
            "AlarmDescription": "CPU usage exceeds 70 percent",
            "StateReasonData": "{\"version\":\"1.0\",\"queryDate\":
\"2014-04-09T18:59:06.419+0000\",\"startDate\":\"2014-04-09T18:44:00.000+0000\",
\"statistic\":\"Average\",\"period\":300,\"recentDatapoints\":[38.958,40.292],
\"threshold\":70.0}",
            "Period": 300,
            "StateValue": "OK",
            "Threshold": 70.0,
            "AlarmName": "myHighCpuAlarm",
            "Dimensions": [
                {
                    "Name": "InstanceId",
                    "Value": "i-0c986c72"
                }
            ],
            "Statistic": "Average",
            "StateReason": "Threshold Crossed: 2 datapoints were not greater than
 the threshold (70.0). The most recent datapoints: [38.958, 40.292].",
            "InsufficientDataActions": [],
            "OKActions": [],
            "ActionsEnabled": false,
            "MetricName": "CPUUtilization"
        }
    ]
}
```

- Para obtener información sobre la API, consulte DescribeAlarmsForMetric en la Referencia de comandos de la AWS CLI.

Java

SDK para Java 2.x

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
   public static void checkForMetricAlarm(CloudWatchClient cw, String fileName)
{
       try {
           // Read values from the JSON file.
           JsonParser parser = new JsonFactory().createParser(new
File(fileName));
           com.fasterxml.jackson.databind.JsonNode rootNode = new
ObjectMapper().readTree(parser);
           String customMetricNamespace =
rootNode.findValue("customMetricNamespace").asText();
           String customMetricName =
rootNode.findValue("customMetricName").asText();
           boolean hasAlarm = false;
           int retries = 10;

           DescribeAlarmsForMetricRequest metricRequest =
DescribeAlarmsForMetricRequest.builder()
                   .metricName(customMetricName)
                   .namespace(customMetricNamespace)
                   .build();

           while (!hasAlarm && retries > 0) {
               DescribeAlarmsForMetricResponse response =
cw.describeAlarmsForMetric(metricRequest);
               hasAlarm = response.hasMetricAlarms();
               retries--;
               Thread.sleep(20000);
               System.out.println(".");
           }
           if (!hasAlarm)
               System.out.println("No Alarm state found for " + customMetricName
+ " after 10 retries.");
```

```
            else
                System.out.println("Alarm state found for " + customMetricName +
 ".");

        } catch (CloudWatchException | IOException | InterruptedException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }
```

- Para obtener información sobre las API, consulte DescribeAlarmsForMetric en la Referencia de la API de AWS SDK for Java 2.x.

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Importar el SDK y los módulos de cliente, y llamar a la API.

```
import { DescribeAlarmsCommand } from "@aws-sdk/client-cloudwatch";
import { client } from "../libs/client.js";

const run = async () => {
  const command = new DescribeAlarmsCommand({
    AlarmNames: [process.env.CLOUDWATCH_ALARM_NAME], // Set the value of
 CLOUDWATCH_ALARM_NAME to the name of an existing alarm.
  });

  try {
    return await client.send(command);
  } catch (err) {
    console.error(err);
  }
};
```

```
export default run();
```

Cree el cliente en un módulo separado y expórtelo.

```
import { CloudWatchClient } from "@aws-sdk/client-cloudwatch";

export const client = new CloudWatchClient({});
```

- Para obtener información, consulte la Guía para desarrolladores de AWS SDK for JavaScript.
- Para obtener información sobre las API, consulte DescribeAlarmsForMetric en la Referencia de la API de AWS SDK for JavaScript.

SDK para JavaScript (v2)

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
// Load the AWS SDK for Node.js
var AWS = require("aws-sdk");
// Set the region
AWS.config.update({ region: "REGION" });

// Create CloudWatch service object
var cw = new AWS.CloudWatch({ apiVersion: "2010-08-01" });

cw.describeAlarms({ StateValue: "INSUFFICIENT_DATA" }, function (err, data) {
  if (err) {
    console.log("Error", err);
  } else {
    // List the names of all current alarms in the console
    data.MetricAlarms.forEach(function (item, index, array) {
      console.log(item.AlarmName);
    });
  }
});
```

- Para obtener información, consulte la Guía para desarrolladores de AWS SDK for JavaScript.
- Para obtener información sobre las API, consulte DescribeAlarmsForMetric en la Referencia de la API de AWS SDK for JavaScript.

Kotlin

SDK para Kotlin

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
suspend fun checkForMetricAlarm(fileName: String?) {
    // Read values from the JSON file.
    val parser = JsonFactory().createParser(File(fileName))
    val rootNode = ObjectMapper().readTree<JsonNode>(parser)
    val customMetricNamespace =
 rootNode.findValue("customMetricNamespace").asText()
    val customMetricName = rootNode.findValue("customMetricName").asText()
    var hasAlarm = false
    var retries = 10

    val metricRequest = DescribeAlarmsForMetricRequest {
        metricName = customMetricName
        namespace = customMetricNamespace
    }
    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        while (!hasAlarm && retries > 0) {
            val response = cwClient.describeAlarmsForMetric(metricRequest)
            if (response.metricAlarms?.count()!! > 0) {
                hasAlarm = true
            }
            retries--
            delay(20000)
            println(".")
        }
```

```
        if (!hasAlarm) println("No Alarm state found for $customMetricName after
 10 retries.") else println("Alarm state found for $customMetricName.")
    }
}
```

- Para obtener información sobre las API, consulte DescribeAlarmsForMetric en la Referencia de la API de AWS SDK for Kotlin.

Python

SDK para Python (Boto3)

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
class CloudWatchWrapper:
    """Encapsulates Amazon CloudWatch functions."""

    def __init__(self, cloudwatch_resource):
        """
        :param cloudwatch_resource: A Boto3 CloudWatch resource.
        """
        self.cloudwatch_resource = cloudwatch_resource


    def get_metric_alarms(self, metric_namespace, metric_name):
        """
        Gets the alarms that are currently watching the specified metric.

        :param metric_namespace: The namespace of the metric.
        :param metric_name: The name of the metric.
        :returns: An iterator that yields the alarms.
        """
        metric = self.cloudwatch_resource.Metric(metric_namespace, metric_name)
        alarm_iter = metric.alarms.all()
        logger.info("Got alarms for metric %s.%s.", metric_namespace,
 metric_name)
```

```
        return alarm_iter
```

- Para obtener información sobre las API, consulte DescribeAlarmsForMetric en la Referencia de la API de AWS SDK para Python (Boto3).

Ruby

SDK para Ruby

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
#
# @param cloudwatch_client [Aws::CloudWatch::Client]
#   An initialized CloudWatch client.
# @example
#   describe_metric_alarms(Aws::CloudWatch::Client.new(region: 'us-east-1'))
def describe_metric_alarms(cloudwatch_client)
  response = cloudwatch_client.describe_alarms

  if response.metric_alarms.count.positive?
    response.metric_alarms.each do |alarm|
      puts "-" * 16
      puts "Name:           " + alarm.alarm_name
      puts "State value:    " + alarm.state_value
      puts "State reason:   " + alarm.state_reason
      puts "Metric:         " + alarm.metric_name
      puts "Namespace:      " + alarm.namespace
      puts "Statistic:      " + alarm.statistic
      puts "Period:         " + alarm.period.to_s
      puts "Unit:           " + alarm.unit.to_s
      puts "Eval. periods:  " + alarm.evaluation_periods.to_s
      puts "Threshold:      " + alarm.threshold.to_s
      puts "Comp. operator: " + alarm.comparison_operator
```

```
      if alarm.key?(:ok_actions) && alarm.ok_actions.count.positive?
        puts "OK actions:"
        alarm.ok_actions.each do |a|
          puts "  " + a
        end
      end

      if alarm.key?(:alarm_actions) && alarm.alarm_actions.count.positive?
        puts "Alarm actions:"
        alarm.alarm_actions.each do |a|
          puts "  " + a
        end
      end

      if alarm.key?(:insufficient_data_actions) &&
          alarm.insufficient_data_actions.count.positive?
        puts "Insufficient data actions:"
        alarm.insufficient_data_actions.each do |a|
          puts "  " + a
        end
      end

      puts "Dimensions:"
      if alarm.key?(:dimensions) && alarm.dimensions.count.positive?
        alarm.dimensions.each do |d|
          puts "  Name: " + d.name + ", Value: " + d.value
        end
      else
        puts "  None for this alarm."
      end
    end
  else
    puts "No alarms found."
  end
rescue StandardError => e
  puts "Error getting information about alarms: #{e.message}"
end

# Example usage:
def run_me
  region = ""

  # Print usage information and then stop.
  if ARGV[0] == "--help" || ARGV[0] == "-h"
```

```
    puts "Usage:   ruby cw-ruby-example-show-alarms.rb REGION"
    puts "Example: ruby cw-ruby-example-show-alarms.rb us-east-1"
    exit 1
  # If no values are specified at the command prompt, use these default values.
  elsif ARGV.count.zero?
    region = "us-east-1"
  # Otherwise, use the values as specified at the command prompt.
  else
    region = ARGV[0]
  end

  cloudwatch_client = Aws::CloudWatch::Client.new(region: region)
  puts "Available alarms:"
  describe_metric_alarms(cloudwatch_client)
end

run_me if $PROGRAM_NAME == __FILE__
```

- Para obtener información sobre las API, consulte DescribeAlarmsForMetric en la Referencia de la API de AWS SDK for Ruby.

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de CloudWatch con SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Uso de **`DescribeAnomalyDetectors`** con un AWS SDK o la CLI

Los siguientes ejemplos de código muestran cómo utilizar `DescribeAnomalyDetectors`.

Los ejemplos de acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Puede ver esta acción en contexto en el siguiente ejemplo de código:

- Primeros pasos para usar las métricas, los paneles y las alarmas

.NET

AWS SDK for .NET

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    /// <summary>
    /// Describe anomaly detectors for a metric and namespace.
    /// </summary>
    /// <param name="metricNamespace">The namespace of the metric.</param>
    /// <param name="metricName">The metric of the anomaly detectors.</param>
    /// <returns>The list of detectors.</returns>
    public async Task<List<AnomalyDetector>> DescribeAnomalyDetectors(string
metricNamespace, string metricName)
    {
        List<AnomalyDetector> detectors = new List<AnomalyDetector>();
        var paginatedDescribeAnomalyDetectors =
_amazonCloudWatch.Paginators.DescribeAnomalyDetectors(
            new DescribeAnomalyDetectorsRequest()
            {
                MetricName = metricName,
                Namespace = metricNamespace
            });

        await foreach (var data in
paginatedDescribeAnomalyDetectors.AnomalyDetectors)
        {
            detectors.Add(data);
        }

        return detectors;
    }
```

- Para obtener información sobre la API, consulte DescribeAnomalyDetectors en Referencia de la API de AWS SDK for .NET.

Java

SDK para Java 2.x

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
   public static void describeAnomalyDetectors(CloudWatchClient cw, String
fileName) {
       try {
           // Read values from the JSON file.
           JsonParser parser = new JsonFactory().createParser(new
File(fileName));
           com.fasterxml.jackson.databind.JsonNode rootNode = new
ObjectMapper().readTree(parser);
           String customMetricNamespace =
rootNode.findValue("customMetricNamespace").asText();
           String customMetricName =
rootNode.findValue("customMetricName").asText();
           DescribeAnomalyDetectorsRequest detectorsRequest =
DescribeAnomalyDetectorsRequest.builder()
                   .maxResults(10)
                   .metricName(customMetricName)
                   .namespace(customMetricNamespace)
                   .build();

           DescribeAnomalyDetectorsResponse response =
cw.describeAnomalyDetectors(detectorsRequest);
           List<AnomalyDetector> anomalyDetectorList =
response.anomalyDetectors();
           for (AnomalyDetector detector : anomalyDetectorList) {
               System.out.println("Metric name: " +
detector.singleMetricAnomalyDetector().metricName());
               System.out.println("State: " + detector.stateValue());
           }

       } catch (CloudWatchException | IOException e) {
           System.err.println(e.getMessage());
           System.exit(1);
```

```
        }
    }
```

- Para obtener información sobre la API, consulte DescribeAnomalyDetectors en Referencia de la API de AWS SDK for Java 2.x.

Kotlin

SDK para Kotlin

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
suspend fun describeAnomalyDetectors(fileName: String) {
    // Read values from the JSON file.
    val parser = JsonFactory().createParser(File(fileName))
    val rootNode = ObjectMapper().readTree<JsonNode>(parser)
    val customMetricNamespace =
 rootNode.findValue("customMetricNamespace").asText()
    val customMetricName = rootNode.findValue("customMetricName").asText()

    val detectorsRequest = DescribeAnomalyDetectorsRequest {
        maxResults = 10
        metricName = customMetricName
        namespace = customMetricNamespace
    }
    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        val response = cwClient.describeAnomalyDetectors(detectorsRequest)
        response.anomalyDetectors?.forEach { detector ->
            println("Metric name:
 ${detector.singleMetricAnomalyDetector?.metricName}")
            println("State: ${detector.stateValue}")
        }
    }
}
```

- Para obtener información acerca de la API, consulte DescribeAnomalyDetectors en la Referencia de la API del SDK de AWS para Kotlin.

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de CloudWatch con SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Uso de **`DisableAlarmActions`** con un AWS SDK o la CLI

Los siguientes ejemplos de código muestran cómo utilizar `DisableAlarmActions`.

Los ejemplos de acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Puede ver esta acción en su contexto en los siguientes ejemplos de código:

- Primeros pasos para usar alarmas
- Gestionar métricas y alarmas

.NET

AWS SDK for .NET

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    /// <summary>
    /// Disable the actions for a list of alarms from CloudWatch.
    /// </summary>
    /// <param name="alarmNames">A list of names of alarms.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> DisableAlarmActions(List<string> alarmNames)
    {
        var disableAlarmActionsResult = await
_amazonCloudWatch.DisableAlarmActionsAsync(
            new DisableAlarmActionsRequest()
            {
                AlarmNames = alarmNames
            });
```

```
        return disableAlarmActionsResult.HttpStatusCode == HttpStatusCode.OK;
    }
```

- Para obtener información sobre las API, consulte DisableAlarmActions en la Referencia de la API de AWS SDK for .NET.

C++

SDK para C++

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Incluir los archivos requeridos.

```
#include <aws/core/Aws.h>
#include <aws/monitoring/CloudWatchClient.h>
#include <aws/monitoring/model/DisableAlarmActionsRequest.h>
#include <iostream>
```

Deshabilite las acciones de alarma.

```
        Aws::CloudWatch::CloudWatchClient cw;

        Aws::CloudWatch::Model::DisableAlarmActionsRequest
disableAlarmActionsRequest;
        disableAlarmActionsRequest.AddAlarmNames(alarm_name);

        auto disableAlarmActionsOutcome =
cw.DisableAlarmActions(disableAlarmActionsRequest);
        if (!disableAlarmActionsOutcome.IsSuccess())
        {
            std::cout << "Failed to disable actions for alarm " << alarm_name <<
                ": " << disableAlarmActionsOutcome.GetError().GetMessage() <<
                std::endl;
```

```
        }
        else
        {
            std::cout << "Successfully disabled actions for alarm " <<
                alarm_name << std::endl;
        }
```

- Para obtener información sobre las API, consulte DisableAlarmActions en la Referencia de la API de AWS SDK for C++.

CLI

AWS CLI

Desactivación de acciones en una alarma

En el siguiente ejemplo, se utiliza el comando `disable-alarm-actions` para deshabilitar todas las acciones de la alarma denominada mialarma:

```
aws cloudwatch disable-alarm-actions --alarm-names myalarm
```

Este comando vuelve a la petición si se ejecuta correctamente.

- Para obtener información sobre la API, consulte DisableAlarmActions en la Referencia de comandos de la AWS CLI.

Java

SDK para Java 2.x

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.cloudwatch.CloudWatchClient;
import software.amazon.awssdk.services.cloudwatch.model.CloudWatchException;
```

```
import
 software.amazon.awssdk.services.cloudwatch.model.DisableAlarmActionsRequest;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class DisableAlarmActions {
    public static void main(String[] args) {
        final String usage = """

                Usage:
                  <alarmName>

                Where:
                  alarmName - An alarm name to disable (for example, MyAlarm).
                """;

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }

        String alarmName = args[0];
        Region region = Region.US_EAST_1;
        CloudWatchClient cw = CloudWatchClient.builder()
                .region(region)
                .build();

        disableActions(cw, alarmName);
        cw.close();
    }

    public static void disableActions(CloudWatchClient cw, String alarmName) {
        try {
            DisableAlarmActionsRequest request =
 DisableAlarmActionsRequest.builder()
                    .alarmNames(alarmName)
                    .build();
```

```
            cw.disableAlarmActions(request);
            System.out.printf("Successfully disabled actions on alarm %s",
 alarmName);

        } catch (CloudWatchException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- Para obtener información sobre las API, consulte DisableAlarmActions en la Referencia de la API de AWS SDK for Java 2.x.

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Importar el SDK y los módulos de cliente, y llamar a la API.

```
import { DisableAlarmActionsCommand } from "@aws-sdk/client-cloudwatch";
import { client } from "../libs/client.js";

const run = async () => {
  const command = new DisableAlarmActionsCommand({
    AlarmNames: process.env.CLOUDWATCH_ALARM_NAME, // Set the value of
 CLOUDWATCH_ALARM_NAME to the name of an existing alarm.
  });

  try {
    return await client.send(command);
  } catch (err) {
    console.error(err);
```

```
  }
};

export default run();
```

Cree el cliente en un módulo separado y expórtelo.

```
import { CloudWatchClient } from "@aws-sdk/client-cloudwatch";

export const client = new CloudWatchClient({});
```

- Para obtener información, consulte la Guía para desarrolladores de AWS SDK for JavaScript.
- Para obtener información sobre las API, consulte DisableAlarmActions en la Referencia de la API de AWS SDK for JavaScript.

SDK para JavaScript (v2)

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Importar el SDK y los módulos de cliente, y llamar a la API.

```
// Load the AWS SDK for Node.js
var AWS = require("aws-sdk");
// Set the region
AWS.config.update({ region: "REGION" });

// Create CloudWatch service object
var cw = new AWS.CloudWatch({ apiVersion: "2010-08-01" });

cw.disableAlarmActions(
  { AlarmNames: ["Web_Server_CPU_Utilization"] },
  function (err, data) {
    if (err) {
      console.log("Error", err);
```

```
    } else {
      console.log("Success", data);
    }
  }
);
```

- Para obtener información, consulte la Guía para desarrolladores de AWS SDK for JavaScript.
- Para obtener información sobre las API, consulte DisableAlarmActions en la Referencia de la API de AWS SDK for JavaScript.

Kotlin

SDK para Kotlin

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
suspend fun disableActions(alarmName: String) {

    val request = DisableAlarmActionsRequest {
        alarmNames = listOf(alarmName)
    }
    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        cwClient.disableAlarmActions(request)
        println("Successfully disabled actions on alarm $alarmName")
    }
}
```

- Para obtener información de las API, consulte DisableAlarmActions en Referencia de la API de SDK de AWS para Kotlin.

Python

SDK para Python (Boto3)

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
class CloudWatchWrapper:
    """Encapsulates Amazon CloudWatch functions."""

    def __init__(self, cloudwatch_resource):
        """
        :param cloudwatch_resource: A Boto3 CloudWatch resource.
        """
        self.cloudwatch_resource = cloudwatch_resource


    def enable_alarm_actions(self, alarm_name, enable):
        """
        Enables or disables actions on the specified alarm. Alarm actions can be
        used to send notifications or automate responses when an alarm enters a
        particular state.

        :param alarm_name: The name of the alarm.
        :param enable: When True, actions are enabled for the alarm. Otherwise,
 they
                       disabled.
        """
        try:
            alarm = self.cloudwatch_resource.Alarm(alarm_name)
            if enable:
                alarm.enable_actions()
            else:
                alarm.disable_actions()
            logger.info(
                "%s actions for alarm %s.",
                "Enabled" if enable else "Disabled",
                alarm_name,
            )
```

```
        except ClientError:
            logger.exception(
                "Couldn't %s actions alarm %s.",
                "enable" if enable else "disable",
                alarm_name,
            )
            raise
```

- Para obtener información sobre las API, consulte DisableAlarmActions en la Referencia de la API de AWS SDK para Python (Boto3).

Ruby

SDK para Ruby

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
# Disables an alarm in Amazon CloudWatch.
#
# Prerequisites.
#
# - The alarm to disable.
#
# @param cloudwatch_client [Aws::CloudWatch::Client]
#   An initialized CloudWatch client.
# @param alarm_name [String] The name of the alarm to disable.
# @return [Boolean] true if the alarm was disabled; otherwise, false.
# @example
#   exit 1 unless alarm_actions_disabled?(
#     Aws::CloudWatch::Client.new(region: 'us-east-1'),
#     'ObjectsInBucket'
#   )
def alarm_actions_disabled?(cloudwatch_client, alarm_name)
  cloudwatch_client.disable_alarm_actions(alarm_names: [alarm_name])
  return true
```

```
rescue StandardError => e
  puts "Error disabling alarm actions: #{e.message}"
  return false
end

# Example usage:
def run_me
  alarm_name = "ObjectsInBucket"
  alarm_description = "Objects exist in this bucket for more than 1 day."
  metric_name = "NumberOfObjects"
  # Notify this Amazon Simple Notification Service (Amazon SNS) topic when
  # the alarm transitions to the ALARM state.
  alarm_actions = ["arn:aws:sns:us-
east-1:111111111111:Default_CloudWatch_Alarms_Topic"]
  namespace = "AWS/S3"
  statistic = "Average"
  dimensions = [
    {
      name: "BucketName",
      value: "doc-example-bucket"
    },
    {
      name: "StorageType",
      value: "AllStorageTypes"
    }
  ]
  period = 86_400 # Daily (24 hours * 60 minutes * 60 seconds = 86400 seconds).
  unit = "Count"
  evaluation_periods = 1 # More than one day.
  threshold = 1 # One object.
  comparison_operator = "GreaterThanThreshold" # More than one object.
  # Replace us-west-2 with the AWS Region you're using for Amazon CloudWatch.
  region = "us-east-1"

  cloudwatch_client = Aws::CloudWatch::Client.new(region: region)

  if alarm_created_or_updated?(
    cloudwatch_client,
    alarm_name,
    alarm_description,
    metric_name,
    alarm_actions,
    namespace,
    statistic,
```

```
    dimensions,
    period,
    unit,
    evaluation_periods,
    threshold,
    comparison_operator
  )
    puts "Alarm '#{alarm_name}' created or updated."
  else
    puts "Could not create or update alarm '#{alarm_name}'."
  end

  if alarm_actions_disabled?(cloudwatch_client, alarm_name)
    puts "Alarm '#{alarm_name}' disabled."
  else
    puts "Could not disable alarm '#{alarm_name}'."
  end
end

run_me if $PROGRAM_NAME == __FILE__
```

- Para obtener información sobre las API, consulte DisableAlarmActions en la Referencia de la API de AWS SDK for Ruby.

SAP ABAP

SDK de SAP ABAP

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    "Disables actions on the specified alarm. "
    TRY.
        lo_cwt->disablealarmactions(
          it_alarmnames = it_alarm_names
        ).
```

```
        MESSAGE 'Alarm actions disabled.' TYPE 'I'.
      CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
        DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
        MESSAGE lv_error TYPE 'E'.
    ENDTRY.
```

- Para más información de las API, consulte DisableAlarmActions en la Referencia de API del SDK AWS para SAP ABAP.

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de CloudWatch con SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Uso de **`EnableAlarmActions`** con un AWS SDK o la CLI

Los siguientes ejemplos de código muestran cómo utilizar `EnableAlarmActions`.

Los ejemplos de acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Puede ver esta acción en contexto en el siguiente ejemplo de código:

- Gestionar métricas y alarmas

.NET

AWS SDK for .NET

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    /// <summary>
    /// Enable the actions for a list of alarms from CloudWatch.
    /// </summary>
    /// <param name="alarmNames">A list of names of alarms.</param>
    /// <returns>True if successful.</returns>
```

```
    public async Task<bool> EnableAlarmActions(List<string> alarmNames)
    {
        var enableAlarmActionsResult = await
_amazonCloudWatch.EnableAlarmActionsAsync(
            new EnableAlarmActionsRequest()
            {
                AlarmNames = alarmNames
            });

        return enableAlarmActionsResult.HttpStatusCode == HttpStatusCode.OK;
    }
```

- Para obtener información sobre las API, consulte EnableAlarmActions en la Referencia de la API de AWS SDK for .NET.

C++

SDK para C++

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Incluir los archivos requeridos.

```
#include <aws/core/Aws.h>
#include <aws/monitoring/CloudWatchClient.h>
#include <aws/monitoring/model/EnableAlarmActionsRequest.h>
#include <aws/monitoring/model/PutMetricAlarmRequest.h>
#include <iostream>
```

Habilite las acciones de alarma.

```
    Aws::CloudWatch::CloudWatchClient cw;
    Aws::CloudWatch::Model::PutMetricAlarmRequest request;
    request.SetAlarmName(alarm_name);
```

```
    request.SetComparisonOperator(
        Aws::CloudWatch::Model::ComparisonOperator::GreaterThanThreshold);
    request.SetEvaluationPeriods(1);
    request.SetMetricName("CPUUtilization");
    request.SetNamespace("AWS/EC2");
    request.SetPeriod(60);
    request.SetStatistic(Aws::CloudWatch::Model::Statistic::Average);
    request.SetThreshold(70.0);
    request.SetActionsEnabled(false);
    request.SetAlarmDescription("Alarm when server CPU exceeds 70%");
    request.SetUnit(Aws::CloudWatch::Model::StandardUnit::Seconds);
    request.AddAlarmActions(actionArn);

    Aws::CloudWatch::Model::Dimension dimension;
    dimension.SetName("InstanceId");
    dimension.SetValue(instanceId);
    request.AddDimensions(dimension);

    auto outcome = cw.PutMetricAlarm(request);
    if (!outcome.IsSuccess())
    {
        std::cout << "Failed to create CloudWatch alarm:" <<
            outcome.GetError().GetMessage() << std::endl;
        return;
    }

    Aws::CloudWatch::Model::EnableAlarmActionsRequest enable_request;
    enable_request.AddAlarmNames(alarm_name);

    auto enable_outcome = cw.EnableAlarmActions(enable_request);
    if (!enable_outcome.IsSuccess())
    {
        std::cout << "Failed to enable alarm actions:" <<
            enable_outcome.GetError().GetMessage() << std::endl;
        return;
    }

    std::cout << "Successfully created alarm " << alarm_name <<
        " and enabled actions on it." << std::endl;
```

- Para obtener información sobre las API, consulte EnableAlarmActions en la Referencia de la API de AWS SDK for C++.

CLI

AWS CLI

Activación de todas las acciones de una alarma

En el siguiente ejemplo, se utiliza el comando `enable-alarm-actions` para activar todas las acciones de la alarma denominada myalarm.:

```
aws cloudwatch enable-alarm-actions --alarm-names myalarm
```

Este comando vuelve a la petición si se ejecuta correctamente.

- Para obtener información sobre las API, consulte EnableAlarmActions en la Referencia de comandos de la AWS CLI.

Java

SDK para Java 2.x

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.cloudwatch.CloudWatchClient;
import software.amazon.awssdk.services.cloudwatch.model.CloudWatchException;
import
 software.amazon.awssdk.services.cloudwatch.model.EnableAlarmActionsRequest;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class EnableAlarmActions {
```

```
    public static void main(String[] args) {
        final String usage = """

                Usage:
                  <alarmName>

                Where:
                  alarmName - An alarm name to enable (for example, MyAlarm).
                """;

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }

        String alarm = args[0];
        Region region = Region.US_EAST_1;
        CloudWatchClient cw = CloudWatchClient.builder()
                .region(region)
                .build();

        enableActions(cw, alarm);
        cw.close();
    }

    public static void enableActions(CloudWatchClient cw, String alarm) {
        try {
            EnableAlarmActionsRequest request =
 EnableAlarmActionsRequest.builder()
                    .alarmNames(alarm)
                    .build();

            cw.enableAlarmActions(request);
            System.out.printf("Successfully enabled actions on alarm %s", alarm);

        } catch (CloudWatchException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- Para obtener información sobre las API, consulte EnableAlarmActions en la Referencia de la API de AWS SDK for Java 2.x.

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Importar el SDK y los módulos de cliente, y llamar a la API.

```
import { EnableAlarmActionsCommand } from "@aws-sdk/client-cloudwatch";
import { client } from "../libs/client.js";

const run = async () => {
  const command = new EnableAlarmActionsCommand({
    AlarmNames: [process.env.CLOUDWATCH_ALARM_NAME], // Set the value of
 CLOUDWATCH_ALARM_NAME to the name of an existing alarm.
  });

  try {
    return await client.send(command);
  } catch (err) {
    console.error(err);
  }
};

export default run();
```

Cree el cliente en un módulo separado y expórtelo.

```
import { CloudWatchClient } from "@aws-sdk/client-cloudwatch";

export const client = new CloudWatchClient({});
```

- Para obtener información, consulte la Guía para desarrolladores de AWS SDK for JavaScript.
- Para obtener información sobre las API, consulte EnableAlarmActions en la Referencia de la API de AWS SDK for JavaScript.

SDK para JavaScript (v2)

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Importar el SDK y los módulos de cliente, y llamar a la API.

```
// Load the AWS SDK for Node.js
var AWS = require("aws-sdk");
// Set the region
AWS.config.update({ region: "REGION" });

// Create CloudWatch service object
var cw = new AWS.CloudWatch({ apiVersion: "2010-08-01" });

var params = {
  AlarmName: "Web_Server_CPU_Utilization",
  ComparisonOperator: "GreaterThanThreshold",
  EvaluationPeriods: 1,
  MetricName: "CPUUtilization",
  Namespace: "AWS/EC2",
  Period: 60,
  Statistic: "Average",
  Threshold: 70.0,
  ActionsEnabled: true,
  AlarmActions: ["ACTION_ARN"],
  AlarmDescription: "Alarm when server CPU exceeds 70%",
  Dimensions: [
    {
      Name: "InstanceId",
      Value: "INSTANCE_ID",
    },
  ],
  Unit: "Percent",
```

```
};

cw.putMetricAlarm(params, function (err, data) {
  if (err) {
    console.log("Error", err);
  } else {
    console.log("Alarm action added", data);
    var paramsEnableAlarmAction = {
      AlarmNames: [params.AlarmName],
    };
    cw.enableAlarmActions(paramsEnableAlarmAction, function (err, data) {
      if (err) {
        console.log("Error", err);
      } else {
        console.log("Alarm action enabled", data);
      }
    });
  }
});
```

- Para obtener información, consulte la Guía para desarrolladores de AWS SDK for JavaScript.
- Para obtener información sobre las API, consulte EnableAlarmActions en la Referencia de la API de AWS SDK for JavaScript.

Kotlin

SDK para Kotlin

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
suspend fun enableActions(alarm: String) {

    val request = EnableAlarmActionsRequest {
        alarmNames = listOf(alarm)
    }
```

```
    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        cwClient.enableAlarmActions(request)
        println("Successfully enabled actions on alarm $alarm")
    }
}
```

- Para obtener información sobre las API, consulte EnableAlarmActions en la Referencia de la API de AWS SDK for Kotlin.

Python

SDK para Python (Boto3)

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
class CloudWatchWrapper:
    """Encapsulates Amazon CloudWatch functions."""

    def __init__(self, cloudwatch_resource):
        """
        :param cloudwatch_resource: A Boto3 CloudWatch resource.
        """
        self.cloudwatch_resource = cloudwatch_resource


    def enable_alarm_actions(self, alarm_name, enable):
        """
        Enables or disables actions on the specified alarm. Alarm actions can be
        used to send notifications or automate responses when an alarm enters a
        particular state.

        :param alarm_name: The name of the alarm.
        :param enable: When True, actions are enabled for the alarm. Otherwise,
 they
                       disabled.
```

```
        """
        try:
            alarm = self.cloudwatch_resource.Alarm(alarm_name)
            if enable:
                alarm.enable_actions()
            else:
                alarm.disable_actions()
            logger.info(
                "%s actions for alarm %s.",
                "Enabled" if enable else "Disabled",
                alarm_name,
            )
        except ClientError:
            logger.exception(
                "Couldn't %s actions alarm %s.",
                "enable" if enable else "disable",
                alarm_name,
            )
            raise
```

- Para obtener información sobre las API, consulte EnableAlarmActions en la Referencia de la API de AWS SDK para Python (Boto3).

SAP ABAP

SDK de SAP ABAP

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
"Enable actions on the specified alarm."
TRY.
    lo_cwt->enablealarmactions(
      it_alarmnames = it_alarm_names
    ).
```

```
        MESSAGE 'Alarm actions enabled.' TYPE 'I'.
      CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
        DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
        MESSAGE lv_error TYPE 'E'.
    ENDTRY.
```

- Para más información acerca de las API, consulte EnableAlarmActions en la Referencia de API del SDK AWS para SAP ABAP.

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de CloudWatch con SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Uso de **`GetDashboard`** con un AWS SDK o la CLI

Los siguientes ejemplos de código muestran cómo utilizar `GetDashboard`.

.NET

AWS SDK for .NET

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    /// <summary>
    /// Get information on a dashboard.
    /// </summary>
    /// <param name="dashboardName">The name of the dashboard.</param>
    /// <returns>A JSON object with dashboard information.</returns>
    public async Task<string> GetDashboard(string dashboardName)
    {
        var dashboardResponse = await _amazonCloudWatch.GetDashboardAsync(
            new GetDashboardRequest()
            {
                DashboardName = dashboardName
```

```
            });

        return dashboardResponse.DashboardBody;
    }
```

- Para obtener información sobre las API, consulte GetDashboard en la Referencia de la API de AWS SDK for .NET.

PowerShell

Herramientas para PowerShell

Ejemplo 1: devuelve el arn al cuerpo del panel especificado.

```
Get-CWDashboard -DashboardName Dashboard1
```

Salida:

```
DashboardArn                                              DashboardBody
------------                                              -------------
arn:aws:cloudwatch::123456789012:dashboard/Dashboard1 {...
```

- Para obtener información sobre la API, consulte GetDashboard en la Referencia de Cmdlet de AWS Tools for PowerShell.

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de CloudWatch con SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Uso de **`GetMetricData`** con un AWS SDK o la CLI

Los siguientes ejemplos de código muestran cómo utilizar `GetMetricData`.

Los ejemplos de acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Puede ver esta acción en contexto en el siguiente ejemplo de código:

- Primeros pasos para usar las métricas, los paneles y las alarmas

.NET

AWS SDK for .NET

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    /// <summary>
    /// Get data for CloudWatch metrics.
    /// </summary>
    /// <param name="minutesOfData">The number of minutes of data to include.</
param>
    /// <param name="useDescendingTime">True to return the data descending by
 time.</param>
    /// <param name="endDateUtc">The end date for the data, in UTC.</param>
    /// <param name="maxDataPoints">The maximum data points to include.</param>
    /// <param name="dataQueries">Optional data queries to include.</param>
    /// <returns>A list of the requested metric data.</returns>
    public async Task<List<MetricDataResult>> GetMetricData(int minutesOfData,
 bool useDescendingTime, DateTime? endDateUtc = null,
        int maxDataPoints = 0, List<MetricDataQuery>? dataQueries = null)
    {
        var metricData = new List<MetricDataResult>();
        // If no end time is provided, use the current time for the end time.
        endDateUtc ??= DateTime.UtcNow;
        var timeZoneOffset =
 TimeZoneInfo.Local.GetUtcOffset(endDateUtc.Value.ToLocalTime());
        var startTimeUtc = endDateUtc.Value.AddMinutes(-minutesOfData);
        // The timezone string should be in the format +0000, so use the timezone
 offset to format it correctly.
        var timeZoneString = $"{timeZoneOffset.Hours:D2}
{timeZoneOffset.Minutes:D2}";
        var paginatedMetricData = _amazonCloudWatch.Paginators.GetMetricData(
            new GetMetricDataRequest()
            {
                StartTimeUtc = startTimeUtc,
                EndTimeUtc = endDateUtc.Value,
                LabelOptions = new LabelOptions { Timezone = timeZoneString },
```

```
                ScanBy = useDescendingTime ? ScanBy.TimestampDescending :
ScanBy.TimestampAscending,
                MaxDatapoints = maxDataPoints,
                MetricDataQueries = dataQueries,
            });

        await foreach (var data in paginatedMetricData.MetricDataResults)
        {
            metricData.Add(data);
        }
        return metricData;
    }
```

- Para obtener información de la API, consulte GetMetricData en la Referencia de la API de AWS SDK for .NET.

Java

SDK para Java 2.x

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    public static void getCustomMetricData(CloudWatchClient cw, String fileName)
{
        try {
            // Read values from the JSON file.
            JsonParser parser = new JsonFactory().createParser(new
File(fileName));
            com.fasterxml.jackson.databind.JsonNode rootNode = new
ObjectMapper().readTree(parser);
            String customMetricNamespace =
rootNode.findValue("customMetricNamespace").asText();
            String customMetricName =
rootNode.findValue("customMetricName").asText();

            // Set the date.
```

```
            Instant nowDate = Instant.now();

            long hours = 1;
            long minutes = 30;
            Instant date2 = nowDate.plus(hours, ChronoUnit.HOURS).plus(minutes,
                    ChronoUnit.MINUTES);

            Metric met = Metric.builder()
                    .metricName(customMetricName)
                    .namespace(customMetricNamespace)
                    .build();

            MetricStat metStat = MetricStat.builder()
                    .stat("Maximum")
                    .period(1)
                    .metric(met)
                    .build();

            MetricDataQuery dataQUery = MetricDataQuery.builder()
                    .metricStat(metStat)
                    .id("foo2")
                    .returnData(true)
                    .build();

            List<MetricDataQuery> dq = new ArrayList<>();
            dq.add(dataQUery);

            GetMetricDataRequest getMetReq = GetMetricDataRequest.builder()
                    .maxDatapoints(10)
                    .scanBy(ScanBy.TIMESTAMP_DESCENDING)
                    .startTime(nowDate)
                    .endTime(date2)
                    .metricDataQueries(dq)
                    .build();

            GetMetricDataResponse response = cw.getMetricData(getMetReq);
            List<MetricDataResult> data = response.metricDataResults();
            for (MetricDataResult item : data) {
                System.out.println("The label is " + item.label());
                System.out.println("The status code is " +
item.statusCode().toString());
            }

        } catch (CloudWatchException | IOException e) {
```

```
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }
```

- Para obtener información de la API, consulte GetMetricData en la Referencia de la API de AWS SDK for Java 2.x.

Kotlin

SDK para Kotlin

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
suspend fun getCustomMetricData(fileName: String) {
    // Read values from the JSON file.
    val parser = JsonFactory().createParser(File(fileName))
    val rootNode = ObjectMapper().readTree<JsonNode>(parser)
    val customMetricNamespace =
 rootNode.findValue("customMetricNamespace").asText()
    val customMetricName = rootNode.findValue("customMetricName").asText()

    // Set the date.
    val nowDate = Instant.now()
    val hours: Long = 1
    val minutes: Long = 30
    val date2 = nowDate.plus(hours, ChronoUnit.HOURS).plus(
        minutes,
        ChronoUnit.MINUTES
    )

    val met = Metric {
        metricName = customMetricName
        namespace = customMetricNamespace
    }
```

```
    val metStat = MetricStat {
        stat = "Maximum"
        period = 1
        metric = met
    }

    val dataQUery = MetricDataQuery {
        metricStat = metStat
        id = "foo2"
        returnData = true
    }

    val dq = ArrayList<MetricDataQuery>()
    dq.add(dataQUery)
    val getMetReq = GetMetricDataRequest {
        maxDatapoints = 10
        scanBy = ScanBy.TimestampDescending
        startTime = aws.smithy.kotlin.runtime.time.Instant(nowDate)
        endTime = aws.smithy.kotlin.runtime.time.Instant(date2)
        metricDataQueries = dq
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        val response = cwClient.getMetricData(getMetReq)
        response.metricDataResults?.forEach { item ->
            println("The label is ${item.label}")
            println("The status code is ${item.statusCode}")
        }
    }
}
```

- Para obtener información acerca de la API, consulte GetMetricData en la Referencia de la API del SDK de AWS para Kotlin.

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de CloudWatch con SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Uso de **`GetMetricStatistics`** con un AWS SDK o la CLI

Los siguientes ejemplos de código muestran cómo utilizar `GetMetricStatistics`.

Los ejemplos de acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Puede ver esta acción en su contexto en los siguientes ejemplos de código:

- Primeros pasos para usar las métricas, los paneles y las alarmas
- Gestionar métricas y alarmas

.NET

AWS SDK for .NET

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    /// <summary>
    /// Get billing statistics using a call to a wrapper class.
    /// </summary>
    /// <returns>A collection of billing statistics.</returns>
    private static async Task<List<Datapoint>> SetupBillingStatistics()
    {
        // Make a request for EstimatedCharges with a period of one day for the
past seven days.
        var billingStatistics = await _cloudWatchWrapper.GetMetricStatistics(
            "AWS/Billing",
            "EstimatedCharges",
            new List<string>() { "Maximum" },
            new List<Dimension>() { new Dimension { Name = "Currency", Value =
"USD" } },
            7,
            86400);

        billingStatistics = billingStatistics.OrderBy(n => n.Timestamp).ToList();

        return billingStatistics;
    }

    /// <summary>
    /// Wrapper to get statistics for a specific CloudWatch metric.
    /// </summary>
```

```
    /// <param name="metricNamespace">The namespace of the metric.</param>
    /// <param name="metricName">The name of the metric.</param>
    /// <param name="statistics">The list of statistics to include.</param>
    /// <param name="dimensions">The list of dimensions to include.</param>
    /// <param name="days">The number of days in the past to include.</param>
    /// <param name="period">The period for the data.</param>
    /// <returns>A list of DataPoint objects for the statistics.</returns>
    public async Task<List<Datapoint>> GetMetricStatistics(string
metricNamespace,
        string metricName, List<string> statistics, List<Dimension> dimensions,
int days, int period)
    {
        var metricStatistics = await _amazonCloudWatch.GetMetricStatisticsAsync(
            new GetMetricStatisticsRequest()
            {
                Namespace = metricNamespace,
                MetricName = metricName,
                Dimensions = dimensions,
                Statistics = statistics,
                StartTimeUtc = DateTime.UtcNow.AddDays(-days),
                EndTimeUtc = DateTime.UtcNow,
                Period = period
            });

        return metricStatistics.Datapoints;
    }
```

- Para ver la información de la API, consulte GetMetricStatistics en la Referencia de la API de AWS SDK for .NET.

CLI

AWS CLI

Obtención de la utilización de la CPU por cada instancia de EC2

En el siguiente ejemplo, se utiliza el comando `get-metric-statistics` para obtener la utilización de la CPU para una instancia de EC2 con el ID i-abcdef.

```
aws cloudwatch get-metric-statistics --metric-name CPUUtilization --start-time
 2014-04-08T23:18:00Z --end-time 2014-04-09T23:18:00Z --period 3600 --namespace
 AWS/EC2 --statistics Maximum --dimensions Name=InstanceId,Value=i-abcdef
```

Salida:

```
{
    "Datapoints": [
        {
            "Timestamp": "2014-04-09T11:18:00Z",
            "Maximum": 44.79,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T20:18:00Z",
            "Maximum": 47.92,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T19:18:00Z",
            "Maximum": 50.85,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T09:18:00Z",
            "Maximum": 47.92,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T03:18:00Z",
            "Maximum": 76.84,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T21:18:00Z",
            "Maximum": 48.96,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T14:18:00Z",
            "Maximum": 47.92,
            "Unit": "Percent"
        },
```

```
        {
            "Timestamp": "2014-04-09T08:18:00Z",
            "Maximum": 47.92,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T16:18:00Z",
            "Maximum": 45.55,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T06:18:00Z",
            "Maximum": 47.92,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T13:18:00Z",
            "Maximum": 45.08,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T05:18:00Z",
            "Maximum": 47.92,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T18:18:00Z",
            "Maximum": 46.88,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T17:18:00Z",
            "Maximum": 52.08,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T07:18:00Z",
            "Maximum": 47.92,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T02:18:00Z",
            "Maximum": 51.23,
            "Unit": "Percent"
```

```
        },
        {
            "Timestamp": "2014-04-09T12:18:00Z",
            "Maximum": 47.67,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-08T23:18:00Z",
            "Maximum": 46.88,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T10:18:00Z",
            "Maximum": 51.91,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T04:18:00Z",
            "Maximum": 47.13,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T15:18:00Z",
            "Maximum": 48.96,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T00:18:00Z",
            "Maximum": 48.16,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T01:18:00Z",
            "Maximum": 49.18,
            "Unit": "Percent"
        }
    ],
    "Label": "CPUUtilization"
}
```


Especificación de varias dimensiones

En el siguiente ejemplo, se ilustra cómo especificar varias dimensiones. Cada dimensión se especifica mediante un par nombre/valor, con una coma entre el nombre y el valor. Cuando existen varias dimensiones se separan con un espacio. Si una métrica incluye varias dimensiones, debe especificar un valor para cada dimensión definida.

Para ver más ejemplos del uso del comando `get-metric-statistics`, consulte Obtener estadísticas para una métrica en la Guía para desarrolladores de Amazon CloudWatch.

```
aws cloudwatch get-metric-statistics --metric-name Buffers --
namespace MyNameSpace --dimensions Name=InstanceID,Value=i-abcdef
 Name=InstanceType,Value=m1.small --start-time 2016-10-15T04:00:00Z --end-time
 2016-10-19T07:00:00Z --statistics Average --period 60
```

- Para obtener información sobre la API, consulte GetMetricStatistics en la Referencia de comandos de la AWS CLI.

Java

SDK para Java 2.x

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
   public static void getAndDisplayMetricStatistics(CloudWatchClient cw, String
nameSpace, String metVal,
           String metricOption, String date, Dimension myDimension) {
       try {
           Instant start = Instant.parse(date);
           Instant endDate = Instant.now();

           GetMetricStatisticsRequest statisticsRequest =
GetMetricStatisticsRequest.builder()
                   .endTime(endDate)
                   .startTime(start)
                   .dimensions(myDimension)
                   .metricName(metVal)
                   .namespace(nameSpace)
```

```
                    .period(86400)
                    .statistics(Statistic.fromValue(metricOption))
                    .build();

            GetMetricStatisticsResponse response =
 cw.getMetricStatistics(statisticsRequest);
            List<Datapoint> data = response.datapoints();
            if (!data.isEmpty()) {
                for (Datapoint datapoint : data) {
                    System.out
                            .println("Timestamp: " + datapoint.timestamp() + "
 Maximum value: " + datapoint.maximum());
                }
            } else {
                System.out.println("The returned data list is empty");
            }

        } catch (CloudWatchException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }
```

- Para ver la información de la API, consulte GetMetricStatistics en la Referencia de la API de AWS SDK for Java 2.x.

Kotlin

SDK para Kotlin

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
suspend fun getAndDisplayMetricStatistics(nameSpaceVal: String, metVal: String,
 metricOption: String, date: String, myDimension: Dimension) {
    val start = Instant.parse(date)
    val endDate = Instant.now()
```

```
    val statisticsRequest = GetMetricStatisticsRequest {
        endTime = aws.smithy.kotlin.runtime.time.Instant(endDate)
        startTime = aws.smithy.kotlin.runtime.time.Instant(start)
        dimensions = listOf(myDimension)
        metricName = metVal
        namespace = nameSpaceVal
        period = 86400
        statistics = listOf(Statistic.fromValue(metricOption))
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        val response = cwClient.getMetricStatistics(statisticsRequest)
        val data = response.datapoints
        if (data != null) {
            if (data.isNotEmpty()) {
                for (datapoint in data) {
                    println("Timestamp: ${datapoint.timestamp} Maximum value:
 ${datapoint.maximum}")
                }
            } else {
                println("The returned data list is empty")
            }
        }
    }
}
```

- Para obtener información sobre la API, consulte GetMetricStatistics en la Referencia de la API de AWS SDK para Kotlin.

Python

SDK para Python (Boto3)

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
class CloudWatchWrapper:
```

```
    """Encapsulates Amazon CloudWatch functions."""

    def __init__(self, cloudwatch_resource):
        """
        :param cloudwatch_resource: A Boto3 CloudWatch resource.
        """
        self.cloudwatch_resource = cloudwatch_resource


    def get_metric_statistics(self, namespace, name, start, end, period,
stat_types):
        """
        Gets statistics for a metric within a specified time span. Metrics are
grouped
        into the specified period.

        :param namespace: The namespace of the metric.
        :param name: The name of the metric.
        :param start: The UTC start time of the time span to retrieve.
        :param end: The UTC end time of the time span to retrieve.
        :param period: The period, in seconds, in which to group metrics. The
period
                       must match the granularity of the metric, which depends on
                       the metric's age. For example, metrics that are older than
                       three hours have a one-minute granularity, so the period
must
                       be at least 60 and must be a multiple of 60.
        :param stat_types: The type of statistics to retrieve, such as average
value
                           or maximum value.
        :return: The retrieved statistics for the metric.
        """
        try:
            metric = self.cloudwatch_resource.Metric(namespace, name)
            stats = metric.get_statistics(
                StartTime=start, EndTime=end, Period=period,
Statistics=stat_types
            )
            logger.info(
                "Got %s statistics for %s.", len(stats["Datapoints"]),
stats["Label"]
            )
        except ClientError:
```

```
            logger.exception("Couldn't get statistics for %s.%s.", namespace,
name)
            raise
        else:
            return stats
```

- Para obtener información sobre las API, consulte GetMetricStatistics en la Referencia de la API de AWS para Python (Boto3).

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de CloudWatch con SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Uso de **`GetMetricWidgetImage`** con un AWS SDK o la CLI

Los siguientes ejemplos de código muestran cómo utilizar `GetMetricWidgetImage`.

Los ejemplos de acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Puede ver esta acción en contexto en el siguiente ejemplo de código:

- Primeros pasos para usar las métricas, los paneles y las alarmas

.NET

AWS SDK for .NET

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    /// <summary>
    /// Get an image for a metric graphed over time.
    /// </summary>
    /// <param name="metricNamespace">The namespace of the metric.</param>
    /// <param name="metric">The name of the metric.</param>
    /// <param name="stat">The name of the stat to chart.</param>
```

```
    /// <param name="period">The period to use for the chart.</param>
    /// <returns>A memory stream for the chart image.</returns>
    public async Task<MemoryStream> GetTimeSeriesMetricImage(string
metricNamespace, string metric, string stat, int period)
    {
        var metricImageWidget = new
        {
            title = "Example Metric Graph",
            view = "timeSeries",
            stacked = false,
            period = period,
            width = 1400,
            height = 600,
            metrics = new List<List<object>>
                { new() { metricNamespace, metric, new { stat } } }
        };

        var metricImageWidgetString =
JsonSerializer.Serialize(metricImageWidget);
        var imageResponse = await _amazonCloudWatch.GetMetricWidgetImageAsync(
            new GetMetricWidgetImageRequest()
            {
                MetricWidget = metricImageWidgetString
            });

        return imageResponse.MetricWidgetImage;
    }

    /// <summary>
    /// Save a metric image to a file.
    /// </summary>
    /// <param name="memoryStream">The MemoryStream for the metric image.</param>
    /// <param name="metricName">The name of the metric.</param>
    /// <returns>The path to the file.</returns>
    public string SaveMetricImage(MemoryStream memoryStream, string metricName)
    {
        var metricFileName = $"{metricName}_{DateTime.Now.Ticks}.png";
        using var sr = new StreamReader(memoryStream);
        // Writes the memory stream to a file.
        File.WriteAllBytes(metricFileName, memoryStream.ToArray());
        var filePath = Path.Join(AppDomain.CurrentDomain.BaseDirectory,
            metricFileName);
        return filePath;
    }
```

- Para obtener información sobre la API, consulte GetMetricWidgetImage en la Referencia de la API de AWS SDK for .NET.

Java

SDK para Java 2.x

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
   public static void getAndOpenMetricImage(CloudWatchClient cw, String
fileName) {
       System.out.println("Getting Image data for custom metric.");
       try {
           String myJSON = "{\n" +
                   "  \"title\": \"Example Metric Graph\",\n" +
                   "  \"view\": \"timeSeries\",\n" +
                   "  \"stacked \": false,\n" +
                   "  \"period\": 10,\n" +
                   "  \"width\": 1400,\n" +
                   "  \"height\": 600,\n" +
                   "  \"metrics\": [\n" +
                   "    [\n" +
                   "      \"AWS/Billing\",\n" +
                   "      \"EstimatedCharges\",\n" +
                   "      \"Currency\",\n" +
                   "      \"USD\"\n" +
                   "    ]\n" +
                   "  ]\n" +
                   "}";

           GetMetricWidgetImageRequest imageRequest =
GetMetricWidgetImageRequest.builder()
                   .metricWidget(myJSON)
                   .build();
```

```
            GetMetricWidgetImageResponse response =
 cw.getMetricWidgetImage(imageRequest);
            SdkBytes sdkBytes = response.metricWidgetImage();
            byte[] bytes = sdkBytes.asByteArray();
            File outputFile = new File(fileName);
            try (FileOutputStream outputStream = new
 FileOutputStream(outputFile)) {
                outputStream.write(bytes);
            }

        } catch (CloudWatchException | IOException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }
```

- Para obtener información sobre la API, consulte GetMetricWidgetImage en la Referencia de la API de AWS SDK for Java 2.x.

Kotlin

SDK para Kotlin

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
suspend fun getAndOpenMetricImage(fileName: String) {
    println("Getting Image data for custom metric.")
    val myJSON = """{
        "title": "Example Metric Graph",
        "view": "timeSeries",
        "stacked ": false,
        "period": 10,
        "width": 1400,
        "height": 600,
        "metrics": [
            [
```

```
            "AWS/Billing",
            "EstimatedCharges",
            "Currency",
            "USD"
            ]
        ]
        }"""

    val imageRequest = GetMetricWidgetImageRequest {
        metricWidget = myJSON
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        val response = cwClient.getMetricWidgetImage(imageRequest)
        val bytes = response.metricWidgetImage
        if (bytes != null) {
            File(fileName).writeBytes(bytes)
        }
    }
    println("You have successfully written data to $fileName")
}
```

- Para obtener información acerca de la API, consulte GetMetricWidgetImage en la Referencia de la API del SDK de AWS para Kotlin.

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de CloudWatch con SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Uso de **`ListDashboards`** con un AWS SDK o la CLI

Los siguientes ejemplos de código muestran cómo utilizar `ListDashboards`.

.NET

AWS SDK for .NET

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
/// <summary>
/// Get a list of dashboards.
/// </summary>
/// <returns>A list of DashboardEntry objects.</returns>
public async Task<List<DashboardEntry>> ListDashboards()
{
    var results = new List<DashboardEntry>();
    var paginateDashboards = _amazonCloudWatch.Paginators.ListDashboards(
        new ListDashboardsRequest());
    // Get the entire list using the paginator.
    await foreach (var data in paginateDashboards.DashboardEntries)
    {
        results.Add(data);
    }

    return results;
}
```

- Para obtener información sobre las API, consulte ListDashboards en la Referencia de la API de AWS SDK for .NET.

Java

SDK para Java 2.x

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    public static void listDashboards(CloudWatchClient cw) {
        try {
            ListDashboardsIterable listRes = cw.listDashboardsPaginator();
            listRes.stream()
                    .flatMap(r -> r.dashboardEntries().stream())
                    .forEach(entry -> {
                        System.out.println("Dashboard name is: " +
entry.dashboardName());
                        System.out.println("Dashboard ARN is: " +
entry.dashboardArn());
                    });

        } catch (CloudWatchException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
```

- Para obtener información sobre las API, consulte ListDashboards en la Referencia de la API de AWS SDK for Java 2.x.

Kotlin

SDK para Kotlin

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
suspend fun listDashboards() {
    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        cwClient.listDashboardsPaginated({})
            .transform { it.dashboardEntries?.forEach { obj -> emit(obj) } }
            .collect { obj ->
                println("Name is ${obj.dashboardName}")
                println("Dashboard ARN is ${obj.dashboardArn}")
            }
    }
```

```
}
```

- Para obtener información sobre la API, consulte ListDashboards en la Referencia de la API de AWS SDK para Kotlin.

PowerShell

Herramientas para PowerShell

Ejemplo 1: devuelve el conjunto de paneles de su cuenta.

```
Get-CWDashboardList
```

Salida:

```
DashboardArn DashboardName LastModified        Size
------------ ------------- ------------        ----
arn:...      Dashboard1    7/6/2017 8:14:15 PM 252
```

Ejemplo 2: devuelve el conjunto de paneles de su cuenta cuyos nombres comienzan con el prefijo “dev”.

```
Get-CWDashboardList -DashboardNamePrefix dev
```

- Para obtener información sobre la API, consulte ListDashboards en la Referencia de Cmdlet de AWS Tools for PowerShell.

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de CloudWatch con SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Uso de **`ListMetrics`** con un AWS SDK o la CLI

Los siguientes ejemplos de código muestran cómo utilizar `ListMetrics`.

Los ejemplos de acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Puede ver esta acción en su contexto en los siguientes ejemplos de código:

- Primeros pasos para usar las métricas, los paneles y las alarmas

- Gestionar métricas y alarmas

.NET

AWS SDK for .NET

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
   /// <summary>
   /// List metrics available, optionally within a namespace.
   /// </summary>
   /// <param name="metricNamespace">Optional CloudWatch namespace to use when
listing metrics.</param>
   /// <param name="filter">Optional dimension filter.</param>
   /// <param name="metricName">Optional metric name filter.</param>
   /// <returns>The list of metrics.</returns>
   public async Task<List<Metric>> ListMetrics(string? metricNamespace = null,
DimensionFilter? filter = null, string? metricName = null)
   {
       var results = new List<Metric>();
       var paginateMetrics = _amazonCloudWatch.Paginators.ListMetrics(
           new ListMetricsRequest
           {
               Namespace = metricNamespace,
               Dimensions = filter != null ? new List<DimensionFilter>
{ filter } : null,
               MetricName = metricName
           });
       // Get the entire list using the paginator.
       await foreach (var metric in paginateMetrics.Metrics)
       {
           results.Add(metric);
       }

       return results;
   }
```

- Para obtener información sobre las API, consulte ListMetrics en la Referencia de la API de AWS SDK for .NET.

C++

SDK para C++

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Incluir los archivos requeridos.

```
#include <aws/core/Aws.h>
#include <aws/monitoring/CloudWatchClient.h>
#include <aws/monitoring/model/ListMetricsRequest.h>
#include <aws/monitoring/model/ListMetricsResult.h>
#include <iomanip>
#include <iostream>
```

Enumere las métricas.

```
        Aws::CloudWatch::CloudWatchClient cw;
        Aws::CloudWatch::Model::ListMetricsRequest request;

        if (argc > 1)
        {
            request.SetMetricName(argv[1]);
        }

        if (argc > 2)
        {
            request.SetNamespace(argv[2]);
        }

        bool done = false;
        bool header = false;
        while (!done)
```

```
        {
            auto outcome = cw.ListMetrics(request);
            if (!outcome.IsSuccess())
            {
                std::cout << "Failed to list CloudWatch metrics:" <<
                    outcome.GetError().GetMessage() << std::endl;
                break;
            }

            if (!header)
            {
                std::cout << std::left << std::setw(48) << "MetricName" <<
                    std::setw(32) << "Namespace" << "DimensionNameValuePairs" <<
                    std::endl;
                header = true;
            }

            const auto &metrics = outcome.GetResult().GetMetrics();
            for (const auto &metric : metrics)
            {
                std::cout << std::left << std::setw(48) <<
                    metric.GetMetricName() << std::setw(32) <<
                    metric.GetNamespace();
                const auto &dimensions = metric.GetDimensions();
                for (auto iter = dimensions.cbegin();
                    iter != dimensions.cend(); ++iter)
                {
                    const auto &dimkv = *iter;
                    std::cout << dimkv.GetName() << " = " << dimkv.GetValue();
                    if (iter + 1 != dimensions.cend())
                    {
                        std::cout << ", ";
                    }
                }
                std::cout << std::endl;
            }

            const auto &next_token = outcome.GetResult().GetNextToken();
            request.SetNextToken(next_token);
            done = next_token.empty();
        }
```

- Para obtener información sobre las API, consulte ListMetrics en la Referencia de la API de AWS SDK for C++.

CLI

AWS CLI

Creación de una lista de las métricas de Amazon SNS

En el siguiente ejemplo de `list-metrics`, se muestran las métricas de Amazon SNS.

```
aws cloudwatch list-metrics \
    --namespace "AWS/SNS"
```

Salida:

```
{
    "Metrics": [
        {
            "Namespace": "AWS/SNS",
            "Dimensions": [
                {
                    "Name": "TopicName",
                    "Value": "NotifyMe"
                }
            ],
            "MetricName": "PublishSize"
        },
        {
            "Namespace": "AWS/SNS",
            "Dimensions": [
                {
                    "Name": "TopicName",
                    "Value": "CFO"
                }
            ],
            "MetricName": "PublishSize"
        },
        {
            "Namespace": "AWS/SNS",
            "Dimensions": [
                {
```

```
                    "Name": "TopicName",
                    "Value": "NotifyMe"
                }
            ],
            "MetricName": "NumberOfNotificationsFailed"
        },
        {
            "Namespace": "AWS/SNS",
            "Dimensions": [
                {
                    "Name": "TopicName",
                    "Value": "NotifyMe"
                }
            ],
            "MetricName": "NumberOfNotificationsDelivered"
        },
        {
            "Namespace": "AWS/SNS",
            "Dimensions": [
                {
                    "Name": "TopicName",
                    "Value": "NotifyMe"
                }
            ],
            "MetricName": "NumberOfMessagesPublished"
        },
        {
            "Namespace": "AWS/SNS",
            "Dimensions": [
                {
                    "Name": "TopicName",
                    "Value": "CFO"
                }
            ],
            "MetricName": "NumberOfMessagesPublished"
        },
        {
            "Namespace": "AWS/SNS",
            "Dimensions": [
                {
                    "Name": "TopicName",
                    "Value": "CFO"
                }
            ],
```

```
            "MetricName": "NumberOfNotificationsDelivered"
        },
        {
            "Namespace": "AWS/SNS",
            "Dimensions": [
                {
                    "Name": "TopicName",
                    "Value": "CFO"
                }
            ],
            "MetricName": "NumberOfNotificationsFailed"
        }
    ]
}
```

- Para obtener información sobre la API, consulte ListMetrics en la Referencia de comandos de la AWS CLI.

Java

SDK para Java 2.x

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.cloudwatch.CloudWatchClient;
import software.amazon.awssdk.services.cloudwatch.model.CloudWatchException;
import software.amazon.awssdk.services.cloudwatch.model.ListMetricsRequest;
import software.amazon.awssdk.services.cloudwatch.model.ListMetricsResponse;
import software.amazon.awssdk.services.cloudwatch.model.Metric;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
```

```
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class ListMetrics {
    public static void main(String[] args) {
        final String usage = """

                Usage:
                  <namespace>\s

                Where:
                  namespace - The namespace to filter against (for example, AWS/
EC2).\s
                """;

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }

        String namespace = args[0];
        Region region = Region.US_EAST_1;
        CloudWatchClient cw = CloudWatchClient.builder()
                .region(region)
                .build();

        listMets(cw, namespace);
        cw.close();
    }

    public static void listMets(CloudWatchClient cw, String namespace) {
        boolean done = false;
        String nextToken = null;

        try {
            while (!done) {

                ListMetricsResponse response;
                if (nextToken == null) {
                    ListMetricsRequest request = ListMetricsRequest.builder()
                            .namespace(namespace)
                            .build();

                    response = cw.listMetrics(request);
```

```
                } else {
                    ListMetricsRequest request = ListMetricsRequest.builder()
                            .namespace(namespace)
                            .nextToken(nextToken)
                            .build();

                    response = cw.listMetrics(request);
                }

                for (Metric metric : response.metrics()) {
                    System.out.printf("Retrieved metric %s",
 metric.metricName());
                    System.out.println();
                }

                if (response.nextToken() == null) {
                    done = true;
                } else {
                    nextToken = response.nextToken();
                }
            }

        } catch (CloudWatchException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- Para obtener información sobre las API, consulte ListMetrics en la Referencia de la API de AWS SDK for Java 2.x.

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Importar el SDK y los módulos de cliente, y llamar a la API.

```
import { ListMetricsCommand } from "@aws-sdk/client-cloudwatch";
import { client } from "../libs/client.js";

export const main = () => {
  // Use the AWS console to see available namespaces and metric names. Custom
 metrics can also be created.
  // https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/
viewing_metrics_with_cloudwatch.html
  const command = new ListMetricsCommand({
    Dimensions: [
      {
        Name: "LogGroupName",
      },
    ],
    MetricName: "IncomingLogEvents",
    Namespace: "AWS/Logs",
  });

  return client.send(command);
};
```

Cree el cliente en un módulo separado y expórtelo.

```
import { CloudWatchClient } from "@aws-sdk/client-cloudwatch";

export const client = new CloudWatchClient({});
```

- Para obtener información, consulte la Guía para desarrolladores de AWS SDK for JavaScript.
- Para obtener información sobre las API, consulte ListMetrics en la Referencia de la API de AWS SDK for JavaScript.

SDK para JavaScript (v2)

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
// Load the AWS SDK for Node.js
var AWS = require("aws-sdk");
// Set the region
AWS.config.update({ region: "REGION" });

// Create CloudWatch service object
var cw = new AWS.CloudWatch({ apiVersion: "2010-08-01" });

var params = {
  Dimensions: [
    {
      Name: "LogGroupName" /* required */,
    },
  ],
  MetricName: "IncomingLogEvents",
  Namespace: "AWS/Logs",
};

cw.listMetrics(params, function (err, data) {
  if (err) {
    console.log("Error", err);
  } else {
    console.log("Metrics", JSON.stringify(data.Metrics));
  }
});
```

- Para obtener información, consulte la Guía para desarrolladores de AWS SDK for JavaScript.
- Para obtener información sobre las API, consulte ListMetrics en la Referencia de la API de AWS SDK for JavaScript.

Kotlin

SDK para Kotlin

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
suspend fun listMets(namespaceVal: String?): ArrayList<String>? {
    val metList = ArrayList<String>()
    val request = ListMetricsRequest {
        namespace = namespaceVal
    }
    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        val reponse = cwClient.listMetrics(request)
        reponse.metrics?.forEach { metrics ->
            val data = metrics.metricName
            if (!metList.contains(data)) {
                metList.add(data!!)
            }
        }
    }
    return metList
}
```

- Para obtener información sobre la API, consulte ListMetrics en la Referencia de la API deAWS SDK para Kotlin.

Python

SDK para Python (Boto3)

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
class CloudWatchWrapper:
    """Encapsulates Amazon CloudWatch functions."""

    def __init__(self, cloudwatch_resource):
        """
        :param cloudwatch_resource: A Boto3 CloudWatch resource.
        """
        self.cloudwatch_resource = cloudwatch_resource


    def list_metrics(self, namespace, name, recent=False):
        """
        Gets the metrics within a namespace that have the specified name.
        If the metric has no dimensions, a single metric is returned.
        Otherwise, metrics for all dimensions are returned.

        :param namespace: The namespace of the metric.
        :param name: The name of the metric.
        :param recent: When True, only metrics that have been active in the last
                       three hours are returned.
        :return: An iterator that yields the retrieved metrics.
        """
        try:
            kwargs = {"Namespace": namespace, "MetricName": name}
            if recent:
                kwargs["RecentlyActive"] = "PT3H"  # List past 3 hours only
            metric_iter = self.cloudwatch_resource.metrics.filter(**kwargs)
            logger.info("Got metrics for %s.%s.", namespace, name)
        except ClientError:
            logger.exception("Couldn't get metrics for %s.%s.", namespace, name)
            raise
        else:
            return metric_iter
```

- Para obtener información sobre las API, consulte ListMetrics en la Referencia de la API de AWS SDK para Python (Boto3).

Ruby

SDK para Ruby

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
# Lists available metrics for a metric namespace in Amazon CloudWatch.
#
# @param cloudwatch_client [Aws::CloudWatch::Client]
#   An initialized CloudWatch client.
# @param metric_namespace [String] The namespace of the metric.
# @example
#   list_metrics_for_namespace(
#     Aws::CloudWatch::Client.new(region: 'us-east-1'),
#     'SITE/TRAFFIC'
#   )
def list_metrics_for_namespace(cloudwatch_client, metric_namespace)
  response = cloudwatch_client.list_metrics(namespace: metric_namespace)

  if response.metrics.count.positive?
    response.metrics.each do |metric|
      puts "  Metric name: #{metric.metric_name}"
      if metric.dimensions.count.positive?
        puts "    Dimensions:"
        metric.dimensions.each do |dimension|
          puts "      Name: #{dimension.name}, Value: #{dimension.value}"
        end
      else
        puts "No dimensions found."
      end
    end
  else
    puts "No metrics found for namespace '#{metric_namespace}'. " \
      "Note that it could take up to 15 minutes for recently-added metrics " \
      "to become available."
  end
end
```

```
# Example usage:
def run_me
  metric_namespace = "SITE/TRAFFIC"
  # Replace us-west-2 with the AWS Region you're using for Amazon CloudWatch.
  region = "us-east-1"

  cloudwatch_client = Aws::CloudWatch::Client.new(region: region)

  # Add three datapoints.
  puts "Continuing..." unless datapoint_added_to_metric?(
    cloudwatch_client,
    metric_namespace,
    "UniqueVisitors",
    "SiteName",
    "example.com",
    5_885.0,
    "Count"
  )

  puts "Continuing..." unless datapoint_added_to_metric?(
    cloudwatch_client,
    metric_namespace,
    "UniqueVisits",
    "SiteName",
    "example.com",
    8_628.0,
    "Count"
  )

  puts "Continuing..." unless datapoint_added_to_metric?(
    cloudwatch_client,
    metric_namespace,
    "PageViews",
    "PageURL",
    "example.html",
    18_057.0,
    "Count"
  )

  puts "Metrics for namespace '#{metric_namespace}':"
  list_metrics_for_namespace(cloudwatch_client, metric_namespace)
end

run_me if $PROGRAM_NAME == __FILE__
```

- Para obtener información sobre las API, consulte ListMetrics en la Referencia de la API de AWS SDK for Ruby.

SAP ABAP

SDK de SAP ABAP

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
   "The following list-metrics example displays the metrics for Amazon
CloudWatch."
   TRY.
       oo_result = lo_cwt->listmetrics(            " oo_result is returned for
testing purposes. "
         iv_namespace = iv_namespace
       ).
       DATA(lt_metrics) = oo_result->get_metrics( ).
       MESSAGE 'Metrics retrieved.' TYPE 'I'.
     CATCH /aws1/cx_cwtinvparamvalueex .
       MESSAGE 'The specified argument was not valid.' TYPE 'E'.
   ENDTRY.
```

- Para más detalles sobre la API, consulte ListMetrics en la referencia de API del SDK AWS para SAP ABAP.

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de CloudWatch con SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Uso de **`PutAnomalyDetector`** con un AWS SDK o la CLI

Los siguientes ejemplos de código muestran cómo utilizar `PutAnomalyDetector`.

Los ejemplos de acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Puede ver esta acción en contexto en el siguiente ejemplo de código:

- Primeros pasos para usar las métricas, los paneles y las alarmas

.NET

AWS SDK for .NET

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
   /// <summary>
   /// Add an anomaly detector for a single metric.
   /// </summary>
   /// <param name="anomalyDetector">A single metric anomaly detector.</param>
   /// <returns>True if successful.</returns>
   public async Task<bool> PutAnomalyDetector(SingleMetricAnomalyDetector
anomalyDetector)
   {
       var putAlarmDetectorResult = await
_amazonCloudWatch.PutAnomalyDetectorAsync(
           new PutAnomalyDetectorRequest()
           {
               SingleMetricAnomalyDetector = anomalyDetector
           });

       return putAlarmDetectorResult.HttpStatusCode == HttpStatusCode.OK;
   }
```

- Para obtener información sobre la API, consulte PutAnomalyDetector en la Referencia de la API de AWS SDK for .NET.

Java

SDK para Java 2.x

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
   public static void addAnomalyDetector(CloudWatchClient cw, String fileName) {
       try {
           // Read values from the JSON file.
           JsonParser parser = new JsonFactory().createParser(new
File(fileName));
           com.fasterxml.jackson.databind.JsonNode rootNode = new
ObjectMapper().readTree(parser);
           String customMetricNamespace =
rootNode.findValue("customMetricNamespace").asText();
           String customMetricName =
rootNode.findValue("customMetricName").asText();

           SingleMetricAnomalyDetector singleMetricAnomalyDetector =
SingleMetricAnomalyDetector.builder()
                   .metricName(customMetricName)
                   .namespace(customMetricNamespace)
                   .stat("Maximum")
                   .build();

           PutAnomalyDetectorRequest anomalyDetectorRequest =
PutAnomalyDetectorRequest.builder()
                   .singleMetricAnomalyDetector(singleMetricAnomalyDetector)
                   .build();

           cw.putAnomalyDetector(anomalyDetectorRequest);
           System.out.println("Added anomaly detector for metric " +
customMetricName + ".");

       } catch (CloudWatchException | IOException e) {
           System.err.println(e.getMessage());
           System.exit(1);
       }
```

```
    }
```

- Para obtener información sobre la API, consulte PutAnomalyDetector en la Referencia de la API de AWS SDK for Java 2.x.

Kotlin

SDK para Kotlin

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
suspend fun addAnomalyDetector(fileName: String?) {
    // Read values from the JSON file.
    val parser = JsonFactory().createParser(File(fileName))
    val rootNode = ObjectMapper().readTree<JsonNode>(parser)
    val customMetricNamespace =
 rootNode.findValue("customMetricNamespace").asText()
    val customMetricName = rootNode.findValue("customMetricName").asText()

    val singleMetricAnomalyDetectorVal = SingleMetricAnomalyDetector {
        metricName = customMetricName
        namespace = customMetricNamespace
        stat = "Maximum"
    }

    val anomalyDetectorRequest = PutAnomalyDetectorRequest {
        singleMetricAnomalyDetector = singleMetricAnomalyDetectorVal
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        cwClient.putAnomalyDetector(anomalyDetectorRequest)
        println("Added anomaly detector for metric $customMetricName.")
    }
}
```

- Para obtener más detalles de las API, consulte PutAnomalyDetector en la Referencia de API de SDK de AWS para Kotlin.

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de CloudWatch con SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Uso de **PutDashboard** con un AWS SDK o la CLI

Los siguientes ejemplos de código muestran cómo utilizar `PutDashboard`.

Los ejemplos de acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Puede ver esta acción en contexto en el siguiente ejemplo de código:

- Primeros pasos para usar las métricas, los paneles y las alarmas

.NET

AWS SDK for .NET

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    /// <summary>
    /// Set up a dashboard using a call to the wrapper class.
    /// </summary>
    /// <param name="customMetricNamespace">The metric namespace.</param>
    /// <param name="customMetricName">The metric name.</param>
    /// <param name="dashboardName">The name of the dashboard.</param>
    /// <returns>A list of validation messages.</returns>
    private static async Task<List<DashboardValidationMessage>> SetupDashboard(
        string customMetricNamespace, string customMetricName, string
dashboardName)
    {
        // Get the dashboard model from configuration.
        var newDashboard = new DashboardModel();
```

```
        _configuration.GetSection("dashboardExampleBody").Bind(newDashboard);

        // Add a new metric to the dashboard.
        newDashboard.Widgets.Add(new Widget
        {
            Height = 8,
            Width = 8,
            Y = 8,
            X = 0,
            Type = "metric",
            Properties = new Properties
            {
                Metrics = new List<List<object>>
                    { new() { customMetricNamespace, customMetricName } },
                View = "timeSeries",
                Region = "us-east-1",
                Stat = "Sum",
                Period = 86400,
                YAxis = new YAxis { Left = new Left { Min = 0, Max = 100 } },
                Title = "Custom Metric Widget",
                LiveData = true,
                Sparkline = true,
                Trend = true,
                Stacked = false,
                SetPeriodToTimeRange = false
            }
        });

        var newDashboardString = JsonSerializer.Serialize(newDashboard,
            new JsonSerializerOptions
            { DefaultIgnoreCondition = JsonIgnoreCondition.WhenWritingNull });
        var validationMessages =
            await _cloudWatchWrapper.PutDashboard(dashboardName,
 newDashboardString);

        return validationMessages;
    }

    /// <summary>
    /// Wrapper to create or add to a dashboard with metrics.
    /// </summary>
    /// <param name="dashboardName">The name for the dashboard.</param>
    /// <param name="dashboardBody">The metric data in JSON for the dashboard.</
param>
```

```
    /// <returns>A list of validation messages for the dashboard.</returns>
    public async Task<List<DashboardValidationMessage>> PutDashboard(string
dashboardName,
        string dashboardBody)
    {
        // Updating a dashboard replaces all contents.
        // Best practice is to include a text widget indicating this dashboard
was created programmatically.
        var dashboardResponse = await _amazonCloudWatch.PutDashboardAsync(
            new PutDashboardRequest()
            {
                DashboardName = dashboardName,
                DashboardBody = dashboardBody
            });

        return dashboardResponse.DashboardValidationMessages;
    }
```

- Para obtener información sobre la API, consulte PutDashboard en la Referencia de la API de AWS SDK for .NET.

Java

SDK para Java 2.x

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    public static void createDashboardWithMetrics(CloudWatchClient cw, String
dashboardName, String fileName) {
        try {
            PutDashboardRequest dashboardRequest = PutDashboardRequest.builder()
                    .dashboardName(dashboardName)
                    .dashboardBody(readFileAsString(fileName))
                    .build();
```

```
            PutDashboardResponse response = cw.putDashboard(dashboardRequest);
            System.out.println(dashboardName + " was successfully created.");
            List<DashboardValidationMessage> messages =
 response.dashboardValidationMessages();
            if (messages.isEmpty()) {
                System.out.println("There are no messages in the new Dashboard");
            } else {
                for (DashboardValidationMessage message : messages) {
                    System.out.println("Message is: " + message.message());
                }
            }

        } catch (CloudWatchException | IOException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }
```

- Para obtener información sobre la API, consulte PutDashboard en la Referencia de la API de AWS SDK for Java 2.x.

Kotlin

SDK para Kotlin

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
suspend fun createDashboardWithMetrics(dashboardNameVal: String, fileNameVal:
 String) {
    val dashboardRequest = PutDashboardRequest {
        dashboardName = dashboardNameVal
        dashboardBody = readFileAsString(fileNameVal)
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        val response = cwClient.putDashboard(dashboardRequest)
```

```
        println("$dashboardNameVal was successfully created.")
        val messages = response.dashboardValidationMessages
        if (messages != null) {
            if (messages.isEmpty()) {
                println("There are no messages in the new Dashboard")
            } else {
                for (message in messages) {
                    println("Message is: ${message.message}")
                }
            }
        }
    }
}
```

- Para obtener información sobre la API, consulte PutDashboard en la Referencia de la API de AWS SDK para Kotlin.

PowerShell

Herramientas para PowerShell

Ejemplo 1: crea o actualiza el panel denominado “Dashboard1” para incluir dos widgets de métricas uno al lado del otro.

```
$dashBody = @"
{
    "widgets":[
        {
             "type":"metric",
             "x":0,
             "y":0,
             "width":12,
             "height":6,
             "properties":{
                "metrics":[
                   [
                      "AWS/EC2",
                      "CPUUtilization",
                      "InstanceId",
                      "i-012345"
                   ]
```

```
                ],
                "period":300,
                "stat":"Average",
                "region":"us-east-1",
                "title":"EC2 Instance CPU"
            }
        },
        {
            "type":"metric",
            "x":12,
            "y":0,
            "width":12,
            "height":6,
            "properties":{
                "metrics":[
                    [
                        "AWS/S3",
                        "BucketSizeBytes",
                        "BucketName",
                        "MyBucketName"
                    ]
                ],
                "period":86400,
                "stat":"Maximum",
                "region":"us-east-1",
                "title":"MyBucketName bytes"
            }
        }
    ]
}
"@

Write-CWDashboard -DashboardName Dashboard1 -DashboardBody $dashBody
```

Ejemplo 2: crea o actualiza el panel y canaliza el contenido que describe el panel al cmdlet.

```
$dashBody = @"
{
...
}
"@

$dashBody | Write-CWDashboard -DashboardName Dashboard1
```

- Para obtener información sobre la API, consulte PuDashboard en la Referencia de Cmdlet de AWS Tools for PowerShell.

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de CloudWatch con SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Uso de **`PutMetricAlarm`** con un AWS SDK o la CLI

Los siguientes ejemplos de código muestran cómo utilizar `PutMetricAlarm`.

Los ejemplos de acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Puede ver esta acción en su contexto en los siguientes ejemplos de código:

- Primeros pasos para usar alarmas
- Primeros pasos para usar las métricas, los paneles y las alarmas
- Gestionar métricas y alarmas

.NET

AWS SDK for .NET

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    /// <summary>
    /// Add a metric alarm to send an email when the metric passes a threshold.
    /// </summary>
    /// <param name="alarmDescription">A description of the alarm.</param>
    /// <param name="alarmName">The name for the alarm.</param>
    /// <param name="comparison">The type of comparison to use.</param>
    /// <param name="metricName">The name of the metric for the alarm.</param>
    /// <param name="metricNamespace">The namespace of the metric.</param>
    /// <param name="threshold">The threshold value for the alarm.</param>
    /// <param name="alarmActions">Optional actions to execute when in an alarm
state.</param>
```

```
   /// <returns>True if successful.</returns>
   public async Task<bool> PutMetricEmailAlarm(string alarmDescription, string
alarmName, ComparisonOperator comparison,
       string metricName, string metricNamespace, double threshold, List<string>
alarmActions = null!)
   {
       try
       {
           var putEmailAlarmResponse = await
_amazonCloudWatch.PutMetricAlarmAsync(
               new PutMetricAlarmRequest()
               {
                   AlarmActions = alarmActions,
                   AlarmDescription = alarmDescription,
                   AlarmName = alarmName,
                   ComparisonOperator = comparison,
                   Threshold = threshold,
                   Namespace = metricNamespace,
                   MetricName = metricName,
                   EvaluationPeriods = 1,
                   Period = 10,
                   Statistic = new Statistic("Maximum"),
                   DatapointsToAlarm = 1,
                   TreatMissingData = "ignore"
               });
           return putEmailAlarmResponse.HttpStatusCode == HttpStatusCode.OK;
       }
       catch (LimitExceededException lex)
       {
           _logger.LogError(lex, $"Unable to add alarm {alarmName}. Alarm quota
has already been reached.");
       }

       return false;
   }

   /// <summary>
   /// Add specific email actions to a list of action strings for a CloudWatch
alarm.
   /// </summary>
   /// <param name="accountId">The AccountId for the alarm.</param>
   /// <param name="region">The region for the alarm.</param>
   /// <param name="emailTopicName">An Amazon Simple Notification Service (SNS)
topic for the alarm email.</param>
```

```
    /// <param name="alarmActions">Optional list of existing alarm actions to
 append to.</param>
    /// <returns>A list of string actions for an alarm.</returns>
    public List<string> AddEmailAlarmAction(string accountId, string region,
        string emailTopicName, List<string>? alarmActions = null)
    {
        alarmActions ??= new List<string>();
        var snsAlarmAction = $"arn:aws:sns:{region}:{accountId}:
{emailTopicName}";
        alarmActions.Add(snsAlarmAction);
        return alarmActions;
    }
```

- Para obtener información sobre las API, consulte PutMetricAlarm en la Referencia de la API de AWS SDK for .NET.

C++

SDK para C++

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Incluir los archivos requeridos.

```
#include <aws/core/Aws.h>
#include <aws/monitoring/CloudWatchClient.h>
#include <aws/monitoring/model/PutMetricAlarmRequest.h>
#include <iostream>
```

Cree la alarma para ver la métrica.

```
        Aws::CloudWatch::CloudWatchClient cw;
        Aws::CloudWatch::Model::PutMetricAlarmRequest request;
        request.SetAlarmName(alarm_name);
```

```
        request.SetComparisonOperator(
            Aws::CloudWatch::Model::ComparisonOperator::GreaterThanThreshold);
        request.SetEvaluationPeriods(1);
        request.SetMetricName("CPUUtilization");
        request.SetNamespace("AWS/EC2");
        request.SetPeriod(60);
        request.SetStatistic(Aws::CloudWatch::Model::Statistic::Average);
        request.SetThreshold(70.0);
        request.SetActionsEnabled(false);
        request.SetAlarmDescription("Alarm when server CPU exceeds 70%");
        request.SetUnit(Aws::CloudWatch::Model::StandardUnit::Seconds);

        Aws::CloudWatch::Model::Dimension dimension;
        dimension.SetName("InstanceId");
        dimension.SetValue(instanceId);

        request.AddDimensions(dimension);

        auto outcome = cw.PutMetricAlarm(request);
        if (!outcome.IsSuccess())
        {
            std::cout << "Failed to create CloudWatch alarm:" <<
                outcome.GetError().GetMessage() << std::endl;
        }
        else
        {
            std::cout << "Successfully created CloudWatch alarm " << alarm_name
                << std::endl;
        }
```

- Para obtener información sobre las API, consulte PutMetricAlarm en la Referencia de la API de AWS SDK for C++.

CLI

AWS CLI

Envío de un mensaje por correo electrónico de Amazon Simple Notification Service cuando el uso de la CPU supere el 70 por ciento

El siguiente ejemplo usa el comando `put-metric-alarm` para enviar un mensaje por correo electrónico de Amazon Simple Notification Service cuando el uso de la CPU supere el 70 por ciento:

```
aws cloudwatch put-metric-alarm --alarm-name cpu-mon --alarm-description "Alarm
 when CPU exceeds 70 percent" --metric-name CPUUtilization --namespace AWS/
EC2 --statistic Average --period 300 --threshold 70 --comparison-operator
 GreaterThanThreshold  --dimensions "Name=InstanceId,Value=i-12345678" --
evaluation-periods 2 --alarm-actions arn:aws:sns:us-east-1:111122223333:MyTopic
 --unit Percent
```

Este comando vuelve a la petición si se ejecuta correctamente. Si existe una alarma con el mismo nombre, la alarma nueva la sobrescribirá.

Especificación de varias dimensiones

En el siguiente ejemplo, se ilustra cómo especificar varias dimensiones. Cada dimensión se especifica mediante un par nombre/valor, con una coma entre el nombre y el valor. Cuando existen varias dimensiones se separan con un espacio:

```
aws cloudwatch put-metric-alarm --alarm-name "Default_Test_Alarm3" --alarm-
description "The default example alarm" --namespace "CW EXAMPLE METRICS" --
metric-name Default_Test --statistic Average --period 60 --evaluation-periods 3
 --threshold 50 --comparison-operator GreaterThanOrEqualToThreshold --dimensions
 Name=key1,Value=value1 Name=key2,Value=value2
```

- Para obtener información sobre la API, consulte PutMetricAlarm en la Referencia de comandos de la AWS CLI.

Java

SDK para Java 2.x

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    public static String createAlarm(CloudWatchClient cw, String fileName) {
```

```
        try {
            // Read values from the JSON file.
            JsonParser parser = new JsonFactory().createParser(new
File(fileName));
            com.fasterxml.jackson.databind.JsonNode rootNode = new
ObjectMapper().readTree(parser);
            String customMetricNamespace =
rootNode.findValue("customMetricNamespace").asText();
            String customMetricName =
rootNode.findValue("customMetricName").asText();
            String alarmName = rootNode.findValue("exampleAlarmName").asText();
            String emailTopic = rootNode.findValue("emailTopic").asText();
            String accountId = rootNode.findValue("accountId").asText();
            String region = rootNode.findValue("region").asText();

            // Create a List for alarm actions.
            List<String> alarmActions = new ArrayList<>();
            alarmActions.add("arn:aws:sns:" + region + ":" + accountId + ":" +
emailTopic);
            PutMetricAlarmRequest alarmRequest = PutMetricAlarmRequest.builder()
                    .alarmActions(alarmActions)
                    .alarmDescription("Example metric alarm")
                    .alarmName(alarmName)

.comparisonOperator(ComparisonOperator.GREATER_THAN_OR_EQUAL_TO_THRESHOLD)
                    .threshold(100.00)
                    .metricName(customMetricName)
                    .namespace(customMetricNamespace)
                    .evaluationPeriods(1)
                    .period(10)
                    .statistic("Maximum")
                    .datapointsToAlarm(1)
                    .treatMissingData("ignore")
                    .build();

            cw.putMetricAlarm(alarmRequest);
            System.out.println(alarmName + " was successfully created!");
            return alarmName;

        } catch (CloudWatchException | IOException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
        return "";
```

```
    }
```

- Para obtener información sobre las API, consulte PutMetricAlarm en la Referencia de la API de AWS SDK for Java 2.x.

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Importar el SDK y los módulos de cliente, y llamar a la API.

```
import { PutMetricAlarmCommand } from "@aws-sdk/client-cloudwatch";
import { client } from "../libs/client.js";

const run = async () => {
  // This alarm triggers when CPUUtilization exceeds 70% for one minute.
  const command = new PutMetricAlarmCommand({
    AlarmName: process.env.CLOUDWATCH_ALARM_NAME, // Set the value of
 CLOUDWATCH_ALARM_NAME to the name of an existing alarm.
    ComparisonOperator: "GreaterThanThreshold",
    EvaluationPeriods: 1,
    MetricName: "CPUUtilization",
    Namespace: "AWS/EC2",
    Period: 60,
    Statistic: "Average",
    Threshold: 70.0,
    ActionsEnabled: false,
    AlarmDescription: "Alarm when server CPU exceeds 70%",
    Dimensions: [
      {
        Name: "InstanceId",
        Value: process.env.EC2_INSTANCE_ID, // Set the value of EC_INSTANCE_ID to
 the Id of an existing Amazon EC2 instance.
      },
    ],
```

```
    Unit: "Percent",
  });

  try {
    return await client.send(command);
  } catch (err) {
    console.error(err);
  }
};

export default run();
```

Cree el cliente en un módulo separado y expórtelo.

```
import { CloudWatchClient } from "@aws-sdk/client-cloudwatch";

export const client = new CloudWatchClient({});
```

- Para obtener información, consulte la Guía para desarrolladores de AWS SDK for JavaScript.
- Para obtener información sobre las API, consulte PutMetricAlarm en la Referencia de la API de AWS SDK for JavaScript.

SDK para JavaScript (v2)

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
// Load the AWS SDK for Node.js
var AWS = require("aws-sdk");
// Set the region
AWS.config.update({ region: "REGION" });

// Create CloudWatch service object
var cw = new AWS.CloudWatch({ apiVersion: "2010-08-01" });

var params = {
```

```
  AlarmName: "Web_Server_CPU_Utilization",
  ComparisonOperator: "GreaterThanThreshold",
  EvaluationPeriods: 1,
  MetricName: "CPUUtilization",
  Namespace: "AWS/EC2",
  Period: 60,
  Statistic: "Average",
  Threshold: 70.0,
  ActionsEnabled: false,
  AlarmDescription: "Alarm when server CPU exceeds 70%",
  Dimensions: [
    {
      Name: "InstanceId",
      Value: "INSTANCE_ID",
    },
  ],
  Unit: "Percent",
};

cw.putMetricAlarm(params, function (err, data) {
  if (err) {
    console.log("Error", err);
  } else {
    console.log("Success", data);
  }
});
```

- Para obtener información, consulte la Guía para desarrolladores de AWS SDK for JavaScript.
- Para obtener información sobre las API, consulte PutMetricAlarm en la Referencia de la API de AWS SDK for JavaScript.

Kotlin

SDK para Kotlin

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
suspend fun putMetricAlarm(alarmNameVal: String, instanceIdVal: String) {

    val dimensionOb = Dimension {
        name = "InstanceId"
        value = instanceIdVal
    }

    val request = PutMetricAlarmRequest {
        alarmName = alarmNameVal
        comparisonOperator = ComparisonOperator.GreaterThanThreshold
        evaluationPeriods = 1
        metricName = "CPUUtilization"
        namespace = "AWS/EC2"
        period = 60
        statistic = Statistic.fromValue("Average")
        threshold = 70.0
        actionsEnabled = false
        alarmDescription = "An Alarm created by the Kotlin SDK when server CPU
 utilization exceeds 70%"
        unit = StandardUnit.fromValue("Seconds")
        dimensions = listOf(dimensionOb)
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        cwClient.putMetricAlarm(request)
        println("Successfully created an alarm with name $alarmNameVal")
    }
}
```

- Para obtener información de las API, consulte PutMetricAlarm en Referencia de la API de SDK de AWS para Kotlin.

Python

SDK para Python (Boto3)

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
class CloudWatchWrapper:
    """Encapsulates Amazon CloudWatch functions."""

    def __init__(self, cloudwatch_resource):
        """
        :param cloudwatch_resource: A Boto3 CloudWatch resource.
        """
        self.cloudwatch_resource = cloudwatch_resource


    def create_metric_alarm(
        self,
        metric_namespace,
        metric_name,
        alarm_name,
        stat_type,
        period,
        eval_periods,
        threshold,
        comparison_op,
    ):
        """
        Creates an alarm that watches a metric.

        :param metric_namespace: The namespace of the metric.
        :param metric_name: The name of the metric.
        :param alarm_name: The name of the alarm.
        :param stat_type: The type of statistic the alarm watches.
        :param period: The period in which metric data are grouped to calculate
                       statistics.
        :param eval_periods: The number of periods that the metric must be over
 the
                             alarm threshold before the alarm is set into an
 alarmed
                             state.
        :param threshold: The threshold value to compare against the metric
 statistic.
        :param comparison_op: The comparison operation used to compare the
 threshold
                              against the metric.
        :return: The newly created alarm.
        """
        try:
```

```
            metric = self.cloudwatch_resource.Metric(metric_namespace,
metric_name)
            alarm = metric.put_alarm(
                AlarmName=alarm_name,
                Statistic=stat_type,
                Period=period,
                EvaluationPeriods=eval_periods,
                Threshold=threshold,
                ComparisonOperator=comparison_op,
            )
            logger.info(
                "Added alarm %s to track metric %s.%s.",
                alarm_name,
                metric_namespace,
                metric_name,
            )
        except ClientError:
            logger.exception(
                "Couldn't add alarm %s to metric %s.%s",
                alarm_name,
                metric_namespace,
                metric_name,
            )
            raise
        else:
            return alarm
```

- Para obtener información sobre las API, consulte PutMetricAlarm en la Referencia de la API de AWS SDK para Python (Boto3).

Ruby

SDK para Ruby

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
# Creates or updates an alarm in Amazon CloudWatch.
#
# @param cloudwatch_client [Aws::CloudWatch::Client]
#   An initialized CloudWatch client.
# @param alarm_name [String] The name of the alarm.
# @param alarm_description [String] A description about the alarm.
# @param metric_name [String] The name of the metric associated with the alarm.
# @param alarm_actions [Array] A list of Strings representing the
#   Amazon Resource Names (ARNs) to execute when the alarm transitions to the
#   ALARM state.
# @param namespace [String] The namespace for the metric to alarm on.
# @param statistic [String] The statistic for the metric.
# @param dimensions [Array] A list of dimensions for the metric, specified as
#   Aws::CloudWatch::Types::Dimension.
# @param period [Integer] The number of seconds before re-evaluating the metric.
# @param unit [String] The unit of measure for the statistic.
# @param evaluation_periods [Integer] The number of periods over which data is
#   compared to the specified threshold.
# @param theshold [Float] The value against which the specified statistic is
 compared.
# @param comparison_operator [String] The arithmetic operation to use when
#   comparing the specified statistic and threshold.
# @return [Boolean] true if the alarm was created or updated; otherwise, false.
# @example
#   exit 1 unless alarm_created_or_updated?(
#     Aws::CloudWatch::Client.new(region: 'us-east-1'),
#     'ObjectsInBucket',
#     'Objects exist in this bucket for more than 1 day.',
#     'NumberOfObjects',
#     ['arn:aws:sns:us-east-1:111111111111:Default_CloudWatch_Alarms_Topic'],
#     'AWS/S3',
#     'Average',
#     [
#       {
#         name: 'BucketName',
#         value: 'doc-example-bucket'
#       },
#       {
#         name: 'StorageType',
#         value: 'AllStorageTypes'
#       }
#     ],
#     86_400,
```

```
#     'Count',
#     1,
#     1,
#     'GreaterThanThreshold'
#   )
def alarm_created_or_updated?(
  cloudwatch_client,
  alarm_name,
  alarm_description,
  metric_name,
  alarm_actions,
  namespace,
  statistic,
  dimensions,
  period,
  unit,
  evaluation_periods,
  threshold,
  comparison_operator
)
  cloudwatch_client.put_metric_alarm(
    alarm_name: alarm_name,
    alarm_description: alarm_description,
    metric_name: metric_name,
    alarm_actions: alarm_actions,
    namespace: namespace,
    statistic: statistic,
    dimensions: dimensions,
    period: period,
    unit: unit,
    evaluation_periods: evaluation_periods,
    threshold: threshold,
    comparison_operator: comparison_operator
  )
  return true
rescue StandardError => e
  puts "Error creating alarm: #{e.message}"
  return false
end
```

- Para obtener información sobre las API, consulte PutMetricAlarm en la Referencia de la API de AWS SDK for Ruby.

SAP ABAP

SDK de SAP ABAP

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
TRY.
    lo_cwt->putmetricalarm(
      iv_alarmname                = iv_alarm_name
      iv_comparisonoperator       = iv_comparison_operator
      iv_evaluationperiods        = iv_evaluation_periods
      iv_metricname               = iv_metric_name
      iv_namespace                = iv_namespace
      iv_statistic                = iv_statistic
      iv_threshold                = iv_threshold
      iv_actionsenabled           = iv_actions_enabled
      iv_alarmdescription         = iv_alarm_description
      iv_unit                     = iv_unit
      iv_period                   = iv_period
      it_dimensions               = it_dimensions
    ).
    MESSAGE 'Alarm created.' TYPE 'I'.
  CATCH /aws1/cx_cwtlimitexceededfault.
    MESSAGE 'The request processing has exceeded the limit' TYPE 'E'.
ENDTRY.
```

- Para más información sobre las API, consulte PutMetricAlarm en la referencia de API del SDK AWS para SAP ABAP.

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de CloudWatch con SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Uso de **`PutMetricData`** con un AWS SDK o la CLI

Los siguientes ejemplos de código muestran cómo utilizar `PutMetricData`.

Los ejemplos de acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Puede ver esta acción en su contexto en los siguientes ejemplos de código:

- Primeros pasos para usar las métricas, los paneles y las alarmas
- Gestionar métricas y alarmas

.NET

AWS SDK for .NET

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
   /// <summary>
   /// Add some metric data using a call to a wrapper class.
   /// </summary>
   /// <param name="customMetricName">The metric name.</param>
   /// <param name="customMetricNamespace">The metric namespace.</param>
   /// <returns></returns>
   private static async Task<List<MetricDatum>> PutRandomMetricData(string
customMetricName,
       string customMetricNamespace)
   {
       List<MetricDatum> customData = new List<MetricDatum>();
       Random rnd = new Random();

       // Add 10 random values up to 100, starting with a timestamp 15 minutes
in the past.
       var utcNowMinus15 = DateTime.UtcNow.AddMinutes(-15);
       for (int i = 0; i < 10; i++)
       {
           var metricValue = rnd.Next(0, 100);
           customData.Add(
               new MetricDatum
               {
                   MetricName = customMetricName,
                   Value = metricValue,
```

```
                    TimestampUtc = utcNowMinus15.AddMinutes(i)
                }
            );
        }

        await _cloudWatchWrapper.PutMetricData(customMetricNamespace,
customData);
        return customData;
    }

    /// <summary>
    /// Wrapper to add metric data to a CloudWatch metric.
    /// </summary>
    /// <param name="metricNamespace">The namespace of the metric.</param>
    /// <param name="metricData">A data object for the metric data.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> PutMetricData(string metricNamespace,
        List<MetricDatum> metricData)
    {
        var putDataResponse = await _amazonCloudWatch.PutMetricDataAsync(
            new PutMetricDataRequest()
            {
                MetricData = metricData,
                Namespace = metricNamespace,
            });

        return putDataResponse.HttpStatusCode == HttpStatusCode.OK;
    }
```

- Para obtener información sobre las API, consulte PutMetricData en la Referencia de la API de AWS SDK for .NET.

C++

SDK para C++

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Incluir los archivos requeridos.

```
#include <aws/core/Aws.h>
#include <aws/monitoring/CloudWatchClient.h>
#include <aws/monitoring/model/PutMetricDataRequest.h>
#include <iostream>
```

Coloque datos en la métrica

```
        Aws::CloudWatch::CloudWatchClient cw;

        Aws::CloudWatch::Model::Dimension dimension;
        dimension.SetName("UNIQUE_PAGES");
        dimension.SetValue("URLS");

        Aws::CloudWatch::Model::MetricDatum datum;
        datum.SetMetricName("PAGES_VISITED");
        datum.SetUnit(Aws::CloudWatch::Model::StandardUnit::None);
        datum.SetValue(data_point);
        datum.AddDimensions(dimension);

        Aws::CloudWatch::Model::PutMetricDataRequest request;
        request.SetNamespace("SITE/TRAFFIC");
        request.AddMetricData(datum);

        auto outcome = cw.PutMetricData(request);
        if (!outcome.IsSuccess())
        {
            std::cout << "Failed to put sample metric data:" <<
                outcome.GetError().GetMessage() << std::endl;
        }
        else
        {
            std::cout << "Successfully put sample metric data" << std::endl;
        }
```

- Para obtener información sobre las API, consulte PutMetricData en la Referencia de la API de AWS SDK for C++.

CLI

AWS CLI

Publicación de métricas personalizadas en Amazon CloudWatch

En el siguiente ejemplo, se utiliza el comando `put-metric-data` para publicar una métrica personalizada en Amazon CloudWatch:

```
aws cloudwatch put-metric-data --namespace "Usage Metrics" --metric-data file://
metric.json
```

Los valores de la métrica en sí se almacenan en el archivo JSON, `metric.json`.

A continuación, se muestra el contenido de ese archivo:

```
[
  {
    "MetricName": "New Posts",
    "Timestamp": "Wednesday, June 12, 2013 8:28:20 PM",
    "Value": 0.50,
    "Unit": "Count"
  }
]
```

Para obtener más información, consulte Publicación de métricas personalizadas en la Guía para desarrolladores de Amazon CloudWatch.

Cómo especificar varias dimensiones

En el siguiente ejemplo, se ilustra cómo especificar varias dimensiones. Cada dimensión se especifica con un par Nombre=Valor. Cuando existen varias dimensiones se separan con una coma:

```
aws cloudwatch put-metric-data --metric-name Buffers --namespace
 MyNameSpace --unit Bytes --value 231434333 --dimensions
 InstanceID=1-23456789,InstanceType=m1.small
```

- Para obtener información sobre la API, consulte PutMetricData en la Referencia de comandos de la AWS CLI.

Java

SDK para Java 2.x

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
   public static void addMetricDataForAlarm(CloudWatchClient cw, String
fileName) {
       try {
           // Read values from the JSON file.
           JsonParser parser = new JsonFactory().createParser(new
File(fileName));
           com.fasterxml.jackson.databind.JsonNode rootNode = new
ObjectMapper().readTree(parser);
           String customMetricNamespace =
rootNode.findValue("customMetricNamespace").asText();
           String customMetricName =
rootNode.findValue("customMetricName").asText();

           // Set an Instant object.
           String time =
ZonedDateTime.now(ZoneOffset.UTC).format(DateTimeFormatter.ISO_INSTANT);
           Instant instant = Instant.parse(time);

           MetricDatum datum = MetricDatum.builder()
                   .metricName(customMetricName)
                   .unit(StandardUnit.NONE)
                   .value(1001.00)
                   .timestamp(instant)
                   .build();

           MetricDatum datum2 = MetricDatum.builder()
                   .metricName(customMetricName)
                   .unit(StandardUnit.NONE)
                   .value(1002.00)
                   .timestamp(instant)
                   .build();
```

```
            List<MetricDatum> metricDataList = new ArrayList<>();
            metricDataList.add(datum);
            metricDataList.add(datum2);

            PutMetricDataRequest request = PutMetricDataRequest.builder()
                    .namespace(customMetricNamespace)
                    .metricData(metricDataList)
                    .build();

            cw.putMetricData(request);
            System.out.println("Added metric values for for metric " +
customMetricName);

        } catch (CloudWatchException | IOException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }
```

- Para obtener información sobre las API, consulte PutMetricData en la Referencia de la API de AWS SDK for Java 2.x.

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Importar el SDK y los módulos de cliente, y llamar a la API.

```
import { PutMetricDataCommand } from "@aws-sdk/client-cloudwatch";
import { client } from "../libs/client.js";

const run = async () => {
  // See https://docs.aws.amazon.com/AmazonCloudWatch/latest/APIReference/
API_PutMetricData.html#API_PutMetricData_RequestParameters
```

```
  // and https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/
publishingMetrics.html
  // for more information about the parameters in this command.
  const command = new PutMetricDataCommand({
    MetricData: [
      {
        MetricName: "PAGES_VISITED",
        Dimensions: [
          {
            Name: "UNIQUE_PAGES",
            Value: "URLS",
          },
        ],
        Unit: "None",
        Value: 1.0,
      },
    ],
    Namespace: "SITE/TRAFFIC",
  });

  try {
    return await client.send(command);
  } catch (err) {
    console.error(err);
  }
};

export default run();
```

Cree el cliente en un módulo separado y expórtelo.

```
import { CloudWatchClient } from "@aws-sdk/client-cloudwatch";

export const client = new CloudWatchClient({});
```

- Para obtener información, consulte la Guía para desarrolladores de AWS SDK for JavaScript.
- Para obtener información sobre las API, consulte PutMetricData en la Referencia de la API de AWS SDK for JavaScript.

SDK para JavaScript (v2)

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
// Load the AWS SDK for Node.js
var AWS = require("aws-sdk");
// Set the region
AWS.config.update({ region: "REGION" });

// Create CloudWatch service object
var cw = new AWS.CloudWatch({ apiVersion: "2010-08-01" });

// Create parameters JSON for putMetricData
var params = {
  MetricData: [
    {
      MetricName: "PAGES_VISITED",
      Dimensions: [
        {
          Name: "UNIQUE_PAGES",
          Value: "URLS",
        },
      ],
      Unit: "None",
      Value: 1.0,
    },
  ],
  Namespace: "SITE/TRAFFIC",
};

cw.putMetricData(params, function (err, data) {
  if (err) {
    console.log("Error", err);
  } else {
    console.log("Success", JSON.stringify(data));
  }
});
```

- Para obtener información, consulte la Guía para desarrolladores de AWS SDK for JavaScript.
- Para obtener información sobre las API, consulte PutMetricData en la Referencia de la API de AWS SDK for JavaScript.

Kotlin

SDK para Kotlin

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
suspend fun addMetricDataForAlarm(fileName: String?) {
    // Read values from the JSON file.
    val parser = JsonFactory().createParser(File(fileName))
    val rootNode = ObjectMapper().readTree<JsonNode>(parser)
    val customMetricNamespace =
 rootNode.findValue("customMetricNamespace").asText()
    val customMetricName = rootNode.findValue("customMetricName").asText()

    // Set an Instant object.
    val time =
 ZonedDateTime.now(ZoneOffset.UTC).format(DateTimeFormatter.ISO_INSTANT)
    val instant = Instant.parse(time)
    val datum = MetricDatum {
        metricName = customMetricName
        unit = StandardUnit.None
        value = 1001.00
        timestamp = aws.smithy.kotlin.runtime.time.Instant(instant)
    }

    val datum2 = MetricDatum {
        metricName = customMetricName
        unit = StandardUnit.None
        value = 1002.00
        timestamp = aws.smithy.kotlin.runtime.time.Instant(instant)
    }
```

```
    val metricDataList = ArrayList<MetricDatum>()
    metricDataList.add(datum)
    metricDataList.add(datum2)

    val request = PutMetricDataRequest {
        namespace = customMetricNamespace
        metricData = metricDataList
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        cwClient.putMetricData(request)
        println("Added metric values for for metric $customMetricName")
    }
}
```

- Para obtener información sobre la API, consulte PutMetricData en la Referencia de la API de AWS SDK para Kotlin.

PowerShell

Herramientas para PowerShell

Ejemplo 1: crea un nuevo objeto MetricDatum y lo escribe en Métricas de CloudWatch de Amazon Web Services.

```
### Create a MetricDatum .NET object
$Metric = New-Object -TypeName Amazon.CloudWatch.Model.MetricDatum
$Metric.Timestamp = [DateTime]::UtcNow
$Metric.MetricName = 'CPU'
$Metric.Value = 50

### Write the metric data to the CloudWatch service
Write-CWMetricData -Namespace instance1 -MetricData $Metric
```

- Para obtener información sobre la API, consulte PutMetricData en la Referencia de Cmdlet de AWS Tools for PowerShell.

Python

SDK para Python (Boto3)

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
class CloudWatchWrapper:
    """Encapsulates Amazon CloudWatch functions."""

    def __init__(self, cloudwatch_resource):
        """
        :param cloudwatch_resource: A Boto3 CloudWatch resource.
        """
        self.cloudwatch_resource = cloudwatch_resource


    def put_metric_data(self, namespace, name, value, unit):
        """
        Sends a single data value to CloudWatch for a metric. This metric is
 given
        a timestamp of the current UTC time.

        :param namespace: The namespace of the metric.
        :param name: The name of the metric.
        :param value: The value of the metric.
        :param unit: The unit of the metric.
        """
        try:
            metric = self.cloudwatch_resource.Metric(namespace, name)
            metric.put_data(
                Namespace=namespace,
                MetricData=[{"MetricName": name, "Value": value, "Unit": unit}],
            )
            logger.info("Put data for metric %s.%s", namespace, name)
        except ClientError:
            logger.exception("Couldn't put data for metric %s.%s", namespace,
 name)
            raise
```

Coloque un conjunto de datos en una métrica de CloudWatch.

```
class CloudWatchWrapper:
    """Encapsulates Amazon CloudWatch functions."""

    def __init__(self, cloudwatch_resource):
        """
        :param cloudwatch_resource: A Boto3 CloudWatch resource.
        """
        self.cloudwatch_resource = cloudwatch_resource


    def put_metric_data_set(self, namespace, name, timestamp, unit, data_set):
        """
        Sends a set of data to CloudWatch for a metric. All of the data in the
 set
        have the same timestamp and unit.

        :param namespace: The namespace of the metric.
        :param name: The name of the metric.
        :param timestamp: The UTC timestamp for the metric.
        :param unit: The unit of the metric.
        :param data_set: The set of data to send. This set is a dictionary that
                         contains a list of values and a list of corresponding
 counts.
                         The value and count lists must be the same length.
        """
        try:
            metric = self.cloudwatch_resource.Metric(namespace, name)
            metric.put_data(
                Namespace=namespace,
                MetricData=[
                    {
                        "MetricName": name,
                        "Timestamp": timestamp,
                        "Values": data_set["values"],
                        "Counts": data_set["counts"],
                        "Unit": unit,
                    }
                ],
```

```
            )
            logger.info("Put data set for metric %s.%s.", namespace, name)
        except ClientError:
            logger.exception("Couldn't put data set for metric %s.%s.",
namespace, name)
            raise
```

- Para obtener información sobre las API, consulte PutMetricData en la Referencia de la API de AWS SDK para Python (Boto3).

Ruby

SDK para Ruby

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
require "aws-sdk-cloudwatch"

# Adds a datapoint to a metric in Amazon CloudWatch.
#
# @param cloudwatch_client [Aws::CloudWatch::Client]
#   An initialized CloudWatch client.
# @param metric_namespace [String] The namespace of the metric to add the
#   datapoint to.
# @param metric_name [String] The name of the metric to add the datapoint to.
# @param dimension_name [String] The name of the dimension to add the
#   datapoint to.
# @param dimension_value [String] The value of the dimension to add the
#   datapoint to.
# @param metric_value [Float] The value of the datapoint.
# @param metric_unit [String] The unit of measurement for the datapoint.
# @return [Boolean]
# @example
#   exit 1 unless datapoint_added_to_metric?(
#     Aws::CloudWatch::Client.new(region: 'us-east-1'),
```

```
#     'SITE/TRAFFIC',
#     'UniqueVisitors',
#     'SiteName',
#     'example.com',
#     5_885.0,
#     'Count'
#   )
def datapoint_added_to_metric?(
  cloudwatch_client,
  metric_namespace,
  metric_name,
  dimension_name,
  dimension_value,
  metric_value,
  metric_unit
)
  cloudwatch_client.put_metric_data(
    namespace: metric_namespace,
    metric_data: [
      {
        metric_name: metric_name,
        dimensions: [
          {
            name: dimension_name,
            value: dimension_value
          }
        ],
        value: metric_value,
        unit: metric_unit
      }
    ]
  )
  puts "Added data about '#{metric_name}' to namespace " \
    "'#{metric_namespace}'."
  return true
rescue StandardError => e
  puts "Error adding data about '#{metric_name}' to namespace " \
    "'#{metric_namespace}': #{e.message}"
  return false
end
```

- Para obtener información sobre las API, consulte PutMetricData en la Referencia de la API de AWS SDK for Ruby.

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de CloudWatch con SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Situaciones de CloudWatch usando los AWS SDK

Los siguientes ejemplos de código muestran cómo implementar escenarios habituales en CloudWatch con los SDK de AWS. Estos escenarios muestran cómo llevar a cabo tareas específicas llamando a varias funciones dentro de CloudWatch. En cada escenario se incluye un enlace a GitHub, con instrucciones de configuración y ejecución del código.

Ejemplos

- Primeros pasos para usar las alarmas de CloudWatch mediante un SDK de AWS
- Primeros pasos para usar las métricas de CloudWatch mediante un SDK de AWS
- Administración de métricas y alarmas de CloudWatch mediante un SDK de AWS

Primeros pasos para usar las alarmas de CloudWatch mediante un SDK de AWS

En el siguiente ejemplo de código, se muestra cómo:

- Cree una alarma.
- Desactive las acciones de alarma.
- Describa una alarma.
- Elimine una alarma.

SAP ABAP

SDK de SAP ABAP

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
DATA lt_alarmnames TYPE /aws1/cl_cwtalarmnames_w=>tt_alarmnames.
DATA lo_alarmname TYPE REF TO /aws1/cl_cwtalarmnames_w.

"Create an alarm"
TRY.
    lo_cwt->putmetricalarm(
      iv_alarmname                = iv_alarm_name
      iv_comparisonoperator       = iv_comparison_operator
      iv_evaluationperiods        = iv_evaluation_periods
      iv_metricname               = iv_metric_name
      iv_namespace                = iv_namespace
      iv_statistic                = iv_statistic
      iv_threshold                = iv_threshold
      iv_actionsenabled           = iv_actions_enabled
      iv_alarmdescription         = iv_alarm_description
      iv_unit                     = iv_unit
      iv_period                   = iv_period
      it_dimensions               = it_dimensions
    ).
    MESSAGE 'Alarm created' TYPE 'I'.
  CATCH /aws1/cx_cwtlimitexceededfault.
    MESSAGE 'The request processing has exceeded the limit' TYPE 'E'.
ENDTRY.

"Create an ABAP internal table for the created alarm."
CREATE OBJECT lo_alarmname EXPORTING iv_value = iv_alarm_name.
INSERT lo_alarmname INTO TABLE lt_alarmnames.

"Disable alarm actions."
TRY.
    lo_cwt->disablealarmactions(
```

```
          it_alarmnames               = lt_alarmnames
        ).
        MESSAGE 'Alarm actions disabled' TYPE 'I'.
      CATCH /aws1/cx_rt_service_generic INTO DATA(lo_disablealarm_exception).
        DATA(lv_disablealarm_error) = |"{ lo_disablealarm_exception-
>av_err_code }" - { lo_disablealarm_exception->av_err_msg }|.
        MESSAGE lv_disablealarm_error TYPE 'E'.
    ENDTRY.

    "Describe alarm using the same ABAP internal table."
    TRY.
        oo_result = lo_cwt->describealarms(                      " oo_result is
 returned for testing purpose "
          it_alarmnames               = lt_alarmnames
        ).
        MESSAGE 'Alarms retrieved' TYPE 'I'.
      CATCH /aws1/cx_rt_service_generic INTO DATA(lo_describealarms_exception).
        DATA(lv_describealarms_error) = |"{ lo_describealarms_exception-
>av_err_code }" - { lo_describealarms_exception->av_err_msg }|.
        MESSAGE lv_describealarms_error TYPE 'E'.
    ENDTRY.

    "Delete alarm."
    TRY.
        lo_cwt->deletealarms(
          it_alarmnames = lt_alarmnames
        ).
        MESSAGE 'Alarms deleted' TYPE 'I'.
      CATCH /aws1/cx_cwtresourcenotfound .
        MESSAGE 'Resource being access is not found.' TYPE 'E'.
    ENDTRY.
```

- Para detalles acerca de la API, consulte los siguientes temas en la Referencia de la API del SDK de AWS para SAP ABAP.
  - DeleteAlarms
  - DescribeAlarms
  - DisableAlarmActions
  - PutMetricAlarm

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de CloudWatch con SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Primeros pasos para usar las métricas de CloudWatch mediante un SDK de AWS

En el siguiente ejemplo de código, se muestra cómo:

- Enumere los espacios de nombres y las métricas de CloudWatch.
- Obtener estadísticas para una métrica y para la facturación estimada.
- Crear y actualizar un panel.
- Crear y agregar datos a una métrica.
- Crear y activar una alarma y, a continuación, consultar el historial de alarmas.
- Crear un detector de anomalías.
- Realice una imagen métrica y, luego, limpie los recursos.

.NET

AWS SDK for .NET

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Ejecutar un escenario interactivo en un símbolo del sistema.

```
public class CloudWatchScenario
{
    /*
    Before running this .NET code example, set up your development environment,
 including your credentials.

    To enable billing metrics and statistics for this example, make sure billing
 alerts are enabled for your account:
```

```
    https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/
monitor_estimated_charges_with_cloudwatch.html#turning_on_billing_metrics

    This .NET example performs the following tasks:
        1. List and select a CloudWatch namespace.
        2. List and select a CloudWatch metric.
        3. Get statistics for a CloudWatch metric.
        4. Get estimated billing statistics for the last week.
        5. Create a new CloudWatch dashboard with two metrics.
        6. List current CloudWatch dashboards.
        7. Create a CloudWatch custom metric and add metric data.
        8. Add the custom metric to the dashboard.
        9. Create a CloudWatch alarm for the custom metric.
       10. Describe current CloudWatch alarms.
       11. Get recent data for the custom metric.
       12. Add data to the custom metric to trigger the alarm.
       13. Wait for an alarm state.
       14. Get history for the CloudWatch alarm.
       15. Add an anomaly detector.
       16. Describe current anomaly detectors.
       17. Get and display a metric image.
       18. Clean up resources.
    */

    private static ILogger logger = null!;
    private static CloudWatchWrapper _cloudWatchWrapper = null!;
    private static IConfiguration _configuration = null!;
    private static readonly List<string> _statTypes = new List<string>
{ "SampleCount", "Average", "Sum", "Minimum", "Maximum" };
    private static SingleMetricAnomalyDetector? anomalyDetector = null!;

    static async Task Main(string[] args)
    {
        // Set up dependency injection for the Amazon service.
        using var host = Host.CreateDefaultBuilder(args)
            .ConfigureLogging(logging =>
                logging.AddFilter("System", LogLevel.Debug)
                    .AddFilter<DebugLoggerProvider>("Microsoft",
 LogLevel.Information)
                    .AddFilter<ConsoleLoggerProvider>("Microsoft",
 LogLevel.Trace))
            .ConfigureServices((_, services) =>
            services.AddAWSService<IAmazonCloudWatch>()
            .AddTransient<CloudWatchWrapper>()
```

```
        )
        .Build();

        _configuration = new ConfigurationBuilder()
            .SetBasePath(Directory.GetCurrentDirectory())
            .AddJsonFile("settings.json") // Load settings from .json file.
            .AddJsonFile("settings.local.json",
                true) // Optionally, load local settings.
            .Build();

        logger = LoggerFactory.Create(builder => { builder.AddConsole(); })
            .CreateLogger<CloudWatchScenario>();

        _cloudWatchWrapper =
host.Services.GetRequiredService<CloudWatchWrapper>();

        Console.WriteLine(new string('-', 80));
        Console.WriteLine("Welcome to the Amazon CloudWatch example scenario.");
        Console.WriteLine(new string('-', 80));

        try
        {
            var selectedNamespace = await SelectNamespace();
            var selectedMetric = await SelectMetric(selectedNamespace);
            await GetAndDisplayMetricStatistics(selectedNamespace,
selectedMetric);
            await GetAndDisplayEstimatedBilling();
            await CreateDashboardWithMetrics();
            await ListDashboards();
            await CreateNewCustomMetric();
            await AddMetricToDashboard();
            await CreateMetricAlarm();
            await DescribeAlarms();
            await GetCustomMetricData();
            await AddMetricDataForAlarm();
            await CheckForMetricAlarm();
            await GetAlarmHistory();
            anomalyDetector = await AddAnomalyDetector();
            await DescribeAnomalyDetectors();
            await GetAndOpenMetricImage();
            await CleanupResources();
        }
        catch (Exception ex)
        {
```

```
            logger.LogError(ex, "There was a problem executing the scenario.");
            await CleanupResources();
        }

    }

    /// <summary>
    /// Select a namespace.
    /// </summary>
    /// <returns>The selected namespace.</returns>
    private static async Task<string> SelectNamespace()
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine($"1. Select a CloudWatch Namespace from a list of
Namespaces.");
        var metrics = await _cloudWatchWrapper.ListMetrics();
        // Get a distinct list of namespaces.
        var namespaces = metrics.Select(m => m.Namespace).Distinct().ToList();
        for (int i = 0; i < namespaces.Count; i++)
        {
            Console.WriteLine($"\t{i + 1}. {namespaces[i]}");
        }

        var namespaceChoiceNumber = 0;
        while (namespaceChoiceNumber < 1 || namespaceChoiceNumber >
namespaces.Count)
        {
            Console.WriteLine(
                "Select a namespace by entering a number from the preceding
list:");
            var choice = Console.ReadLine();
            Int32.TryParse(choice, out namespaceChoiceNumber);
        }

        var selectedNamespace = namespaces[namespaceChoiceNumber - 1];

        Console.WriteLine(new string('-', 80));

        return selectedNamespace;
    }

    /// <summary>
    /// Select a metric from a namespace.
    /// </summary>
```

```
    /// <param name="metricNamespace">The namespace for metrics.</param>
    /// <returns>The metric name.</returns>
    private static async Task<Metric> SelectMetric(string metricNamespace)
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine($"2. Select a CloudWatch metric from a namespace.");

        var namespaceMetrics = await
_cloudWatchWrapper.ListMetrics(metricNamespace);

        for (int i = 0; i < namespaceMetrics.Count && i < 15; i++)
        {
            var dimensionsWithValues = namespaceMetrics[i].Dimensions
                .Where(d => !string.Equals("None", d.Value));
            Console.WriteLine($"\t{i + 1}. {namespaceMetrics[i].MetricName} " +
                              $"{string.Join(", :", dimensionsWithValues.Select(d
=> d.Value))}");
        }

        var metricChoiceNumber = 0;
        while (metricChoiceNumber < 1 || metricChoiceNumber >
namespaceMetrics.Count)
        {
            Console.WriteLine(
                "Select a metric by entering a number from the preceding list:");
            var choice = Console.ReadLine();
            Int32.TryParse(choice, out metricChoiceNumber);
        }

        var selectedMetric = namespaceMetrics[metricChoiceNumber - 1];

        Console.WriteLine(new string('-', 80));

        return selectedMetric;
    }

    /// <summary>
    /// Get and display metric statistics for a specific metric.
    /// </summary>
    /// <param name="metricNamespace">The namespace for metrics.</param>
    /// <param name="metric">The CloudWatch metric.</param>
    /// <returns>Async task.</returns>
    private static async Task GetAndDisplayMetricStatistics(string
metricNamespace, Metric metric)
```

```
   {
       Console.WriteLine(new string('-', 80));
       Console.WriteLine($"3. Get CloudWatch metric statistics for the last
day.");

       for (int i = 0; i < _statTypes.Count; i++)
       {
           Console.WriteLine($"\t{i + 1}. {_statTypes[i]}");
       }

       var statisticChoiceNumber = 0;
       while (statisticChoiceNumber < 1 || statisticChoiceNumber >
_statTypes.Count)
       {
           Console.WriteLine(
               "Select a metric statistic by entering a number from the
preceding list:");
           var choice = Console.ReadLine();
           Int32.TryParse(choice, out statisticChoiceNumber);
       }

       var selectedStatistic = _statTypes[statisticChoiceNumber - 1];
       var statisticsList = new List<string> { selectedStatistic };

       var metricStatistics = await
_cloudWatchWrapper.GetMetricStatistics(metricNamespace, metric.MetricName,
statisticsList, metric.Dimensions, 1, 60);

       if (!metricStatistics.Any())
       {
           Console.WriteLine($"No {selectedStatistic} statistics found for
{metric} in namespace {metricNamespace}.");
       }

       metricStatistics = metricStatistics.OrderBy(s => s.Timestamp).ToList();
       for (int i = 0; i < metricStatistics.Count && i < 10; i++)
       {
           var metricStat = metricStatistics[i];
           var statValue =
metricStat.GetType().GetProperty(selectedStatistic)!.GetValue(metricStat, null);
           Console.WriteLine($"\t{i + 1}. Timestamp
{metricStatistics[i].Timestamp:G} {selectedStatistic}: {statValue}");
       }
```

```
        Console.WriteLine(new string('-', 80));
    }

    /// <summary>
    /// Get and display estimated billing statistics.
    /// </summary>
    /// <param name="metricNamespace">The namespace for metrics.</param>
    /// <param name="metric">The CloudWatch metric.</param>
    /// <returns>Async task.</returns>
    private static async Task GetAndDisplayEstimatedBilling()
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine($"4. Get CloudWatch estimated billing for the last
week.");

        var billingStatistics = await SetupBillingStatistics();

        for (int i = 0; i < billingStatistics.Count; i++)
        {
            Console.WriteLine($"\t{i + 1}. Timestamp
{billingStatistics[i].Timestamp:G} : {billingStatistics[i].Maximum}");
        }

        Console.WriteLine(new string('-', 80));
    }

    /// <summary>
    /// Get billing statistics using a call to a wrapper class.
    /// </summary>
    /// <returns>A collection of billing statistics.</returns>
    private static async Task<List<Datapoint>> SetupBillingStatistics()
    {
        // Make a request for EstimatedCharges with a period of one day for the
past seven days.
        var billingStatistics = await _cloudWatchWrapper.GetMetricStatistics(
            "AWS/Billing",
            "EstimatedCharges",
            new List<string>() { "Maximum" },
            new List<Dimension>() { new Dimension { Name = "Currency", Value =
"USD" } },
            7,
            86400);

        billingStatistics = billingStatistics.OrderBy(n => n.Timestamp).ToList();
```

```
        return billingStatistics;
    }

    /// <summary>
    /// Create a dashboard with metrics.
    /// </summary>
    /// <param name="metricNamespace">The namespace for metrics.</param>
    /// <param name="metric">The CloudWatch metric.</param>
    /// <returns>Async task.</returns>
    private static async Task CreateDashboardWithMetrics()
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine($"5. Create a new CloudWatch dashboard with metrics.");
        var dashboardName = _configuration["dashboardName"];
        var newDashboard = new DashboardModel();
        _configuration.GetSection("dashboardExampleBody").Bind(newDashboard);
        var newDashboardString = JsonSerializer.Serialize(
            newDashboard,
            new JsonSerializerOptions
            {
                DefaultIgnoreCondition = JsonIgnoreCondition.WhenWritingNull
            });
        var validationMessages =
            await _cloudWatchWrapper.PutDashboard(dashboardName,
newDashboardString);

        Console.WriteLine(validationMessages.Any() ? $"\tValidation messages:" :
null);
        for (int i = 0; i < validationMessages.Count; i++)
        {
            Console.WriteLine($"\t{i + 1}. {validationMessages[i].Message}");
        }
        Console.WriteLine($"\tDashboard {dashboardName} was created.");
        Console.WriteLine(new string('-', 80));
    }

    /// <summary>
    /// List dashboards.
    /// </summary>
    /// <returns>Async task.</returns>
    private static async Task ListDashboards()
    {
        Console.WriteLine(new string('-', 80));
```

```
        Console.WriteLine($"6. List the CloudWatch dashboards in the current
account.");

        var dashboards = await _cloudWatchWrapper.ListDashboards();

        for (int i = 0; i < dashboards.Count; i++)
        {
            Console.WriteLine($"\t{i + 1}. {dashboards[i].DashboardName}");
        }

        Console.WriteLine(new string('-', 80));
    }

    /// <summary>
    /// Create and add data for a new custom metric.
    /// </summary>
    /// <returns>Async task.</returns>
    private static async Task CreateNewCustomMetric()
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine($"7. Create and add data for a new custom metric.");

        var customMetricNamespace = _configuration["customMetricNamespace"];
        var customMetricName = _configuration["customMetricName"];

        var customData = await PutRandomMetricData(customMetricName,
customMetricNamespace);

        var valuesString = string.Join(',', customData.Select(d => d.Value));
        Console.WriteLine($"\tAdded metric values for for metric
{customMetricName}: \n\t{valuesString}");

        Console.WriteLine(new string('-', 80));
    }


    /// <summary>
    /// Add some metric data using a call to a wrapper class.
    /// </summary>
    /// <param name="customMetricName">The metric name.</param>
    /// <param name="customMetricNamespace">The metric namespace.</param>
    /// <returns></returns>
    private static async Task<List<MetricDatum>> PutRandomMetricData(string
customMetricName,
```

```
        string customMetricNamespace)
    {
        List<MetricDatum> customData = new List<MetricDatum>();
        Random rnd = new Random();

        // Add 10 random values up to 100, starting with a timestamp 15 minutes
in the past.
        var utcNowMinus15 = DateTime.UtcNow.AddMinutes(-15);
        for (int i = 0; i < 10; i++)
        {
            var metricValue = rnd.Next(0, 100);
            customData.Add(
                new MetricDatum
                {
                    MetricName = customMetricName,
                    Value = metricValue,
                    TimestampUtc = utcNowMinus15.AddMinutes(i)
                }
            );
        }

        await _cloudWatchWrapper.PutMetricData(customMetricNamespace,
customData);
        return customData;
    }

    /// <summary>
    /// Add the custom metric to the dashboard.
    /// </summary>
    /// <returns>Async task.</returns>
    private static async Task AddMetricToDashboard()
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine($"8. Add the new custom metric to the dashboard.");

        var dashboardName = _configuration["dashboardName"];

        var customMetricNamespace = _configuration["customMetricNamespace"];
        var customMetricName = _configuration["customMetricName"];

        var validationMessages = await SetupDashboard(customMetricNamespace,
customMetricName, dashboardName);
```

```
        Console.WriteLine(validationMessages.Any() ? $"\tValidation messages:" :
null);
        for (int i = 0; i < validationMessages.Count; i++)
        {
            Console.WriteLine($"\t{i + 1}. {validationMessages[i].Message}");
        }
        Console.WriteLine($"\tDashboard {dashboardName} updated with metric
{customMetricName}.");
        Console.WriteLine(new string('-', 80));
    }


    /// <summary>
    /// Set up a dashboard using a call to the wrapper class.
    /// </summary>
    /// <param name="customMetricNamespace">The metric namespace.</param>
    /// <param name="customMetricName">The metric name.</param>
    /// <param name="dashboardName">The name of the dashboard.</param>
    /// <returns>A list of validation messages.</returns>
    private static async Task<List<DashboardValidationMessage>> SetupDashboard(
        string customMetricNamespace, string customMetricName, string
dashboardName)
    {
        // Get the dashboard model from configuration.
        var newDashboard = new DashboardModel();
        _configuration.GetSection("dashboardExampleBody").Bind(newDashboard);

        // Add a new metric to the dashboard.
        newDashboard.Widgets.Add(new Widget
        {
            Height = 8,
            Width = 8,
            Y = 8,
            X = 0,
            Type = "metric",
            Properties = new Properties
            {
                Metrics = new List<List<object>>
                    { new() { customMetricNamespace, customMetricName } },
                View = "timeSeries",
                Region = "us-east-1",
                Stat = "Sum",
                Period = 86400,
                YAxis = new YAxis { Left = new Left { Min = 0, Max = 100 } },
```

```
                Title = "Custom Metric Widget",
                LiveData = true,
                Sparkline = true,
                Trend = true,
                Stacked = false,
                SetPeriodToTimeRange = false
            }
        });

        var newDashboardString = JsonSerializer.Serialize(newDashboard,
            new JsonSerializerOptions
            { DefaultIgnoreCondition = JsonIgnoreCondition.WhenWritingNull });
        var validationMessages =
            await _cloudWatchWrapper.PutDashboard(dashboardName,
newDashboardString);

        return validationMessages;
    }

    /// <summary>
    /// Create a CloudWatch alarm for the new metric.
    /// </summary>
    /// <returns>Async task.</returns>
    private static async Task CreateMetricAlarm()
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine($"9. Create a CloudWatch alarm for the new metric.");

        var customMetricNamespace = _configuration["customMetricNamespace"];
        var customMetricName = _configuration["customMetricName"];

        var alarmName = _configuration["exampleAlarmName"];
        var accountId = _configuration["accountId"];
        var region = _configuration["region"];
        var emailTopic = _configuration["emailTopic"];
        var alarmActions = new List<string>();

        if (GetYesNoResponse(
                $"\tAdd an email action for topic {emailTopic} to alarm
{alarmName}? (y/n)"))
        {
            _cloudWatchWrapper.AddEmailAlarmAction(accountId, region, emailTopic,
alarmActions);
        }
```

```
        await _cloudWatchWrapper.PutMetricEmailAlarm(
            "Example metric alarm",
            alarmName,
            ComparisonOperator.GreaterThanOrEqualToThreshold,
            customMetricName,
            customMetricNamespace,
            100,
            alarmActions);

        Console.WriteLine($"\tAlarm {alarmName} added for metric
{customMetricName}.");
        Console.WriteLine(new string('-', 80));
    }

    /// <summary>
    /// Describe Alarms.
    /// </summary>
    /// <returns>Async task.</returns>
    private static async Task DescribeAlarms()
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine($"10. Describe CloudWatch alarms in the current
account.");

        var alarms = await _cloudWatchWrapper.DescribeAlarms();
        alarms = alarms.OrderByDescending(a => a.StateUpdatedTimestamp).ToList();

        for (int i = 0; i < alarms.Count && i < 10; i++)
        {
            var alarm = alarms[i];
            Console.WriteLine($"\t{i + 1}. {alarm.AlarmName}");
            Console.WriteLine($"\tState: {alarm.StateValue} for
{alarm.MetricName} {alarm.ComparisonOperator} {alarm.Threshold}");
        }

        Console.WriteLine(new string('-', 80));
    }

    /// <summary>
    /// Get the recent data for the metric.
    /// </summary>
    /// <returns>Async task.</returns>
    private static async Task GetCustomMetricData()
```

```
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine($"11. Get current data for new custom metric.");

        var customMetricNamespace = _configuration["customMetricNamespace"];
        var customMetricName = _configuration["customMetricName"];
        var accountId = _configuration["accountId"];

        var query = new List<MetricDataQuery>
        {
            new MetricDataQuery
            {
                AccountId = accountId,
                Id = "m1",
                Label = "Custom Metric Data",
                MetricStat = new MetricStat
                {
                    Metric = new Metric
                    {
                        MetricName = customMetricName,
                        Namespace = customMetricNamespace,
                    },
                    Period = 1,
                    Stat = "Maximum"
                }
            }
        };

        var metricData = await _cloudWatchWrapper.GetMetricData(
            20,
            true,
            DateTime.UtcNow.AddMinutes(1),
            20,
            query);

        for (int i = 0; i < metricData.Count; i++)
        {
            for (int j = 0; j < metricData[i].Values.Count; j++)
            {
                Console.WriteLine(
                    $"\tTimestamp {metricData[i].Timestamps[j]:G} Value:
{metricData[i].Values[j]}");
            }
        }
```

```
        Console.WriteLine(new string('-', 80));
    }

    /// <summary>
    /// Add metric data to trigger an alarm.
    /// </summary>
    /// <returns>Async task.</returns>
    private static async Task AddMetricDataForAlarm()
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine($"12. Add metric data to the custom metric to trigger
an alarm.");

        var customMetricNamespace = _configuration["customMetricNamespace"];
        var customMetricName = _configuration["customMetricName"];
        var nowUtc = DateTime.UtcNow;
        List<MetricDatum> customData = new List<MetricDatum>
        {
            new MetricDatum
            {
                MetricName = customMetricName,
                Value = 101,
                TimestampUtc = nowUtc.AddMinutes(-2)
            },
            new MetricDatum
            {
                MetricName = customMetricName,
                Value = 101,
                TimestampUtc = nowUtc.AddMinutes(-1)
            },
            new MetricDatum
            {
                MetricName = customMetricName,
                Value = 101,
                TimestampUtc = nowUtc
            }
        };
        var valuesString = string.Join(',', customData.Select(d => d.Value));
        Console.WriteLine($"\tAdded metric values for for metric
{customMetricName}: \n\t{valuesString}");
        await _cloudWatchWrapper.PutMetricData(customMetricNamespace,
customData);
```

```
        Console.WriteLine(new string('-', 80));
    }

    /// <summary>
    /// Check for a metric alarm using the DescribeAlarmsForMetric action.
    /// </summary>
    /// <returns>Async task.</returns>
    private static async Task CheckForMetricAlarm()
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine($"13. Checking for an alarm state.");

        var customMetricNamespace = _configuration["customMetricNamespace"];
        var customMetricName = _configuration["customMetricName"];
        var hasAlarm = false;
        var retries = 10;
        while (!hasAlarm && retries > 0)
        {
            var alarms = await
_cloudWatchWrapper.DescribeAlarmsForMetric(customMetricNamespace,
customMetricName);
            hasAlarm = alarms.Any(a => a.StateValue == StateValue.ALARM);
            retries--;
            Thread.Sleep(20000);
        }

        Console.WriteLine(hasAlarm
            ? $"\tAlarm state found for {customMetricName}."
            : $"\tNo Alarm state found for {customMetricName} after 10
retries.");

        Console.WriteLine(new string('-', 80));
    }

    /// <summary>
    /// Get history for an alarm.
    /// </summary>
    /// <returns>Async task.</returns>
    private static async Task GetAlarmHistory()
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine($"14. Get alarm history.");

        var exampleAlarmName = _configuration["exampleAlarmName"];
```

```
        var alarmHistory = await
_cloudWatchWrapper.DescribeAlarmHistory(exampleAlarmName, 2);

        for (int i = 0; i < alarmHistory.Count; i++)
        {
            var history = alarmHistory[i];
            Console.WriteLine($"\t{i + 1}. {history.HistorySummary}, time
{history.Timestamp:g}");
        }
        if (!alarmHistory.Any())
        {
            Console.WriteLine($"\tNo alarm history data found for
{exampleAlarmName}.");
        }

        Console.WriteLine(new string('-', 80));
    }

    /// <summary>
    /// Add an anomaly detector.
    /// </summary>
    /// <returns>Async task.</returns>
    private static async Task<SingleMetricAnomalyDetector> AddAnomalyDetector()
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine($"15. Add an anomaly detector.");

        var customMetricNamespace = _configuration["customMetricNamespace"];
        var customMetricName = _configuration["customMetricName"];

        var detector = new SingleMetricAnomalyDetector
        {
            MetricName = customMetricName,
            Namespace = customMetricNamespace,
            Stat = "Maximum"
        };
        await _cloudWatchWrapper.PutAnomalyDetector(detector);
        Console.WriteLine($"\tAdded anomaly detector for metric
{customMetricName}.");

        Console.WriteLine(new string('-', 80));
        return detector;
    }
```

```
    /// <summary>
    /// Describe anomaly detectors.
    /// </summary>
    /// <returns>Async task.</returns>
    private static async Task DescribeAnomalyDetectors()
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine($"16. Describe anomaly detectors in the current
account.");

        var customMetricNamespace = _configuration["customMetricNamespace"];
        var customMetricName = _configuration["customMetricName"];

        var detectors = await
_cloudWatchWrapper.DescribeAnomalyDetectors(customMetricNamespace,
customMetricName);

        for (int i = 0; i < detectors.Count; i++)
        {
            var detector = detectors[i];
            Console.WriteLine($"\t{i + 1}.
{detector.SingleMetricAnomalyDetector.MetricName}, state
{detector.StateValue}");
        }

        Console.WriteLine(new string('-', 80));
    }

    /// <summary>
    /// Fetch and open a metrics image for a CloudWatch metric and namespace.
    /// </summary>
    /// <returns>Async task.</returns>
    private static async Task GetAndOpenMetricImage()
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine("17. Get a metric image from CloudWatch.");

        Console.WriteLine($"\tGetting Image data for custom metric.");
        var customMetricNamespace = _configuration["customMetricNamespace"];
        var customMetricName = _configuration["customMetricName"];
```

```
        var memoryStream = await
_cloudWatchWrapper.GetTimeSeriesMetricImage(customMetricNamespace,
customMetricName, "Maximum", 10);
        var file = _cloudWatchWrapper.SaveMetricImage(memoryStream,
"MetricImages");

        ProcessStartInfo info = new ProcessStartInfo();

        Console.WriteLine($"\tFile saved as {Path.GetFileName(file)}.");
        Console.WriteLine($"\tPress enter to open the image.");
        Console.ReadLine();
        info.FileName = Path.Combine("ms-photos://", file);
        info.UseShellExecute = true;
        info.CreateNoWindow = true;
        info.Verb = string.Empty;

        Process.Start(info);

        Console.WriteLine(new string('-', 80));
    }

    /// <summary>
    /// Clean up created resources.
    /// </summary>
    /// <param name="metricNamespace">The namespace for metrics.</param>
    /// <param name="metric">The CloudWatch metric.</param>
    /// <returns>Async task.</returns>
    private static async Task CleanupResources()
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine($"18. Clean up resources.");

        var dashboardName = _configuration["dashboardName"];
        if (GetYesNoResponse($"\tDelete dashboard {dashboardName}? (y/n)"))
        {
            Console.WriteLine($"\tDeleting dashboard.");
            var dashboardList = new List<string> { dashboardName };
            await _cloudWatchWrapper.DeleteDashboards(dashboardList);
        }

        var alarmName = _configuration["exampleAlarmName"];
        if (GetYesNoResponse($"\tDelete alarm {alarmName}? (y/n)"))
        {
            Console.WriteLine($"\tCleaning up alarms.");
```

```
            var alarms = new List<string> { alarmName };
            await _cloudWatchWrapper.DeleteAlarms(alarms);
        }

        if (GetYesNoResponse($"\tDelete anomaly detector? (y/n)") &&
 anomalyDetector != null)
        {
            Console.WriteLine($"\tCleaning up anomaly detector.");

            await _cloudWatchWrapper.DeleteAnomalyDetector(
                anomalyDetector);
        }

        Console.WriteLine(new string('-', 80));
    }

    /// <summary>
    /// Get a yes or no response from the user.
    /// </summary>
    /// <param name="question">The question string to print on the console.</
param>
    /// <returns>True if the user responds with a yes.</returns>
    private static bool GetYesNoResponse(string question)
    {
        Console.WriteLine(question);
        var ynResponse = Console.ReadLine();
        var response = ynResponse != null &&
                       ynResponse.Equals("y",
                           StringComparison.InvariantCultureIgnoreCase);
        return response;
    }
}
```

Métodos envolventes utilizados por el escenario para las acciones de CloudWatch.

```
/// <summary>
/// Wrapper class for Amazon CloudWatch methods.
/// </summary>
public class CloudWatchWrapper
{
    private readonly IAmazonCloudWatch _amazonCloudWatch;
    private readonly ILogger<CloudWatchWrapper> _logger;
```

```
    /// <summary>
    /// Constructor for the CloudWatch wrapper.
    /// </summary>
    /// <param name="amazonCloudWatch">The injected CloudWatch client.</param>
    /// <param name="logger">The injected logger for the wrapper.</param>
    public CloudWatchWrapper(IAmazonCloudWatch amazonCloudWatch,
ILogger<CloudWatchWrapper> logger)

    {
        _logger = logger;
        _amazonCloudWatch = amazonCloudWatch;
    }

    /// <summary>
    /// List metrics available, optionally within a namespace.
    /// </summary>
    /// <param name="metricNamespace">Optional CloudWatch namespace to use when
listing metrics.</param>
    /// <param name="filter">Optional dimension filter.</param>
    /// <param name="metricName">Optional metric name filter.</param>
    /// <returns>The list of metrics.</returns>
    public async Task<List<Metric>> ListMetrics(string? metricNamespace = null,
DimensionFilter? filter = null, string? metricName = null)
    {
        var results = new List<Metric>();
        var paginateMetrics = _amazonCloudWatch.Paginators.ListMetrics(
            new ListMetricsRequest
            {
                Namespace = metricNamespace,
                Dimensions = filter != null ? new List<DimensionFilter>
{ filter } : null,
                MetricName = metricName
            });
        // Get the entire list using the paginator.
        await foreach (var metric in paginateMetrics.Metrics)
        {
            results.Add(metric);
        }

        return results;
    }

    /// <summary>
```

```
    /// Wrapper to get statistics for a specific CloudWatch metric.
    /// </summary>
    /// <param name="metricNamespace">The namespace of the metric.</param>
    /// <param name="metricName">The name of the metric.</param>
    /// <param name="statistics">The list of statistics to include.</param>
    /// <param name="dimensions">The list of dimensions to include.</param>
    /// <param name="days">The number of days in the past to include.</param>
    /// <param name="period">The period for the data.</param>
    /// <returns>A list of DataPoint objects for the statistics.</returns>
    public async Task<List<Datapoint>> GetMetricStatistics(string
 metricNamespace,
        string metricName, List<string> statistics, List<Dimension> dimensions,
 int days, int period)
    {
        var metricStatistics = await _amazonCloudWatch.GetMetricStatisticsAsync(
            new GetMetricStatisticsRequest()
            {
                Namespace = metricNamespace,
                MetricName = metricName,
                Dimensions = dimensions,
                Statistics = statistics,
                StartTimeUtc = DateTime.UtcNow.AddDays(-days),
                EndTimeUtc = DateTime.UtcNow,
                Period = period
            });

        return metricStatistics.Datapoints;
    }

    /// <summary>
    /// Wrapper to create or add to a dashboard with metrics.
    /// </summary>
    /// <param name="dashboardName">The name for the dashboard.</param>
    /// <param name="dashboardBody">The metric data in JSON for the dashboard.</
param>
    /// <returns>A list of validation messages for the dashboard.</returns>
    public async Task<List<DashboardValidationMessage>> PutDashboard(string
 dashboardName,
        string dashboardBody)
    {
        // Updating a dashboard replaces all contents.
        // Best practice is to include a text widget indicating this dashboard
 was created programmatically.
        var dashboardResponse = await _amazonCloudWatch.PutDashboardAsync(
```

```
            new PutDashboardRequest()
            {
                DashboardName = dashboardName,
                DashboardBody = dashboardBody
            });

        return dashboardResponse.DashboardValidationMessages;
    }


    /// <summary>
    /// Get information on a dashboard.
    /// </summary>
    /// <param name="dashboardName">The name of the dashboard.</param>
    /// <returns>A JSON object with dashboard information.</returns>
    public async Task<string> GetDashboard(string dashboardName)
    {
        var dashboardResponse = await _amazonCloudWatch.GetDashboardAsync(
            new GetDashboardRequest()
            {
                DashboardName = dashboardName
            });

        return dashboardResponse.DashboardBody;
    }


    /// <summary>
    /// Get a list of dashboards.
    /// </summary>
    /// <returns>A list of DashboardEntry objects.</returns>
    public async Task<List<DashboardEntry>> ListDashboards()
    {
        var results = new List<DashboardEntry>();
        var paginateDashboards = _amazonCloudWatch.Paginators.ListDashboards(
            new ListDashboardsRequest());
        // Get the entire list using the paginator.
        await foreach (var data in paginateDashboards.DashboardEntries)
        {
            results.Add(data);
        }

        return results;
    }
```

```
    /// <summary>
    /// Wrapper to add metric data to a CloudWatch metric.
    /// </summary>
    /// <param name="metricNamespace">The namespace of the metric.</param>
    /// <param name="metricData">A data object for the metric data.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> PutMetricData(string metricNamespace,
        List<MetricDatum> metricData)
    {
        var putDataResponse = await _amazonCloudWatch.PutMetricDataAsync(
            new PutMetricDataRequest()
            {
                MetricData = metricData,
                Namespace = metricNamespace,
            });

        return putDataResponse.HttpStatusCode == HttpStatusCode.OK;
    }

    /// <summary>
    /// Get an image for a metric graphed over time.
    /// </summary>
    /// <param name="metricNamespace">The namespace of the metric.</param>
    /// <param name="metric">The name of the metric.</param>
    /// <param name="stat">The name of the stat to chart.</param>
    /// <param name="period">The period to use for the chart.</param>
    /// <returns>A memory stream for the chart image.</returns>
    public async Task<MemoryStream> GetTimeSeriesMetricImage(string
metricNamespace, string metric, string stat, int period)
    {
        var metricImageWidget = new
        {
            title = "Example Metric Graph",
            view = "timeSeries",
            stacked = false,
            period = period,
            width = 1400,
            height = 600,
            metrics = new List<List<object>>
                { new() { metricNamespace, metric, new { stat } } }
        };
```

```
        var metricImageWidgetString =
 JsonSerializer.Serialize(metricImageWidget);
        var imageResponse = await _amazonCloudWatch.GetMetricWidgetImageAsync(
            new GetMetricWidgetImageRequest()
            {
                MetricWidget = metricImageWidgetString
            });

        return imageResponse.MetricWidgetImage;
    }

    /// <summary>
    /// Save a metric image to a file.
    /// </summary>
    /// <param name="memoryStream">The MemoryStream for the metric image.</param>
    /// <param name="metricName">The name of the metric.</param>
    /// <returns>The path to the file.</returns>
    public string SaveMetricImage(MemoryStream memoryStream, string metricName)
    {
        var metricFileName = $"{metricName}_{DateTime.Now.Ticks}.png";
        using var sr = new StreamReader(memoryStream);
        // Writes the memory stream to a file.
        File.WriteAllBytes(metricFileName, memoryStream.ToArray());
        var filePath = Path.Join(AppDomain.CurrentDomain.BaseDirectory,
            metricFileName);
        return filePath;
    }

    /// <summary>
    /// Get data for CloudWatch metrics.
    /// </summary>
    /// <param name="minutesOfData">The number of minutes of data to include.</
param>
    /// <param name="useDescendingTime">True to return the data descending by
 time.</param>
    /// <param name="endDateUtc">The end date for the data, in UTC.</param>
    /// <param name="maxDataPoints">The maximum data points to include.</param>
    /// <param name="dataQueries">Optional data queries to include.</param>
    /// <returns>A list of the requested metric data.</returns>
    public async Task<List<MetricDataResult>> GetMetricData(int minutesOfData,
 bool useDescendingTime, DateTime? endDateUtc = null,
        int maxDataPoints = 0, List<MetricDataQuery>? dataQueries = null)
    {
        var metricData = new List<MetricDataResult>();
```

```
        // If no end time is provided, use the current time for the end time.
        endDateUtc ??= DateTime.UtcNow;
        var timeZoneOffset =
 TimeZoneInfo.Local.GetUtcOffset(endDateUtc.Value.ToLocalTime());
        var startTimeUtc = endDateUtc.Value.AddMinutes(-minutesOfData);
        // The timezone string should be in the format +0000, so use the timezone
 offset to format it correctly.
        var timeZoneString = $"{timeZoneOffset.Hours:D2}
{timeZoneOffset.Minutes:D2}";
        var paginatedMetricData = _amazonCloudWatch.Paginators.GetMetricData(
            new GetMetricDataRequest()
            {
                StartTimeUtc = startTimeUtc,
                EndTimeUtc = endDateUtc.Value,
                LabelOptions = new LabelOptions { Timezone = timeZoneString },
                ScanBy = useDescendingTime ? ScanBy.TimestampDescending :
 ScanBy.TimestampAscending,
                MaxDatapoints = maxDataPoints,
                MetricDataQueries = dataQueries,
            });

        await foreach (var data in paginatedMetricData.MetricDataResults)
        {
            metricData.Add(data);
        }
        return metricData;
    }

    /// <summary>
    /// Add a metric alarm to send an email when the metric passes a threshold.
    /// </summary>
    /// <param name="alarmDescription">A description of the alarm.</param>
    /// <param name="alarmName">The name for the alarm.</param>
    /// <param name="comparison">The type of comparison to use.</param>
    /// <param name="metricName">The name of the metric for the alarm.</param>
    /// <param name="metricNamespace">The namespace of the metric.</param>
    /// <param name="threshold">The threshold value for the alarm.</param>
    /// <param name="alarmActions">Optional actions to execute when in an alarm
 state.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> PutMetricEmailAlarm(string alarmDescription, string
 alarmName, ComparisonOperator comparison,
        string metricName, string metricNamespace, double threshold, List<string>
 alarmActions = null!)
```

```
    {
        try
        {
            var putEmailAlarmResponse = await
_amazonCloudWatch.PutMetricAlarmAsync(
                new PutMetricAlarmRequest()
                {
                    AlarmActions = alarmActions,
                    AlarmDescription = alarmDescription,
                    AlarmName = alarmName,
                    ComparisonOperator = comparison,
                    Threshold = threshold,
                    Namespace = metricNamespace,
                    MetricName = metricName,
                    EvaluationPeriods = 1,
                    Period = 10,
                    Statistic = new Statistic("Maximum"),
                    DatapointsToAlarm = 1,
                    TreatMissingData = "ignore"
                });
            return putEmailAlarmResponse.HttpStatusCode == HttpStatusCode.OK;
        }
        catch (LimitExceededException lex)
        {
            _logger.LogError(lex, $"Unable to add alarm {alarmName}. Alarm quota
has already been reached.");
        }

        return false;
    }

    /// <summary>
    /// Add specific email actions to a list of action strings for a CloudWatch
alarm.
    /// </summary>
    /// <param name="accountId">The AccountId for the alarm.</param>
    /// <param name="region">The region for the alarm.</param>
    /// <param name="emailTopicName">An Amazon Simple Notification Service (SNS)
topic for the alarm email.</param>
    /// <param name="alarmActions">Optional list of existing alarm actions to
append to.</param>
    /// <returns>A list of string actions for an alarm.</returns>
    public List<string> AddEmailAlarmAction(string accountId, string region,
        string emailTopicName, List<string>? alarmActions = null)
```

```
    {
        alarmActions ??= new List<string>();
        var snsAlarmAction = $"arn:aws:sns:{region}:{accountId}:
{emailTopicName}";
        alarmActions.Add(snsAlarmAction);
        return alarmActions;
    }

    /// <summary>
    /// Describe the current alarms, optionally filtered by state.
    /// </summary>
    /// <param name="stateValue">Optional filter for alarm state.</param>
    /// <returns>The list of alarm data.</returns>
    public async Task<List<MetricAlarm>> DescribeAlarms(StateValue? stateValue =
 null)
    {
        List<MetricAlarm> alarms = new List<MetricAlarm>();
        var paginatedDescribeAlarms =
 _amazonCloudWatch.Paginators.DescribeAlarms(
            new DescribeAlarmsRequest()
            {
                StateValue = stateValue
            });

        await foreach (var data in paginatedDescribeAlarms.MetricAlarms)
        {
            alarms.Add(data);
        }
        return alarms;
    }

    /// <summary>
    /// Describe the current alarms for a specific metric.
    /// </summary>
    /// <param name="metricNamespace">The namespace of the metric.</param>
    /// <param name="metricName">The name of the metric.</param>
    /// <returns>The list of alarm data.</returns>
    public async Task<List<MetricAlarm>> DescribeAlarmsForMetric(string
 metricNamespace, string metricName)
    {
        var alarmsResult = await _amazonCloudWatch.DescribeAlarmsForMetricAsync(
            new DescribeAlarmsForMetricRequest()
            {
                Namespace = metricNamespace,
```

```
                MetricName = metricName
            });

        return alarmsResult.MetricAlarms;
    }

    /// <summary>
    /// Describe the history of an alarm for a number of days in the past.
    /// </summary>
    /// <param name="alarmName">The name of the alarm.</param>
    /// <param name="historyDays">The number of days in the past.</param>
    /// <returns>The list of alarm history data.</returns>
    public async Task<List<AlarmHistoryItem>> DescribeAlarmHistory(string
alarmName, int historyDays)
    {
        List<AlarmHistoryItem> alarmHistory = new List<AlarmHistoryItem>();
        var paginatedAlarmHistory =
_amazonCloudWatch.Paginators.DescribeAlarmHistory(
            new DescribeAlarmHistoryRequest()
            {
                AlarmName = alarmName,
                EndDateUtc = DateTime.UtcNow,
                HistoryItemType = HistoryItemType.StateUpdate,
                StartDateUtc = DateTime.UtcNow.AddDays(-historyDays)
            });

        await foreach (var data in paginatedAlarmHistory.AlarmHistoryItems)
        {
            alarmHistory.Add(data);
        }
        return alarmHistory;
    }

    /// <summary>
    /// Delete a list of alarms from CloudWatch.
    /// </summary>
    /// <param name="alarmNames">A list of names of alarms to delete.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> DeleteAlarms(List<string> alarmNames)
    {
        var deleteAlarmsResult = await _amazonCloudWatch.DeleteAlarmsAsync(
            new DeleteAlarmsRequest()
            {
                AlarmNames = alarmNames
```

```
            });

        return deleteAlarmsResult.HttpStatusCode == HttpStatusCode.OK;
    }

    /// <summary>
    /// Disable the actions for a list of alarms from CloudWatch.
    /// </summary>
    /// <param name="alarmNames">A list of names of alarms.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> DisableAlarmActions(List<string> alarmNames)
    {
        var disableAlarmActionsResult = await
_amazonCloudWatch.DisableAlarmActionsAsync(
            new DisableAlarmActionsRequest()
            {
                AlarmNames = alarmNames
            });

        return disableAlarmActionsResult.HttpStatusCode == HttpStatusCode.OK;
    }

    /// <summary>
    /// Enable the actions for a list of alarms from CloudWatch.
    /// </summary>
    /// <param name="alarmNames">A list of names of alarms.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> EnableAlarmActions(List<string> alarmNames)
    {
        var enableAlarmActionsResult = await
_amazonCloudWatch.EnableAlarmActionsAsync(
            new EnableAlarmActionsRequest()
            {
                AlarmNames = alarmNames
            });

        return enableAlarmActionsResult.HttpStatusCode == HttpStatusCode.OK;
    }

    /// <summary>
    /// Add an anomaly detector for a single metric.
    /// </summary>
    /// <param name="anomalyDetector">A single metric anomaly detector.</param>
    /// <returns>True if successful.</returns>
```

```
   public async Task<bool> PutAnomalyDetector(SingleMetricAnomalyDetector
anomalyDetector)
   {
       var putAlarmDetectorResult = await
_amazonCloudWatch.PutAnomalyDetectorAsync(
           new PutAnomalyDetectorRequest()
           {
               SingleMetricAnomalyDetector = anomalyDetector
           });

       return putAlarmDetectorResult.HttpStatusCode == HttpStatusCode.OK;
   }

   /// <summary>
   /// Describe anomaly detectors for a metric and namespace.
   /// </summary>
   /// <param name="metricNamespace">The namespace of the metric.</param>
   /// <param name="metricName">The metric of the anomaly detectors.</param>
   /// <returns>The list of detectors.</returns>
   public async Task<List<AnomalyDetector>> DescribeAnomalyDetectors(string
metricNamespace, string metricName)
   {
       List<AnomalyDetector> detectors = new List<AnomalyDetector>();
       var paginatedDescribeAnomalyDetectors =
_amazonCloudWatch.Paginators.DescribeAnomalyDetectors(
           new DescribeAnomalyDetectorsRequest()
           {
               MetricName = metricName,
               Namespace = metricNamespace
           });

       await foreach (var data in
paginatedDescribeAnomalyDetectors.AnomalyDetectors)
       {
           detectors.Add(data);
       }

       return detectors;
   }

   /// <summary>
   /// Delete a single metric anomaly detector.
   /// </summary>
   /// <param name="anomalyDetector">The anomaly detector to delete.</param>
```

```
    /// <returns>True if successful.</returns>
    public async Task<bool> DeleteAnomalyDetector(SingleMetricAnomalyDetector
 anomalyDetector)
    {
        var deleteAnomalyDetectorResponse = await
 _amazonCloudWatch.DeleteAnomalyDetectorAsync(
            new DeleteAnomalyDetectorRequest()
            {
                SingleMetricAnomalyDetector = anomalyDetector
            });

        return deleteAnomalyDetectorResponse.HttpStatusCode == HttpStatusCode.OK;
    }

    /// <summary>
    /// Delete a list of CloudWatch dashboards.
    /// </summary>
    /// <param name="dashboardNames">List of dashboard names to delete.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> DeleteDashboards(List<string> dashboardNames)
    {
        var deleteDashboardsResponse = await
 _amazonCloudWatch.DeleteDashboardsAsync(
            new DeleteDashboardsRequest()
            {
                DashboardNames = dashboardNames
            });

        return deleteDashboardsResponse.HttpStatusCode == HttpStatusCode.OK;
    }
}
```

- Para obtener información sobre la API, consulte los siguientes temas en la referencia de la API de AWS SDK for .NET.
  - DeleteAlarms
  - DeleteAnomalyDetector
  - DeleteDashboards
  - DescribeAlarmHistory
  - DescribeAlarms
  - DescribeAlarmsForMetric

- DescribeAnomalyDetectors
- GetMetricData
- GetMetricStatistics
- GetMetricWidgetImage
- ListMetrics
- PutAnomalyDetector
- PutDashboard
- PutMetricAlarm
- PutMetricData

Java

SDK para Java 2.x

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
import com.fasterxml.jackson.core.JsonFactory;
import com.fasterxml.jackson.core.JsonParser;
import com.fasterxml.jackson.databind.ObjectMapper;
import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.core.SdkBytes;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.cloudwatch.CloudWatchClient;
import software.amazon.awssdk.services.cloudwatch.model.AlarmHistoryItem;
import software.amazon.awssdk.services.cloudwatch.model.AlarmType;
import software.amazon.awssdk.services.cloudwatch.model.AnomalyDetector;
import software.amazon.awssdk.services.cloudwatch.model.CloudWatchException;
import software.amazon.awssdk.services.cloudwatch.model.ComparisonOperator;
import
 software.amazon.awssdk.services.cloudwatch.model.DashboardValidationMessage;
import software.amazon.awssdk.services.cloudwatch.model.Datapoint;
import software.amazon.awssdk.services.cloudwatch.model.DeleteAlarmsRequest;
import
 software.amazon.awssdk.services.cloudwatch.model.DeleteAnomalyDetectorRequest;
```

```
import software.amazon.awssdk.services.cloudwatch.model.DeleteDashboardsRequest;
import
 software.amazon.awssdk.services.cloudwatch.model.DescribeAlarmHistoryRequest;
import
 software.amazon.awssdk.services.cloudwatch.model.DescribeAlarmHistoryResponse;
import
 software.amazon.awssdk.services.cloudwatch.model.DescribeAlarmsForMetricRequest;
import
 software.amazon.awssdk.services.cloudwatch.model.DescribeAlarmsForMetricResponse;
import software.amazon.awssdk.services.cloudwatch.model.DescribeAlarmsRequest;
import software.amazon.awssdk.services.cloudwatch.model.DescribeAlarmsResponse;
import
 software.amazon.awssdk.services.cloudwatch.model.DescribeAnomalyDetectorsRequest;
import
 software.amazon.awssdk.services.cloudwatch.model.DescribeAnomalyDetectorsResponse;
import software.amazon.awssdk.services.cloudwatch.model.Dimension;
import software.amazon.awssdk.services.cloudwatch.model.GetMetricDataRequest;
import software.amazon.awssdk.services.cloudwatch.model.GetMetricDataResponse;
import
 software.amazon.awssdk.services.cloudwatch.model.GetMetricStatisticsRequest;
import
 software.amazon.awssdk.services.cloudwatch.model.GetMetricStatisticsResponse;
import
 software.amazon.awssdk.services.cloudwatch.model.GetMetricWidgetImageRequest;
import
 software.amazon.awssdk.services.cloudwatch.model.GetMetricWidgetImageResponse;
import software.amazon.awssdk.services.cloudwatch.model.HistoryItemType;
import software.amazon.awssdk.services.cloudwatch.model.ListMetricsRequest;
import software.amazon.awssdk.services.cloudwatch.model.ListMetricsResponse;
import software.amazon.awssdk.services.cloudwatch.model.Metric;
import software.amazon.awssdk.services.cloudwatch.model.MetricAlarm;
import software.amazon.awssdk.services.cloudwatch.model.MetricDataQuery;
import software.amazon.awssdk.services.cloudwatch.model.MetricDataResult;
import software.amazon.awssdk.services.cloudwatch.model.MetricDatum;
import software.amazon.awssdk.services.cloudwatch.model.MetricStat;
import
 software.amazon.awssdk.services.cloudwatch.model.PutAnomalyDetectorRequest;
import software.amazon.awssdk.services.cloudwatch.model.PutDashboardRequest;
import software.amazon.awssdk.services.cloudwatch.model.PutDashboardResponse;
import software.amazon.awssdk.services.cloudwatch.model.PutMetricAlarmRequest;
import software.amazon.awssdk.services.cloudwatch.model.PutMetricDataRequest;
import software.amazon.awssdk.services.cloudwatch.model.ScanBy;
import
 software.amazon.awssdk.services.cloudwatch.model.SingleMetricAnomalyDetector;
```

```
import software.amazon.awssdk.services.cloudwatch.model.StandardUnit;
import software.amazon.awssdk.services.cloudwatch.model.Statistic;
import
 software.amazon.awssdk.services.cloudwatch.paginators.ListDashboardsIterable;
import software.amazon.awssdk.services.cloudwatch.paginators.ListMetricsIterable;
import java.io.BufferedReader;
import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStreamReader;
import java.nio.file.Files;
import java.nio.file.Paths;
import java.time.Instant;
import java.time.ZoneOffset;
import java.time.ZonedDateTime;
import java.time.format.DateTimeFormatter;
import java.time.temporal.ChronoUnit;
import java.util.ArrayList;
import java.util.List;
import java.util.Scanner;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 *
 * To enable billing metrics and statistics for this example, make sure billing
 * alerts are enabled for your account:
 * https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/
monitor_estimated_charges_with_cloudwatch.html#turning_on_billing_metrics
 *
 * This Java code example performs the following tasks:
 *
 * 1. List available namespaces from Amazon CloudWatch.
 * 2. List available metrics within the selected Namespace.
 * 3. Get statistics for the selected metric over the last day.
 * 4. Get CloudWatch estimated billing for the last week.
 * 5. Create a new CloudWatch dashboard with metrics.
 * 6. List dashboards using a paginator.
 * 7. Create a new custom metric by adding data for it.
```

```
 * 8. Add the custom metric to the dashboard.
 * 9. Create an alarm for the custom metric.
 * 10. Describe current alarms.
 * 11. Get current data for the new custom metric.
 * 12. Push data into the custom metric to trigger the alarm.
 * 13. Check the alarm state using the action DescribeAlarmsForMetric.
 * 14. Get alarm history for the new alarm.
 * 15. Add an anomaly detector for the custom metric.
 * 16. Describe current anomaly detectors.
 * 17. Get a metric image for the custom metric.
 * 18. Clean up the Amazon CloudWatch resources.
 */
public class CloudWatchScenario {
    public static final String DASHES = new String(new char[80]).replace("\0",
 "-");

    public static void main(String[] args) throws IOException {
        final String usage = """

                Usage:
                  <myDate> <costDateWeek> <dashboardName> <dashboardJson>
 <dashboardAdd> <settings> <metricImage> \s

                Where:
                  myDate - The start date to use to get metric statistics. (For
 example, 2023-01-11T18:35:24.00Z.)\s
                  costDateWeek - The start date to use to get AWS/Billinget
 statistics. (For example, 2023-01-11T18:35:24.00Z.)\s
                  dashboardName - The name of the dashboard to create.\s
                  dashboardJson - The location of a JSON file to use to create a
 dashboard. (See Readme file.)\s
                  dashboardAdd - The location of a JSON file to use to update a
 dashboard. (See Readme file.)\s
                  settings - The location of a JSON file from which various
 values are read. (See Readme file.)\s
                  metricImage - The location of a BMP file that is used to create
 a graph.\s
                """;

        if (args.length != 7) {
            System.out.println(usage);
            System.exit(1);
        }
```

```
        Region region = Region.US_EAST_1;
        String myDate = args[0];
        String costDateWeek = args[1];
        String dashboardName = args[2];
        String dashboardJson = args[3];
        String dashboardAdd = args[4];
        String settings = args[5];
        String metricImage = args[6];

        Double dataPoint = Double.parseDouble("10.0");
        Scanner sc = new Scanner(System.in);
        CloudWatchClient cw = CloudWatchClient.builder()
                .region(region)
                .credentialsProvider(ProfileCredentialsProvider.create())
                .build();

        System.out.println(DASHES);
        System.out.println("Welcome to the Amazon CloudWatch example scenario.");
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println(
                "1. List at least five available unique namespaces from Amazon
CloudWatch. Select one from the list.");
        ArrayList<String> list = listNameSpaces(cw);
        for (int z = 0; z < 5; z++) {
            int index = z + 1;
            System.out.println("    " + index + ". " + list.get(z));
        }

        String selectedNamespace = "";
        String selectedMetrics = "";
        int num = Integer.parseInt(sc.nextLine());
        if (1 <= num && num <= 5) {
            selectedNamespace = list.get(num - 1);
        } else {
            System.out.println("You did not select a valid option.");
            System.exit(1);
        }
        System.out.println("You selected " + selectedNamespace);
        System.out.println(DASHES);

        System.out.println(DASHES);
```

```
        System.out.println("2. List available metrics within the selected
namespace and select one from the list.");
        ArrayList<String> metList = listMets(cw, selectedNamespace);
        for (int z = 0; z < 5; z++) {
            int index = z + 1;
            System.out.println("    " + index + ". " + metList.get(z));
        }
        num = Integer.parseInt(sc.nextLine());
        if (1 <= num && num <= 5) {
            selectedMetrics = metList.get(num - 1);
        } else {
            System.out.println("You did not select a valid option.");
            System.exit(1);
        }
        System.out.println("You selected " + selectedMetrics);
        Dimension myDimension = getSpecificMet(cw, selectedNamespace);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("3. Get statistics for the selected metric over the
last day.");
        String metricOption = "";
        ArrayList<String> statTypes = new ArrayList<>();
        statTypes.add("SampleCount");
        statTypes.add("Average");
        statTypes.add("Sum");
        statTypes.add("Minimum");
        statTypes.add("Maximum");

        for (int t = 0; t < 5; t++) {
            System.out.println("    " + (t + 1) + ". " + statTypes.get(t));
        }
        System.out.println("Select a metric statistic by entering a number from
the preceding list:");
        num = Integer.parseInt(sc.nextLine());
        if (1 <= num && num <= 5) {
            metricOption = statTypes.get(num - 1);
        } else {
            System.out.println("You did not select a valid option.");
            System.exit(1);
        }
        System.out.println("You selected " + metricOption);
        getAndDisplayMetricStatistics(cw, selectedNamespace, selectedMetrics,
metricOption, myDate, myDimension);
```

```
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("4. Get CloudWatch estimated billing for the last
week.");
        getMetricStatistics(cw, costDateWeek);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("5. Create a new CloudWatch dashboard with metrics.");
        createDashboardWithMetrics(cw, dashboardName, dashboardJson);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("6. List dashboards using a paginator.");
        listDashboards(cw);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("7. Create a new custom metric by adding data to
it.");
        createNewCustomMetric(cw, dataPoint);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("8. Add an additional metric to the dashboard.");
        addMetricToDashboard(cw, dashboardAdd, dashboardName);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("9. Create an alarm for the custom metric.");
        String alarmName = createAlarm(cw, settings);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("10. Describe ten current alarms.");
        describeAlarms(cw);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("11. Get current data for new custom metric.");
        getCustomMetricData(cw, settings);
        System.out.println(DASHES);
```

```
        System.out.println(DASHES);
        System.out.println("12. Push data into the custom metric to trigger the
alarm.");
        addMetricDataForAlarm(cw, settings);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("13. Check the alarm state using the action
DescribeAlarmsForMetric.");
        checkForMetricAlarm(cw, settings);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("14. Get alarm history for the new alarm.");
        getAlarmHistory(cw, settings, myDate);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("15. Add an anomaly detector for the custom metric.");
        addAnomalyDetector(cw, settings);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("16. Describe current anomaly detectors.");
        describeAnomalyDetectors(cw, settings);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("17. Get a metric image for the custom metric.");
        getAndOpenMetricImage(cw, metricImage);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("18. Clean up the Amazon CloudWatch resources.");
        deleteDashboard(cw, dashboardName);
        deleteCWAlarm(cw, alarmName);
        deleteAnomalyDetector(cw, settings);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("The Amazon CloudWatch example scenario is
complete.");
        System.out.println(DASHES);
        cw.close();
```

```
   }

   public static void deleteAnomalyDetector(CloudWatchClient cw, String
fileName) {
       try {
           // Read values from the JSON file.
           JsonParser parser = new JsonFactory().createParser(new
File(fileName));
           com.fasterxml.jackson.databind.JsonNode rootNode = new
ObjectMapper().readTree(parser);
           String customMetricNamespace =
rootNode.findValue("customMetricNamespace").asText();
           String customMetricName =
rootNode.findValue("customMetricName").asText();

           SingleMetricAnomalyDetector singleMetricAnomalyDetector =
SingleMetricAnomalyDetector.builder()
                   .metricName(customMetricName)
                   .namespace(customMetricNamespace)
                   .stat("Maximum")
                   .build();

           DeleteAnomalyDetectorRequest request =
DeleteAnomalyDetectorRequest.builder()
                   .singleMetricAnomalyDetector(singleMetricAnomalyDetector)
                   .build();

           cw.deleteAnomalyDetector(request);
           System.out.println("Successfully deleted the Anomaly Detector.");

       } catch (CloudWatchException e) {
           System.err.println(e.awsErrorDetails().errorMessage());
           System.exit(1);
       } catch (IOException e) {
           e.printStackTrace();
       }
   }

   public static void deleteCWAlarm(CloudWatchClient cw, String alarmName) {
       try {
           DeleteAlarmsRequest request = DeleteAlarmsRequest.builder()
                   .alarmNames(alarmName)
                   .build();
```

```
            cw.deleteAlarms(request);
            System.out.println("Successfully deleted alarm " + alarmName);

        } catch (CloudWatchException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }

    public static void deleteDashboard(CloudWatchClient cw, String dashboardName)
{
        try {
            DeleteDashboardsRequest dashboardsRequest =
DeleteDashboardsRequest.builder()
                    .dashboardNames(dashboardName)
                    .build();
            cw.deleteDashboards(dashboardsRequest);
            System.out.println(dashboardName + " was successfully deleted.");

        } catch (CloudWatchException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }

    public static void getAndOpenMetricImage(CloudWatchClient cw, String
fileName) {
        System.out.println("Getting Image data for custom metric.");
        try {
            String myJSON = "{\n" +
                    "  \"title\": \"Example Metric Graph\",\n" +
                    "  \"view\": \"timeSeries\",\n" +
                    "  \"stacked \": false,\n" +
                    "  \"period\": 10,\n" +
                    "  \"width\": 1400,\n" +
                    "  \"height\": 600,\n" +
                    "  \"metrics\": [\n" +
                    "    [\n" +
                    "      \"AWS/Billing\",\n" +
                    "      \"EstimatedCharges\",\n" +
                    "      \"Currency\",\n" +
                    "      \"USD\"\n" +
                    "    ]\n" +
                    "  ]\n" +
```

```
                    "}";

            GetMetricWidgetImageRequest imageRequest =
GetMetricWidgetImageRequest.builder()
                    .metricWidget(myJSON)
                    .build();

            GetMetricWidgetImageResponse response =
cw.getMetricWidgetImage(imageRequest);
            SdkBytes sdkBytes = response.metricWidgetImage();
            byte[] bytes = sdkBytes.asByteArray();
            File outputFile = new File(fileName);
            try (FileOutputStream outputStream = new
FileOutputStream(outputFile)) {
                outputStream.write(bytes);
            }

        } catch (CloudWatchException | IOException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }

    public static void describeAnomalyDetectors(CloudWatchClient cw, String
fileName) {
        try {
            // Read values from the JSON file.
            JsonParser parser = new JsonFactory().createParser(new
File(fileName));
            com.fasterxml.jackson.databind.JsonNode rootNode = new
ObjectMapper().readTree(parser);
            String customMetricNamespace =
rootNode.findValue("customMetricNamespace").asText();
            String customMetricName =
rootNode.findValue("customMetricName").asText();
            DescribeAnomalyDetectorsRequest detectorsRequest =
DescribeAnomalyDetectorsRequest.builder()
                    .maxResults(10)
                    .metricName(customMetricName)
                    .namespace(customMetricNamespace)
                    .build();

            DescribeAnomalyDetectorsResponse response =
cw.describeAnomalyDetectors(detectorsRequest);
```

```
            List<AnomalyDetector> anomalyDetectorList =
response.anomalyDetectors();
            for (AnomalyDetector detector : anomalyDetectorList) {
                System.out.println("Metric name: " +
detector.singleMetricAnomalyDetector().metricName());
                System.out.println("State: " + detector.stateValue());
            }

        } catch (CloudWatchException | IOException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }

    public static void addAnomalyDetector(CloudWatchClient cw, String fileName) {
        try {
            // Read values from the JSON file.
            JsonParser parser = new JsonFactory().createParser(new
File(fileName));
            com.fasterxml.jackson.databind.JsonNode rootNode = new
ObjectMapper().readTree(parser);
            String customMetricNamespace =
rootNode.findValue("customMetricNamespace").asText();
            String customMetricName =
rootNode.findValue("customMetricName").asText();

            SingleMetricAnomalyDetector singleMetricAnomalyDetector =
SingleMetricAnomalyDetector.builder()
                    .metricName(customMetricName)
                    .namespace(customMetricNamespace)
                    .stat("Maximum")
                    .build();

            PutAnomalyDetectorRequest anomalyDetectorRequest =
PutAnomalyDetectorRequest.builder()
                    .singleMetricAnomalyDetector(singleMetricAnomalyDetector)
                    .build();

            cw.putAnomalyDetector(anomalyDetectorRequest);
            System.out.println("Added anomaly detector for metric " +
customMetricName + ".");

        } catch (CloudWatchException | IOException e) {
            System.err.println(e.getMessage());
```

```
            System.exit(1);
        }
    }

    public static void getAlarmHistory(CloudWatchClient cw, String fileName,
String date) {
        try {
            // Read values from the JSON file.
            JsonParser parser = new JsonFactory().createParser(new
File(fileName));
            com.fasterxml.jackson.databind.JsonNode rootNode = new
ObjectMapper().readTree(parser);
            String alarmName = rootNode.findValue("exampleAlarmName").asText();

            Instant start = Instant.parse(date);
            Instant endDate = Instant.now();
            DescribeAlarmHistoryRequest historyRequest =
DescribeAlarmHistoryRequest.builder()
                    .startDate(start)
                    .endDate(endDate)
                    .alarmName(alarmName)
                    .historyItemType(HistoryItemType.ACTION)
                    .build();

            DescribeAlarmHistoryResponse response =
cw.describeAlarmHistory(historyRequest);
            List<AlarmHistoryItem> historyItems = response.alarmHistoryItems();
            if (historyItems.isEmpty()) {
                System.out.println("No alarm history data found for " + alarmName
+ ".");
            } else {
                for (AlarmHistoryItem item : historyItems) {
                    System.out.println("History summary: " +
item.historySummary());
                    System.out.println("Time stamp: " + item.timestamp());
                }
            }

        } catch (CloudWatchException | IOException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }
```

```
   public static void checkForMetricAlarm(CloudWatchClient cw, String fileName)
{
       try {
           // Read values from the JSON file.
           JsonParser parser = new JsonFactory().createParser(new
File(fileName));
           com.fasterxml.jackson.databind.JsonNode rootNode = new
ObjectMapper().readTree(parser);
           String customMetricNamespace =
rootNode.findValue("customMetricNamespace").asText();
           String customMetricName =
rootNode.findValue("customMetricName").asText();
           boolean hasAlarm = false;
           int retries = 10;

           DescribeAlarmsForMetricRequest metricRequest =
DescribeAlarmsForMetricRequest.builder()
                   .metricName(customMetricName)
                   .namespace(customMetricNamespace)
                   .build();

           while (!hasAlarm && retries > 0) {
               DescribeAlarmsForMetricResponse response =
cw.describeAlarmsForMetric(metricRequest);
               hasAlarm = response.hasMetricAlarms();
               retries--;
               Thread.sleep(20000);
               System.out.println(".");
           }
           if (!hasAlarm)
               System.out.println("No Alarm state found for " + customMetricName
+ " after 10 retries.");
           else
               System.out.println("Alarm state found for " + customMetricName +
".");

       } catch (CloudWatchException | IOException | InterruptedException e) {
           System.err.println(e.getMessage());
           System.exit(1);
       }
   }

   public static void addMetricDataForAlarm(CloudWatchClient cw, String
fileName) {
```

```
        try {
            // Read values from the JSON file.
            JsonParser parser = new JsonFactory().createParser(new
File(fileName));
            com.fasterxml.jackson.databind.JsonNode rootNode = new
ObjectMapper().readTree(parser);
            String customMetricNamespace =
rootNode.findValue("customMetricNamespace").asText();
            String customMetricName =
rootNode.findValue("customMetricName").asText();

            // Set an Instant object.
            String time =
ZonedDateTime.now(ZoneOffset.UTC).format(DateTimeFormatter.ISO_INSTANT);
            Instant instant = Instant.parse(time);

            MetricDatum datum = MetricDatum.builder()
                    .metricName(customMetricName)
                    .unit(StandardUnit.NONE)
                    .value(1001.00)
                    .timestamp(instant)
                    .build();

            MetricDatum datum2 = MetricDatum.builder()
                    .metricName(customMetricName)
                    .unit(StandardUnit.NONE)
                    .value(1002.00)
                    .timestamp(instant)
                    .build();

            List<MetricDatum> metricDataList = new ArrayList<>();
            metricDataList.add(datum);
            metricDataList.add(datum2);

            PutMetricDataRequest request = PutMetricDataRequest.builder()
                    .namespace(customMetricNamespace)
                    .metricData(metricDataList)
                    .build();

            cw.putMetricData(request);
            System.out.println("Added metric values for for metric " +
customMetricName);

        } catch (CloudWatchException | IOException e) {
```

```
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }

    public static void getCustomMetricData(CloudWatchClient cw, String fileName)
{
        try {
            // Read values from the JSON file.
            JsonParser parser = new JsonFactory().createParser(new
File(fileName));
            com.fasterxml.jackson.databind.JsonNode rootNode = new
ObjectMapper().readTree(parser);
            String customMetricNamespace =
rootNode.findValue("customMetricNamespace").asText();
            String customMetricName =
rootNode.findValue("customMetricName").asText();

            // Set the date.
            Instant nowDate = Instant.now();

            long hours = 1;
            long minutes = 30;
            Instant date2 = nowDate.plus(hours, ChronoUnit.HOURS).plus(minutes,
                    ChronoUnit.MINUTES);

            Metric met = Metric.builder()
                    .metricName(customMetricName)
                    .namespace(customMetricNamespace)
                    .build();

            MetricStat metStat = MetricStat.builder()
                    .stat("Maximum")
                    .period(1)
                    .metric(met)
                    .build();

            MetricDataQuery dataQUery = MetricDataQuery.builder()
                    .metricStat(metStat)
                    .id("foo2")
                    .returnData(true)
                    .build();

            List<MetricDataQuery> dq = new ArrayList<>();
```

```
            dq.add(dataQUery);

            GetMetricDataRequest getMetReq = GetMetricDataRequest.builder()
                    .maxDatapoints(10)
                    .scanBy(ScanBy.TIMESTAMP_DESCENDING)
                    .startTime(nowDate)
                    .endTime(date2)
                    .metricDataQueries(dq)
                    .build();

            GetMetricDataResponse response = cw.getMetricData(getMetReq);
            List<MetricDataResult> data = response.metricDataResults();
            for (MetricDataResult item : data) {
                System.out.println("The label is " + item.label());
                System.out.println("The status code is " +
item.statusCode().toString());
            }

        } catch (CloudWatchException | IOException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }

    public static void describeAlarms(CloudWatchClient cw) {
        try {
            List<AlarmType> typeList = new ArrayList<>();
            typeList.add(AlarmType.METRIC_ALARM);

            DescribeAlarmsRequest alarmsRequest = DescribeAlarmsRequest.builder()
                    .alarmTypes(typeList)
                    .maxRecords(10)
                    .build();

            DescribeAlarmsResponse response = cw.describeAlarms(alarmsRequest);
            List<MetricAlarm> alarmList = response.metricAlarms();
            for (MetricAlarm alarm : alarmList) {
                System.out.println("Alarm name: " + alarm.alarmName());
                System.out.println("Alarm description: " +
alarm.alarmDescription());
            }
        } catch (CloudWatchException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
```

```
        }
    }

    public static String createAlarm(CloudWatchClient cw, String fileName) {
        try {
            // Read values from the JSON file.
            JsonParser parser = new JsonFactory().createParser(new
File(fileName));
            com.fasterxml.jackson.databind.JsonNode rootNode = new
ObjectMapper().readTree(parser);
            String customMetricNamespace =
rootNode.findValue("customMetricNamespace").asText();
            String customMetricName =
rootNode.findValue("customMetricName").asText();
            String alarmName = rootNode.findValue("exampleAlarmName").asText();
            String emailTopic = rootNode.findValue("emailTopic").asText();
            String accountId = rootNode.findValue("accountId").asText();
            String region = rootNode.findValue("region").asText();

            // Create a List for alarm actions.
            List<String> alarmActions = new ArrayList<>();
            alarmActions.add("arn:aws:sns:" + region + ":" + accountId + ":" +
emailTopic);
            PutMetricAlarmRequest alarmRequest = PutMetricAlarmRequest.builder()
                    .alarmActions(alarmActions)
                    .alarmDescription("Example metric alarm")
                    .alarmName(alarmName)

.comparisonOperator(ComparisonOperator.GREATER_THAN_OR_EQUAL_TO_THRESHOLD)
                    .threshold(100.00)
                    .metricName(customMetricName)
                    .namespace(customMetricNamespace)
                    .evaluationPeriods(1)
                    .period(10)
                    .statistic("Maximum")
                    .datapointsToAlarm(1)
                    .treatMissingData("ignore")
                    .build();

            cw.putMetricAlarm(alarmRequest);
            System.out.println(alarmName + " was successfully created!");
            return alarmName;

        } catch (CloudWatchException | IOException e) {
```

```
            System.err.println(e.getMessage());
            System.exit(1);
        }
        return "";
    }

    public static void addMetricToDashboard(CloudWatchClient cw, String fileName,
String dashboardName) {
        try {
            PutDashboardRequest dashboardRequest = PutDashboardRequest.builder()
                    .dashboardName(dashboardName)
                    .dashboardBody(readFileAsString(fileName))
                    .build();

            cw.putDashboard(dashboardRequest);
            System.out.println(dashboardName + " was successfully updated.");

        } catch (CloudWatchException | IOException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }

    public static void createNewCustomMetric(CloudWatchClient cw, Double
dataPoint) {
        try {
            Dimension dimension = Dimension.builder()
                    .name("UNIQUE_PAGES")
                    .value("URLS")
                    .build();

            // Set an Instant object.
            String time =
ZonedDateTime.now(ZoneOffset.UTC).format(DateTimeFormatter.ISO_INSTANT);
            Instant instant = Instant.parse(time);

            MetricDatum datum = MetricDatum.builder()
                    .metricName("PAGES_VISITED")
                    .unit(StandardUnit.NONE)
                    .value(dataPoint)
                    .timestamp(instant)
                    .dimensions(dimension)
                    .build();
```

```
            PutMetricDataRequest request = PutMetricDataRequest.builder()
                    .namespace("SITE/TRAFFIC")
                    .metricData(datum)
                    .build();

            cw.putMetricData(request);
            System.out.println("Added metric values for for metric
PAGES_VISITED");

        } catch (CloudWatchException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }

    public static void listDashboards(CloudWatchClient cw) {
        try {
            ListDashboardsIterable listRes = cw.listDashboardsPaginator();
            listRes.stream()
                    .flatMap(r -> r.dashboardEntries().stream())
                    .forEach(entry -> {
                        System.out.println("Dashboard name is: " +
entry.dashboardName());
                        System.out.println("Dashboard ARN is: " +
entry.dashboardArn());
                    });

        } catch (CloudWatchException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }

    public static void createDashboardWithMetrics(CloudWatchClient cw, String
dashboardName, String fileName) {
        try {
            PutDashboardRequest dashboardRequest = PutDashboardRequest.builder()
                    .dashboardName(dashboardName)
                    .dashboardBody(readFileAsString(fileName))
                    .build();

            PutDashboardResponse response = cw.putDashboard(dashboardRequest);
            System.out.println(dashboardName + " was successfully created.");
```

```
            List<DashboardValidationMessage> messages =
response.dashboardValidationMessages();
            if (messages.isEmpty()) {
                System.out.println("There are no messages in the new Dashboard");
            } else {
                for (DashboardValidationMessage message : messages) {
                    System.out.println("Message is: " + message.message());
                }
            }

        } catch (CloudWatchException | IOException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }

    public static String readFileAsString(String file) throws IOException {
        return new String(Files.readAllBytes(Paths.get(file)));
    }

    public static void getMetricStatistics(CloudWatchClient cw, String
costDateWeek) {
        try {
            Instant start = Instant.parse(costDateWeek);
            Instant endDate = Instant.now();
            Dimension dimension = Dimension.builder()
                    .name("Currency")
                    .value("USD")
                    .build();

            List<Dimension> dimensionList = new ArrayList<>();
            dimensionList.add(dimension);
            GetMetricStatisticsRequest statisticsRequest =
GetMetricStatisticsRequest.builder()
                    .metricName("EstimatedCharges")
                    .namespace("AWS/Billing")
                    .dimensions(dimensionList)
                    .statistics(Statistic.MAXIMUM)
                    .startTime(start)
                    .endTime(endDate)
                    .period(86400)
                    .build();
```

```
            GetMetricStatisticsResponse response =
cw.getMetricStatistics(statisticsRequest);
            List<Datapoint> data = response.datapoints();
            if (!data.isEmpty()) {
                for (Datapoint datapoint : data) {
                    System.out
                            .println("Timestamp: " + datapoint.timestamp() + "
Maximum value: " + datapoint.maximum());
                }
            } else {
                System.out.println("The returned data list is empty");
            }

        } catch (CloudWatchException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }

    public static void getAndDisplayMetricStatistics(CloudWatchClient cw, String
nameSpace, String metVal,
            String metricOption, String date, Dimension myDimension) {
        try {
            Instant start = Instant.parse(date);
            Instant endDate = Instant.now();

            GetMetricStatisticsRequest statisticsRequest =
GetMetricStatisticsRequest.builder()
                    .endTime(endDate)
                    .startTime(start)
                    .dimensions(myDimension)
                    .metricName(metVal)
                    .namespace(nameSpace)
                    .period(86400)
                    .statistics(Statistic.fromValue(metricOption))
                    .build();

            GetMetricStatisticsResponse response =
cw.getMetricStatistics(statisticsRequest);
            List<Datapoint> data = response.datapoints();
            if (!data.isEmpty()) {
                for (Datapoint datapoint : data) {
                    System.out
```

```
                            .println("Timestamp: " + datapoint.timestamp() + "
Maximum value: " + datapoint.maximum());
                }
            } else {
                System.out.println("The returned data list is empty");
            }

        } catch (CloudWatchException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }

    public static Dimension getSpecificMet(CloudWatchClient cw, String namespace)
{
        try {
            ListMetricsRequest request = ListMetricsRequest.builder()
                    .namespace(namespace)
                    .build();

            ListMetricsResponse response = cw.listMetrics(request);
            List<Metric> myList = response.metrics();
            Metric metric = myList.get(0);
            return metric.dimensions().get(0);

        } catch (CloudWatchException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        return null;
    }

    public static ArrayList<String> listMets(CloudWatchClient cw, String
namespace) {
        try {
            ArrayList<String> metList = new ArrayList<>();
            ListMetricsRequest request = ListMetricsRequest.builder()
                    .namespace(namespace)
                    .build();

            ListMetricsIterable listRes = cw.listMetricsPaginator(request);
            listRes.stream()
                    .flatMap(r -> r.metrics().stream())
                    .forEach(metrics -> metList.add(metrics.metricName()));
```

```
            return metList;

        } catch (CloudWatchException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        return null;
    }

    public static ArrayList<String> listNameSpaces(CloudWatchClient cw) {
        try {
            ArrayList<String> nameSpaceList = new ArrayList<>();
            ListMetricsRequest request = ListMetricsRequest.builder()
                    .build();

            ListMetricsIterable listRes = cw.listMetricsPaginator(request);
            listRes.stream()
                    .flatMap(r -> r.metrics().stream())
                    .forEach(metrics -> {
                        String data = metrics.namespace();
                        if (!nameSpaceList.contains(data)) {
                            nameSpaceList.add(data);
                        }
                    });

            return nameSpaceList;
        } catch (CloudWatchException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        return null;
    }
}
```

- Para obtener información sobre la API, consulte los siguientes temas en la referencia de la API de AWS SDK for Java 2.x.
  - DeleteAlarms
  - DeleteAnomalyDetector
  - DeleteDashboards
  - DescribeAlarmHistory

- DescribeAlarms
- DescribeAlarmsForMetric
- DescribeAnomalyDetectors
- GetMetricData
- GetMetricStatistics
- GetMetricWidgetImage
- ListMetrics
- PutAnomalyDetector
- PutDashboard
- PutMetricAlarm
- PutMetricData

Kotlin

SDK para Kotlin

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
/**
 Before running this Kotlin code example, set up your development environment,
 including your credentials.

 For more information, see the following documentation topic:
 https://docs.aws.amazon.com/sdk-for-kotlin/latest/developer-guide/setup.html

 To enable billing metrics and statistics for this example, make sure billing
 alerts are enabled for your account:
 https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/
monitor_estimated_charges_with_cloudwatch.html#turning_on_billing_metrics

 This Kotlin code example performs the following tasks:
```

```
 1. List available namespaces from Amazon CloudWatch. Select a namespace from the
 list.
 2. List available metrics within the selected namespace.
 3. Get statistics for the selected metric over the last day.
 4. Get CloudWatch estimated billing for the last week.
 5. Create a new CloudWatch dashboard with metrics.
 6. List dashboards using a paginator.
 7. Create a new custom metric by adding data for it.
 8. Add the custom metric to the dashboard.
 9. Create an alarm for the custom metric.
 10. Describe current alarms.
 11. Get current data for the new custom metric.
 12. Push data into the custom metric to trigger the alarm.
 13. Check the alarm state using the action DescribeAlarmsForMetric.
 14. Get alarm history for the new alarm.
 15. Add an anomaly detector for the custom metric.
 16. Describe current anomaly detectors.
 17. Get a metric image for the custom metric.
 18. Clean up the Amazon CloudWatch resources.
 */

val DASHES: String? = String(CharArray(80)).replace("\u0000", "-")
suspend fun main(args: Array<String>) {
    val usage = """
        Usage:
            <myDate> <costDateWeek> <dashboardName> <dashboardJson>
 <dashboardAdd> <settings> <metricImage>

        Where:
            myDate - The start date to use to get metric statistics. (For
 example, 2023-01-11T18:35:24.00Z.)
            costDateWeek - The start date to use to get AWS Billing and Cost
 Management statistics. (For example, 2023-01-11T18:35:24.00Z.)
            dashboardName - The name of the dashboard to create.
            dashboardJson - The location of a JSON file to use to create a
 dashboard. (See Readme file.)
            dashboardAdd - The location of a JSON file to use to update a
 dashboard. (See Readme file.)
            settings - The location of a JSON file from which various values are
 read. (See Readme file.)
            metricImage - The location of a BMP file that is used to create a
 graph.
    """
```

```
    if (args.size != 7) {
        println(usage)
        System.exit(1)
    }

    val myDate = args[0]
    val costDateWeek = args[1]
    val dashboardName = args[2]
    val dashboardJson = args[3]
    val dashboardAdd = args[4]
    val settings = args[5]
    var metricImage = args[6]
    val dataPoint = "10.0".toDouble()
    val inOb = Scanner(System.`in`)

    println(DASHES)
    println("Welcome to the Amazon CloudWatch example scenario.")
    println(DASHES)

    println(DASHES)
    println("1. List at least five available unique namespaces from Amazon
 CloudWatch. Select a CloudWatch namespace from the list.")
    val list: ArrayList<String> = listNameSpaces()
    for (z in 0..4) {
        println("    ${z + 1}. ${list[z]}")
    }

    var selectedNamespace: String
    var selectedMetrics = ""
    var num = inOb.nextLine().toInt()
    println("You selected $num")

    if (1 <= num && num <= 5) {
        selectedNamespace = list[num - 1]
    } else {
        println("You did not select a valid option.")
        exitProcess(1)
    }
    println("You selected $selectedNamespace")
    println(DASHES)

    println(DASHES)
    println("2. List available metrics within the selected namespace and select
 one from the list.")
```

```
    val metList = listMets(selectedNamespace)
    for (z in 0..4) {
        println("    ${ z + 1}. ${metList?.get(z)}")
    }
    num = inOb.nextLine().toInt()
    if (1 <= num && num <= 5) {
        selectedMetrics = metList!![num - 1]
    } else {
        println("You did not select a valid option.")
        System.exit(1)
    }
    println("You selected $selectedMetrics")
    val myDimension = getSpecificMet(selectedNamespace)
    if (myDimension == null) {
        println("Error - Dimension is null")
        exitProcess(1)
    }
    println(DASHES)

    println(DASHES)
    println("3. Get statistics for the selected metric over the last day.")
    val metricOption: String
    val statTypes = ArrayList<String>()
    statTypes.add("SampleCount")
    statTypes.add("Average")
    statTypes.add("Sum")
    statTypes.add("Minimum")
    statTypes.add("Maximum")

    for (t in 0..4) {
        println("    ${t + 1}. ${statTypes[t]}")
    }
    println("Select a metric statistic by entering a number from the preceding
list:")
    num = inOb.nextLine().toInt()
    if (1 <= num && num <= 5) {
        metricOption = statTypes[num - 1]
    } else {
        println("You did not select a valid option.")
        exitProcess(1)
    }
    println("You selected $metricOption")
    getAndDisplayMetricStatistics(selectedNamespace, selectedMetrics,
metricOption, myDate, myDimension)
```

```
    println(DASHES)

    println(DASHES)
    println("4. Get CloudWatch estimated billing for the last week.")
    getMetricStatistics(costDateWeek)
    println(DASHES)

    println(DASHES)
    println("5. Create a new CloudWatch dashboard with metrics.")
    createDashboardWithMetrics(dashboardName, dashboardJson)
    println(DASHES)

    println(DASHES)
    println("6. List dashboards using a paginator.")
    listDashboards()
    println(DASHES)

    println(DASHES)
    println("7. Create a new custom metric by adding data to it.")
    createNewCustomMetric(dataPoint)
    println(DASHES)

    println(DASHES)
    println("8. Add an additional metric to the dashboard.")
    addMetricToDashboard(dashboardAdd, dashboardName)
    println(DASHES)

    println(DASHES)
    println("9. Create an alarm for the custom metric.")
    val alarmName: String = createAlarm(settings)
    println(DASHES)

    println(DASHES)
    println("10. Describe 10 current alarms.")
    describeAlarms()
    println(DASHES)

    println(DASHES)
    println("11. Get current data for the new custom metric.")
    getCustomMetricData(settings)
    println(DASHES)

    println(DASHES)
    println("12. Push data into the custom metric to trigger the alarm.")
```

```
    addMetricDataForAlarm(settings)
    println(DASHES)

    println(DASHES)
    println("13. Check the alarm state using the action
 DescribeAlarmsForMetric.")
    checkForMetricAlarm(settings)
    println(DASHES)

    println(DASHES)
    println("14. Get alarm history for the new alarm.")
    getAlarmHistory(settings, myDate)
    println(DASHES)

    println(DASHES)
    println("15. Add an anomaly detector for the custom metric.")
    addAnomalyDetector(settings)
    println(DASHES)

    println(DASHES)
    println("16. Describe current anomaly detectors.")
    describeAnomalyDetectors(settings)
    println(DASHES)

    println(DASHES)
    println("17. Get a metric image for the custom metric.")
    getAndOpenMetricImage(metricImage)
    println(DASHES)

    println(DASHES)
    println("18. Clean up the Amazon CloudWatch resources.")
    deleteDashboard(dashboardName)
    deleteAlarm(alarmName)
    deleteAnomalyDetector(settings)
    println(DASHES)

    println(DASHES)
    println("The Amazon CloudWatch example scenario is complete.")
    println(DASHES)
}

suspend fun deleteAnomalyDetector(fileName: String) {
    // Read values from the JSON file.
    val parser = JsonFactory().createParser(File(fileName))
```

```
    val rootNode = ObjectMapper().readTree<JsonNode>(parser)
    val customMetricNamespace =
 rootNode.findValue("customMetricNamespace").asText()
    val customMetricName = rootNode.findValue("customMetricName").asText()

    val singleMetricAnomalyDetectorVal = SingleMetricAnomalyDetector {
        metricName = customMetricName
        namespace = customMetricNamespace
        stat = "Maximum"
    }

    val request = DeleteAnomalyDetectorRequest {
        singleMetricAnomalyDetector = singleMetricAnomalyDetectorVal
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        cwClient.deleteAnomalyDetector(request)
        println("Successfully deleted the Anomaly Detector.")
    }
}

suspend fun deleteAlarm(alarmNameVal: String) {
    val request = DeleteAlarmsRequest {
        alarmNames = listOf(alarmNameVal)
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        cwClient.deleteAlarms(request)
        println("Successfully deleted alarm $alarmNameVal")
    }
}

suspend fun deleteDashboard(dashboardName: String) {
    val dashboardsRequest = DeleteDashboardsRequest {
        dashboardNames = listOf(dashboardName)
    }
    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        cwClient.deleteDashboards(dashboardsRequest)
        println("$dashboardName was successfully deleted.")
    }
}

suspend fun getAndOpenMetricImage(fileName: String) {
    println("Getting Image data for custom metric.")
```

```
    val myJSON = """{
        "title": "Example Metric Graph",
        "view": "timeSeries",
        "stacked ": false,
        "period": 10,
        "width": 1400,
        "height": 600,
        "metrics": [
            [
            "AWS/Billing",
            "EstimatedCharges",
            "Currency",
            "USD"
            ]
        ]
        }"""

    val imageRequest = GetMetricWidgetImageRequest {
        metricWidget = myJSON
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        val response = cwClient.getMetricWidgetImage(imageRequest)
        val bytes = response.metricWidgetImage
        if (bytes != null) {
            File(fileName).writeBytes(bytes)
        }
    }
    println("You have successfully written data to $fileName")
}

suspend fun describeAnomalyDetectors(fileName: String) {
    // Read values from the JSON file.
    val parser = JsonFactory().createParser(File(fileName))
    val rootNode = ObjectMapper().readTree<JsonNode>(parser)
    val customMetricNamespace =
 rootNode.findValue("customMetricNamespace").asText()
    val customMetricName = rootNode.findValue("customMetricName").asText()

    val detectorsRequest = DescribeAnomalyDetectorsRequest {
        maxResults = 10
        metricName = customMetricName
        namespace = customMetricNamespace
    }
```

```
    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        val response = cwClient.describeAnomalyDetectors(detectorsRequest)
        response.anomalyDetectors?.forEach { detector ->
            println("Metric name:
 ${detector.singleMetricAnomalyDetector?.metricName}")
            println("State: ${detector.stateValue}")
        }
    }
}

suspend fun addAnomalyDetector(fileName: String?) {
    // Read values from the JSON file.
    val parser = JsonFactory().createParser(File(fileName))
    val rootNode = ObjectMapper().readTree<JsonNode>(parser)
    val customMetricNamespace =
 rootNode.findValue("customMetricNamespace").asText()
    val customMetricName = rootNode.findValue("customMetricName").asText()

    val singleMetricAnomalyDetectorVal = SingleMetricAnomalyDetector {
        metricName = customMetricName
        namespace = customMetricNamespace
        stat = "Maximum"
    }

    val anomalyDetectorRequest = PutAnomalyDetectorRequest {
        singleMetricAnomalyDetector = singleMetricAnomalyDetectorVal
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        cwClient.putAnomalyDetector(anomalyDetectorRequest)
        println("Added anomaly detector for metric $customMetricName.")
    }
}

suspend fun getAlarmHistory(fileName: String, date: String) {
    // Read values from the JSON file.
    val parser = JsonFactory().createParser(File(fileName))
    val rootNode = ObjectMapper().readTree<JsonNode>(parser)
    val alarmNameVal = rootNode.findValue("exampleAlarmName").asText()
    val start = Instant.parse(date)
    val endDateVal = Instant.now()

    val historyRequest = DescribeAlarmHistoryRequest {
        startDate = aws.smithy.kotlin.runtime.time.Instant(start)
```

```
        endDate = aws.smithy.kotlin.runtime.time.Instant(endDateVal)
        alarmName = alarmNameVal
        historyItemType = HistoryItemType.Action
    }

    CloudWatchClient { credentialsProvider = EnvironmentCredentialsProvider();
 region = "us-east-1" }.use { cwClient ->
        val response = cwClient.describeAlarmHistory(historyRequest)
        val historyItems = response.alarmHistoryItems
        if (historyItems != null) {
            if (historyItems.isEmpty()) {
                println("No alarm history data found for $alarmNameVal.")
            } else {
                for (item in historyItems) {
                    println("History summary ${item.historySummary}")
                    println("Time stamp: ${item.timestamp}")
                }
            }
        }
    }
}

suspend fun checkForMetricAlarm(fileName: String?) {
    // Read values from the JSON file.
    val parser = JsonFactory().createParser(File(fileName))
    val rootNode = ObjectMapper().readTree<JsonNode>(parser)
    val customMetricNamespace =
 rootNode.findValue("customMetricNamespace").asText()
    val customMetricName = rootNode.findValue("customMetricName").asText()
    var hasAlarm = false
    var retries = 10

    val metricRequest = DescribeAlarmsForMetricRequest {
        metricName = customMetricName
        namespace = customMetricNamespace
    }
    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        while (!hasAlarm && retries > 0) {
            val response = cwClient.describeAlarmsForMetric(metricRequest)
            if (response.metricAlarms?.count()!! > 0) {
                hasAlarm = true
            }
            retries--
            delay(20000)
```

```
            println(".")
        }
        if (!hasAlarm) println("No Alarm state found for $customMetricName after
 10 retries.") else println("Alarm state found for $customMetricName.")
    }
}

suspend fun addMetricDataForAlarm(fileName: String?) {
    // Read values from the JSON file.
    val parser = JsonFactory().createParser(File(fileName))
    val rootNode = ObjectMapper().readTree<JsonNode>(parser)
    val customMetricNamespace =
 rootNode.findValue("customMetricNamespace").asText()
    val customMetricName = rootNode.findValue("customMetricName").asText()

    // Set an Instant object.
    val time =
 ZonedDateTime.now(ZoneOffset.UTC).format(DateTimeFormatter.ISO_INSTANT)
    val instant = Instant.parse(time)
    val datum = MetricDatum {
        metricName = customMetricName
        unit = StandardUnit.None
        value = 1001.00
        timestamp = aws.smithy.kotlin.runtime.time.Instant(instant)
    }

    val datum2 = MetricDatum {
        metricName = customMetricName
        unit = StandardUnit.None
        value = 1002.00
        timestamp = aws.smithy.kotlin.runtime.time.Instant(instant)
    }

    val metricDataList = ArrayList<MetricDatum>()
    metricDataList.add(datum)
    metricDataList.add(datum2)

    val request = PutMetricDataRequest {
        namespace = customMetricNamespace
        metricData = metricDataList
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        cwClient.putMetricData(request)
```

```
        println("Added metric values for for metric $customMetricName")
    }
}

suspend fun getCustomMetricData(fileName: String) {
    // Read values from the JSON file.
    val parser = JsonFactory().createParser(File(fileName))
    val rootNode = ObjectMapper().readTree<JsonNode>(parser)
    val customMetricNamespace =
 rootNode.findValue("customMetricNamespace").asText()
    val customMetricName = rootNode.findValue("customMetricName").asText()

    // Set the date.
    val nowDate = Instant.now()
    val hours: Long = 1
    val minutes: Long = 30
    val date2 = nowDate.plus(hours, ChronoUnit.HOURS).plus(
        minutes,
        ChronoUnit.MINUTES
    )

    val met = Metric {
        metricName = customMetricName
        namespace = customMetricNamespace
    }

    val metStat = MetricStat {
        stat = "Maximum"
        period = 1
        metric = met
    }

    val dataQUery = MetricDataQuery {
        metricStat = metStat
        id = "foo2"
        returnData = true
    }

    val dq = ArrayList<MetricDataQuery>()
    dq.add(dataQUery)
    val getMetReq = GetMetricDataRequest {
        maxDatapoints = 10
        scanBy = ScanBy.TimestampDescending
        startTime = aws.smithy.kotlin.runtime.time.Instant(nowDate)
```

```
        endTime = aws.smithy.kotlin.runtime.time.Instant(date2)
        metricDataQueries = dq
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        val response = cwClient.getMetricData(getMetReq)
        response.metricDataResults?.forEach { item ->
            println("The label is ${item.label}")
            println("The status code is ${item.statusCode}")
        }
    }
}

suspend fun describeAlarms() {
    val typeList = ArrayList<AlarmType>()
    typeList.add(AlarmType.MetricAlarm)
    val alarmsRequest = DescribeAlarmsRequest {
        alarmTypes = typeList
        maxRecords = 10
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        val response = cwClient.describeAlarms(alarmsRequest)
        response.metricAlarms?.forEach { alarm ->
            println("Alarm name: ${alarm.alarmName}")
            println("Alarm description: ${alarm.alarmDescription}")
        }
    }
}

suspend fun createAlarm(fileName: String): String {
    // Read values from the JSON file.
    val parser = JsonFactory().createParser(File(fileName))
    val rootNode: JsonNode = ObjectMapper().readTree(parser)
    val customMetricNamespace =
 rootNode.findValue("customMetricNamespace").asText()
    val customMetricName = rootNode.findValue("customMetricName").asText()
    val alarmNameVal = rootNode.findValue("exampleAlarmName").asText()
    val emailTopic = rootNode.findValue("emailTopic").asText()
    val accountId = rootNode.findValue("accountId").asText()
    val region2 = rootNode.findValue("region").asText()

    // Create a List for alarm actions.
    val alarmActionObs: MutableList<String> = ArrayList()
```

```
    alarmActionObs.add("arn:aws:sns:$region2:$accountId:$emailTopic")
    val alarmRequest = PutMetricAlarmRequest {
        alarmActions = alarmActionObs
        alarmDescription = "Example metric alarm"
        alarmName = alarmNameVal
        comparisonOperator = ComparisonOperator.GreaterThanOrEqualToThreshold
        threshold = 100.00
        metricName = customMetricName
        namespace = customMetricNamespace
        evaluationPeriods = 1
        period = 10
        statistic = Statistic.Maximum
        datapointsToAlarm = 1
        treatMissingData = "ignore"
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        cwClient.putMetricAlarm(alarmRequest)
        println("$alarmNameVal was successfully created!")
        return alarmNameVal
    }
}

suspend fun addMetricToDashboard(fileNameVal: String, dashboardNameVal: String) {
    val dashboardRequest = PutDashboardRequest {
        dashboardName = dashboardNameVal
        dashboardBody = readFileAsString(fileNameVal)
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        cwClient.putDashboard(dashboardRequest)
        println("$dashboardNameVal was successfully updated.")
    }
}

suspend fun createNewCustomMetric(dataPoint: Double) {
    val dimension = Dimension {
        name = "UNIQUE_PAGES"
        value = "URLS"
    }

    // Set an Instant object.
    val time =
 ZonedDateTime.now(ZoneOffset.UTC).format(DateTimeFormatter.ISO_INSTANT)
```

```
    val instant = Instant.parse(time)
    val datum = MetricDatum {
        metricName = "PAGES_VISITED"
        unit = StandardUnit.None
        value = dataPoint
        timestamp = aws.smithy.kotlin.runtime.time.Instant(instant)
        dimensions = listOf(dimension)
    }

    val request = PutMetricDataRequest {
        namespace = "SITE/TRAFFIC"
        metricData = listOf(datum)
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        cwClient.putMetricData(request)
        println("Added metric values for for metric PAGES_VISITED")
    }
}

suspend fun listDashboards() {
    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        cwClient.listDashboardsPaginated({})
            .transform { it.dashboardEntries?.forEach { obj -> emit(obj) } }
            .collect { obj ->
                println("Name is ${obj.dashboardName}")
                println("Dashboard ARN is ${obj.dashboardArn}")
            }
    }
}

suspend fun createDashboardWithMetrics(dashboardNameVal: String, fileNameVal:
 String) {
    val dashboardRequest = PutDashboardRequest {
        dashboardName = dashboardNameVal
        dashboardBody = readFileAsString(fileNameVal)
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        val response = cwClient.putDashboard(dashboardRequest)
        println("$dashboardNameVal was successfully created.")
        val messages = response.dashboardValidationMessages
        if (messages != null) {
            if (messages.isEmpty()) {
```

```
                println("There are no messages in the new Dashboard")
            } else {
                for (message in messages) {
                    println("Message is: ${message.message}")
                }
            }
        }
    }
}

fun readFileAsString(file: String): String {
    return String(Files.readAllBytes(Paths.get(file)))
}

suspend fun getMetricStatistics(costDateWeek: String?) {
    val start = Instant.parse(costDateWeek)
    val endDate = Instant.now()
    val dimension = Dimension {
        name = "Currency"
        value = "USD"
    }

    val dimensionList: MutableList<Dimension> = ArrayList()
    dimensionList.add(dimension)

    val statisticsRequest = GetMetricStatisticsRequest {
        metricName = "EstimatedCharges"
        namespace = "AWS/Billing"
        dimensions = dimensionList
        statistics = listOf(Statistic.Maximum)
        startTime = aws.smithy.kotlin.runtime.time.Instant(start)
        endTime = aws.smithy.kotlin.runtime.time.Instant(endDate)
        period = 86400
    }
    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        val response = cwClient.getMetricStatistics(statisticsRequest)
        val data: List<Datapoint>? = response.datapoints
        if (data != null) {
            if (!data.isEmpty()) {
                for (datapoint in data) {
                    println("Timestamp:  ${datapoint.timestamp} Maximum value:
 ${datapoint.maximum}")
                }
            } else {
```

```
                println("The returned data list is empty")
            }
        }
    }
}

suspend fun getAndDisplayMetricStatistics(nameSpaceVal: String, metVal: String,
 metricOption: String, date: String, myDimension: Dimension) {
    val start = Instant.parse(date)
    val endDate = Instant.now()
    val statisticsRequest = GetMetricStatisticsRequest {
        endTime = aws.smithy.kotlin.runtime.time.Instant(endDate)
        startTime = aws.smithy.kotlin.runtime.time.Instant(start)
        dimensions = listOf(myDimension)
        metricName = metVal
        namespace = nameSpaceVal
        period = 86400
        statistics = listOf(Statistic.fromValue(metricOption))
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        val response = cwClient.getMetricStatistics(statisticsRequest)
        val data = response.datapoints
        if (data != null) {
            if (data.isNotEmpty()) {
                for (datapoint in data) {
                    println("Timestamp: ${datapoint.timestamp} Maximum value:
 ${datapoint.maximum}")
                }
            } else {
                println("The returned data list is empty")
            }
        }
    }
}

suspend fun listMets(namespaceVal: String?): ArrayList<String>? {
    val metList = ArrayList<String>()
    val request = ListMetricsRequest {
        namespace = namespaceVal
    }
    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        val reponse = cwClient.listMetrics(request)
        reponse.metrics?.forEach { metrics ->
```

```
            val data = metrics.metricName
            if (!metList.contains(data)) {
                metList.add(data!!)
            }
        }
    }
    return metList
}

suspend fun getSpecificMet(namespaceVal: String?): Dimension? {
    val request = ListMetricsRequest {
        namespace = namespaceVal
    }
    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        val response = cwClient.listMetrics(request)
        val myList = response.metrics
        if (myList != null) {
            return myList[0].dimensions?.get(0)
        }
    }
    return null
}

suspend fun listNameSpaces(): ArrayList<String> {
    val nameSpaceList = ArrayList<String>()
    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        val response = cwClient.listMetrics(ListMetricsRequest {})
        response.metrics?.forEach { metrics ->
            val data = metrics.namespace
            if (!nameSpaceList.contains(data)) {
                nameSpaceList.add(data!!)
            }
        }
    }
    return nameSpaceList
}
```

- Para obtener información sobre la API, consulte los siguientes temas en la Referencia de la API de AWS SDK para Kotlin.
  - DeleteAlarms
  - DeleteAnomalyDetector

- DeleteDashboards
- DescribeAlarmHistory
- DescribeAlarms
- DescribeAlarmsForMetric
- DescribeAnomalyDetectors
- GetMetricData
- GetMetricStatistics
- GetMetricWidgetImage
- ListMetrics
- PutAnomalyDetector
- PutDashboard
- PutMetricAlarm
- PutMetricData

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de CloudWatch con SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Administración de métricas y alarmas de CloudWatch mediante un SDK de AWS

En el siguiente ejemplo de código, se muestra cómo:

- Crear una alarma para vigilar una métrica de CloudWatch.
- Coloque los datos en una métrica y desencadene la alarma.
- Obtenga datos de la alarma.
- Elimine la alarma.

Python

SDK para Python (Boto3)

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Cree una clase que ajuste las operaciones de CloudWatch.

```
from datetime import datetime, timedelta
import logging
from pprint import pprint
import random
import time
import boto3
from botocore.exceptions import ClientError

logger = logging.getLogger(__name__)


class CloudWatchWrapper:
    """Encapsulates Amazon CloudWatch functions."""

    def __init__(self, cloudwatch_resource):
        """
        :param cloudwatch_resource: A Boto3 CloudWatch resource.
        """
        self.cloudwatch_resource = cloudwatch_resource


    def put_metric_data_set(self, namespace, name, timestamp, unit, data_set):
        """
        Sends a set of data to CloudWatch for a metric. All of the data in the
 set
        have the same timestamp and unit.

        :param namespace: The namespace of the metric.
        :param name: The name of the metric.
        :param timestamp: The UTC timestamp for the metric.
        :param unit: The unit of the metric.
```

```
        :param data_set: The set of data to send. This set is a dictionary that
                         contains a list of values and a list of corresponding
counts.
                         The value and count lists must be the same length.
        """
        try:
            metric = self.cloudwatch_resource.Metric(namespace, name)
            metric.put_data(
                Namespace=namespace,
                MetricData=[
                    {
                        "MetricName": name,
                        "Timestamp": timestamp,
                        "Values": data_set["values"],
                        "Counts": data_set["counts"],
                        "Unit": unit,
                    }
                ],
            )
            logger.info("Put data set for metric %s.%s.", namespace, name)
        except ClientError:
            logger.exception("Couldn't put data set for metric %s.%s.",
namespace, name)
            raise


    def create_metric_alarm(
        self,
        metric_namespace,
        metric_name,
        alarm_name,
        stat_type,
        period,
        eval_periods,
        threshold,
        comparison_op,
    ):
        """
        Creates an alarm that watches a metric.

        :param metric_namespace: The namespace of the metric.
        :param metric_name: The name of the metric.
        :param alarm_name: The name of the alarm.
        :param stat_type: The type of statistic the alarm watches.
```

```
        :param period: The period in which metric data are grouped to calculate
                       statistics.
        :param eval_periods: The number of periods that the metric must be over
the
                             alarm threshold before the alarm is set into an
alarmed
                             state.
        :param threshold: The threshold value to compare against the metric
statistic.
        :param comparison_op: The comparison operation used to compare the
threshold
                              against the metric.
        :return: The newly created alarm.
        """
        try:
            metric = self.cloudwatch_resource.Metric(metric_namespace,
metric_name)
            alarm = metric.put_alarm(
                AlarmName=alarm_name,
                Statistic=stat_type,
                Period=period,
                EvaluationPeriods=eval_periods,
                Threshold=threshold,
                ComparisonOperator=comparison_op,
            )
            logger.info(
                "Added alarm %s to track metric %s.%s.",
                alarm_name,
                metric_namespace,
                metric_name,
            )
        except ClientError:
            logger.exception(
                "Couldn't add alarm %s to metric %s.%s",
                alarm_name,
                metric_namespace,
                metric_name,
            )
            raise
        else:
            return alarm


    def put_metric_data(self, namespace, name, value, unit):
```

```
        """
        Sends a single data value to CloudWatch for a metric. This metric is
given
        a timestamp of the current UTC time.

        :param namespace: The namespace of the metric.
        :param name: The name of the metric.
        :param value: The value of the metric.
        :param unit: The unit of the metric.
        """
        try:
            metric = self.cloudwatch_resource.Metric(namespace, name)
            metric.put_data(
                Namespace=namespace,
                MetricData=[{"MetricName": name, "Value": value, "Unit": unit}],
            )
            logger.info("Put data for metric %s.%s", namespace, name)
        except ClientError:
            logger.exception("Couldn't put data for metric %s.%s", namespace,
name)
            raise


    def get_metric_statistics(self, namespace, name, start, end, period,
stat_types):
        """
        Gets statistics for a metric within a specified time span. Metrics are
grouped
        into the specified period.

        :param namespace: The namespace of the metric.
        :param name: The name of the metric.
        :param start: The UTC start time of the time span to retrieve.
        :param end: The UTC end time of the time span to retrieve.
        :param period: The period, in seconds, in which to group metrics. The
period
                       must match the granularity of the metric, which depends on
                       the metric's age. For example, metrics that are older than
                       three hours have a one-minute granularity, so the period
must
                       be at least 60 and must be a multiple of 60.
        :param stat_types: The type of statistics to retrieve, such as average
value
                           or maximum value.
```

```
        :return: The retrieved statistics for the metric.
        """
        try:
            metric = self.cloudwatch_resource.Metric(namespace, name)
            stats = metric.get_statistics(
                StartTime=start, EndTime=end, Period=period,
Statistics=stat_types
            )
            logger.info(
                "Got %s statistics for %s.", len(stats["Datapoints"]),
stats["Label"]
            )
        except ClientError:
            logger.exception("Couldn't get statistics for %s.%s.", namespace,
name)
            raise
        else:
            return stats


    def get_metric_alarms(self, metric_namespace, metric_name):
        """
        Gets the alarms that are currently watching the specified metric.

        :param metric_namespace: The namespace of the metric.
        :param metric_name: The name of the metric.
        :returns: An iterator that yields the alarms.
        """
        metric = self.cloudwatch_resource.Metric(metric_namespace, metric_name)
        alarm_iter = metric.alarms.all()
        logger.info("Got alarms for metric %s.%s.", metric_namespace,
metric_name)
        return alarm_iter


    def delete_metric_alarms(self, metric_namespace, metric_name):
        """
        Deletes all of the alarms that are currently watching the specified
metric.

        :param metric_namespace: The namespace of the metric.
        :param metric_name: The name of the metric.
        """
        try:
```

```
            metric = self.cloudwatch_resource.Metric(metric_namespace,
 metric_name)
            metric.alarms.delete()
            logger.info(
                "Deleted alarms for metric %s.%s.", metric_namespace, metric_name
            )
        except ClientError:
            logger.exception(
                "Couldn't delete alarms for metric %s.%s.",
                metric_namespace,
                metric_name,
            )
            raise
```

Utilice la clase wrapper para colocar datos en una métrica, desencadenar una alarma que observa la métrica y obtener datos de la alarma.

```
def usage_demo():
    print("-" * 88)
    print("Welcome to the Amazon CloudWatch metrics and alarms demo!")
    print("-" * 88)

    logging.basicConfig(level=logging.INFO, format="%(levelname)s: %(message)s")

    cw_wrapper = CloudWatchWrapper(boto3.resource("cloudwatch"))

    minutes = 20
    metric_namespace = "doc-example-metric"
    metric_name = "page_views"
    start = datetime.utcnow() - timedelta(minutes=minutes)
    print(
        f"Putting data into metric {metric_namespace}.{metric_name} spanning the
 "
        f"last {minutes} minutes."
    )
    for offset in range(0, minutes):
        stamp = start + timedelta(minutes=offset)
        cw_wrapper.put_metric_data_set(
            metric_namespace,
            metric_name,
```

```
            stamp,
            "Count",
            {
                "values": [
                    random.randint(bound, bound * 2)
                    for bound in range(offset + 1, offset + 11)
                ],
                "counts": [random.randint(1, offset + 1) for _ in range(10)],
            },
        )

    alarm_name = "high_page_views"
    period = 60
    eval_periods = 2
    print(f"Creating alarm {alarm_name} for metric {metric_name}.")
    alarm = cw_wrapper.create_metric_alarm(
        metric_namespace,
        metric_name,
        alarm_name,
        "Maximum",
        period,
        eval_periods,
        100,
        "GreaterThanThreshold",
    )
    print(f"Alarm ARN is {alarm.alarm_arn}.")
    print(f"Current alarm state is: {alarm.state_value}.")

    print(
        f"Sending data to trigger the alarm. This requires data over the
threshold "
        f"for {eval_periods} periods of {period} seconds each."
    )
    while alarm.state_value == "INSUFFICIENT_DATA":
        print("Sending data for the metric.")
        cw_wrapper.put_metric_data(
            metric_namespace, metric_name, random.randint(100, 200), "Count"
        )
        alarm.load()
        print(f"Current alarm state is: {alarm.state_value}.")
        if alarm.state_value == "INSUFFICIENT_DATA":
            print(f"Waiting for {period} seconds...")
            time.sleep(period)
        else:
```

```
            print("Wait for a minute for eventual consistency of metric data.")
            time.sleep(period)
            if alarm.state_value == "OK":
                alarm.load()
                print(f"Current alarm state is: {alarm.state_value}.")

    print(
        f"Getting data for metric {metric_namespace}.{metric_name} during
timespan "
        f"of {start} to {datetime.utcnow()} (times are UTC)."
    )
    stats = cw_wrapper.get_metric_statistics(
        metric_namespace,
        metric_name,
        start,
        datetime.utcnow(),
        60,
        ["Average", "Minimum", "Maximum"],
    )
    print(
        f"Got {len(stats['Datapoints'])} data points for metric "
        f"{metric_namespace}.{metric_name}."
    )
    pprint(sorted(stats["Datapoints"], key=lambda x: x["Timestamp"]))

    print(f"Getting alarms for metric {metric_name}.")
    alarms = cw_wrapper.get_metric_alarms(metric_namespace, metric_name)
    for alarm in alarms:
        print(f"Alarm {alarm.name} is currently in state {alarm.state_value}.")

    print(f"Deleting alarms for metric {metric_name}.")
    cw_wrapper.delete_metric_alarms(metric_namespace, metric_name)

    print("Thanks for watching!")
    print("-" * 88)
```

- Para obtener información sobre la API, consulte los siguientes temas en la Referencia de la API de AWS SDK para Python (Boto3).
  - DeleteAlarms

- DescribeAlarmsForMetric
- DisableAlarmActions
- EnableAlarmActions
- GetMetricStatistics
- ListMetrics
- PutMetricAlarm
- PutMetricData

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de CloudWatch con SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Ejemplos de servicios combinados para CloudWatch con AWS SDK

Las siguientes aplicaciones de ejemplo utilizan AWS SDK para combinar CloudWatch con otros Servicios de AWS. Cada ejemplo incluye un enlace a GitHub, con instrucciones de configuración y ejecución de la aplicación.

Ejemplos

- Supervisión del rendimiento de Amazon DynamoDB mediante AWS SDK

Supervisión del rendimiento de Amazon DynamoDB mediante AWS SDK

En el siguiente ejemplo se muestra cómo configurar el uso de DynamoDB en una aplicación para supervisar el rendimiento.

Java

SDK para Java 2.x

En este ejemplo se muestra cómo configurar una aplicación Java para supervisar el rendimiento de DynamoDB. La aplicación envía los datos de las métricas a CloudWatch, donde usted puede supervisar el rendimiento.

Para ver el código fuente completo y las instrucciones de configuración y ejecución, consulte el ejemplo completo en GitHub.

Servicios utilizados en este ejemplo

- CloudWatch
- DynamoDB

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de CloudWatch con SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Seguridad en Amazon CloudWatch

La seguridad en la nube de AWS es la máxima prioridad. Como cliente de AWS, se beneficia de una arquitectura de red y un centro de datos diseñados para satisfacer los requisitos de seguridad de las organizaciones más exigentes.

La seguridad es una responsabilidad compartida entre AWS y usted. El modelo de responsabilidad compartida la describe como seguridad de la nube y seguridad en la nube:

- Seguridad de la nube: AWS es responsable de proteger la infraestructura que ejecuta los servicios de AWS en la nube de AWS. AWS también proporciona servicios que puede utilizar de forma segura. Los auditores externos prueban y verifican periódicamente la eficacia de nuestra seguridad como parte de los Programas de conformidad de AWS . Para conocer los programas de conformidad que se aplican a CloudWatch, consulte Servicios de AWS en el ámbito del programa de conformidad.
- Seguridad en la nube: su responsabilidad viene determinada por el servicio de AWS que utilice. También es responsable de otros factores, incluida la confidencialidad de los datos, los requisitos de la empresa y la legislación y los reglamentos aplicables

Esta documentación le ayuda a comprender cómo se aplica el modelo de responsabilidad compartida cuando se utiliza Amazon CloudWatch. Muestra cómo se configura Amazon CloudWatch para cumplir los objetivos de seguridad y conformidad. También obtendrá información sobre cómo se utilizan otros servicios de AWS que lo ayudarán a monitorear y proteger los recursos de CloudWatch.

Contenido

Protección de datos en Amazon CloudWatch

El AWS shared responsibility model (Modelo de responsabilidad compartida) se aplica a la protección de datos en Amazon CloudWatch. Como se describe en este modelo, AWS es responsable de proteger la infraestructura global que ejecuta la totalidad de Nube de AWS. Usted es responsable de mantener el control sobre el contenido alojado en esta infraestructura. Usted también es responsable de las tareas de administración y configuración de seguridad para los Servicios de AWS que utiliza. Para obtener más información sobre la privacidad de los datos, consulte las Preguntas frecuentes sobre la privacidad de datos. Para obtener información sobre la protección de datos en Europa, consulte la publicación de blog Modelo de responsabilidad compartida y GDPR de AWS en el Blog de seguridad de AWS.

Con fines de protección de datos, recomendamos proteger las credenciales de la cuenta de Cuenta de AWS y configurar cuentas de usuario individuales con AWS IAM Identity Center o AWS Identity and Access Management (IAM). De esta manera, solo se conceden a cada usuario los permisos necesarios para cumplir con sus obligaciones laborales. También recomendamos proteger sus datos de la siguiente manera:

- Utilice autenticación multifactor (MFA) en cada cuenta.
- Utilice SSL/TLS para comunicarse con los recursos de AWS. reSe recomienda el uso de TLS 1.2 y recomendamos TLS 1.3.
- Configure la API y el registro de actividad del usuario con AWS CloudTrail.
- Utilice las soluciones de cifrado de AWS, junto con todos los controles de seguridad predeterminados dentro de los Servicios de AWS.
- Utilice servicios de seguridad gestionados avanzados, como Amazon Macie, que lo ayuden a detectar y proteger los datos confidenciales almacenados en Amazon S3.
- Si necesita módulos criptográficos validados FIPS 140-2 al acceder a AWS a través de una interfaz de la línea de comandos o una API, utilice un punto de conexión de FIPS. Para obtener más información sobre los puntos de conexión de FIPS disponibles, consulte Estándar de procesamiento de la información federal (FIPS) 140-2.

Recomendamos encarecidamente que nunca introduzca información de identificación confidencial, como, por ejemplo, direcciones de email de sus clientes, en etiquetas o en los campos de formato libre, como el campo Name (Nombre). Esto incluye el momento en el que trabaje con CloudWatch u otros servicios de Servicios de AWS a través de consola, la API, la AWS CLI, o los SDK de AWS. Los datos que ingresa en las etiquetas o los campos de formato libre para los nombres se pueden

utilizar para los registros de facturación o diagnóstico. Si proporciona una URL a un servidor externo, recomendamos encarecidamente que no incluya información de credenciales en la URL a fin de validar la solicitud para ese servidor.

Cifrado en tránsito

CloudWatch utiliza el cifrado de extremo a extremo de los datos en tránsito.

Identity and Access Management para Amazon CloudWatch

AWS Identity and Access Management (IAM) es un Servicio de AWS que ayuda a los administradores a controlar de forma segura el acceso a los recursos de AWS. Los administradores de IAM controlan quién está autenticado (ha iniciado sesión) y autorizado (tiene permisos) para utilizar recursos de CloudWatch. IAM es un Servicio de AWS que se puede utilizar sin cargo adicional.

Temas

- Políticas administradas de AWS para Información de aplicaciones de Amazon CloudWatch
- Referencia de permisos de Amazon CloudWatch

Público

La forma en que utilice AWS Identity and Access Management (IAM) difiere en función del trabajo que realice en CloudWatch.

Usuario de servicio: si utiliza el servicio de CloudWatch para realizar el trabajo, el administrador le proporciona las credenciales y los permisos necesarios. Es posible que a medida que utilice más características de CloudWatch para realizar su trabajo, necesite permisos adicionales. Entender cómo se administra el acceso puede ayudarlo a solicitar los permisos correctos al administrador. Si no puede acceder a una característica en CloudWatch, consulte Resolución de problemas de identidad y acceso de Amazon CloudWatch.

Administrador de servicio: si está a cargo de los recursos de CloudWatch en la empresa, probablemente tenga acceso completo a CloudWatch. Su trabajo consiste en determinar a qué características y recursos de CloudWatch deben acceder los usuarios del servicio. Luego, debe enviar solicitudes a su administrador de IAM para cambiar los permisos de los usuarios de su servicio. Revise la información de esta página para conocer los conceptos básicos de IAM. Para obtener más información sobre cómo su empresa puede utilizar IAM con CloudWatch, consulte Cómo funciona Amazon CloudWatch con IAM.

Administrador de IAM: si es un administrador de IAM, es posible que desee obtener información sobre cómo escribir políticas para administrar el acceso a CloudWatch. Para consultar ejemplos de políticas basadas en identidad de CloudWatch que puede utilizar en IAM, consulte Ejemplos de políticas basadas en identidades para Amazon CloudWatch.

Autenticación con identidades

La autenticación es la manera de iniciar sesión en AWS mediante credenciales de identidad. Debe estar autenticado (haber iniciado sesión en AWS) como Usuario raíz de la cuenta de AWS, como un usuario de IAM o asumiendo un rol de IAM.

Puede iniciar sesión en AWS como una identidad federada mediante las credenciales proporcionadas a través de una fuente de identidad. AWS IAM Identity Center Los usuarios (del Centro de identidades de IAM), la autenticación de inicio de sesión único de su empresa y sus credenciales de Google o Facebook son ejemplos de identidades federadas. Al iniciar sesión

como una identidad federada, su administrador habrá configurado previamente la federación de identidades mediante roles de IAM. Cuando accede a AWS mediante la federación, está asumiendo un rol de forma indirecta.

Según el tipo de usuario que sea, puede iniciar sesión en AWS Management Console o en el portal de acceso AWS. Para obtener más información sobre el inicio de sesión en AWS, consulte Cómo iniciar sesión en su Cuenta de AWS en la Guía del usuario de AWS Sign-In.

Si accede a AWS mediante programación, AWS proporciona un kit de desarrollo de software (SDK) y una interfaz de la línea de comandos (CLI) para firmar criptográficamente las solicitudes mediante el uso de las credenciales. Si no usa las herramientas de AWS, debe firmar las solicitudes. Para obtener más información sobre la firma de solicitudes, consulte Firma de solicitudes API de AWS en la Guía del usuario de IAM.

Independientemente del método de autenticación que use, es posible que deba proporcionar información de seguridad adicional. Por ejemplo, AWS le recomienda el uso de la autenticación multifactor (MFA) para aumentar la seguridad de su cuenta. Para obtener más información, consulte Autenticación multifactor en la Guía del usuario de AWS IAM Identity Center y Uso de la autenticación multifactor (MFA) en AWSen la Guía del usuario de IAM.

Usuario raíz de Cuenta de AWS

Cuando se crea una Cuenta de AWS, se comienza con una identidad de inicio de sesión que tiene acceso completo a todos los recursos y Servicios de AWS de la cuenta. Esta identidad recibe el nombre de usuario raíz de la Cuenta de AWS y se accede a ella iniciando sesión con el email y la contraseña que utilizó para crear la cuenta. Recomendamos encarecidamente que no utilice el usuario raíz para sus tareas diarias. Proteja las credenciales del usuario raíz y utilícelas solo para las tareas que solo el usuario raíz pueda realizar. Para ver la lista completa de las tareas que requieren que inicie sesión como usuario raíz, consulte Tareas que requieren credenciales de usuario raíz en la Guía del usuario de IAM.

Identidad federada

Como práctica recomendada, solicite que los usuarios humanos, incluidos los que requieren acceso de administrador, utilicen la federación con un proveedor de identidades para acceder a los Servicios de AWS utilizando credenciales temporales.

Una identidad federada es un usuario del directorio de usuarios de su empresa, un proveedor de identidad web, el AWS Directory Service, el directorio del Identity Center, o cualquier usuario que acceda a Servicios de AWS utilizando credenciales proporcionadas a través de una fuente de

identidad. Cuando identidades federadas acceden a Cuentas de AWS, asumen roles y los roles proporcionan credenciales temporales.

Para una administración de acceso centralizada, le recomendamos que utilice AWS IAM Identity Center. Puede crear usuarios y grupos en el IAM Identity Center o puede conectarse y sincronizar con un conjunto de usuarios y grupos de su propia fuente de identidad para usarlos en todas sus aplicaciones y Cuentas de AWS. Para obtener más información, consulte ¿Qué es el Centro de identidades de IAM? en la Guía del usuario de AWS IAM Identity Center.

Usuarios y grupos de IAM

Un usuario de IAM es una identidad de la Cuenta de AWSque dispone de permisos específicos para una sola persona o aplicación. Siempre que sea posible, recomendamos emplear credenciales temporales, en lugar de crear usuarios de IAM que tengan credenciales de larga duración como contraseñas y claves de acceso. No obstante, si tiene casos de uso específicos que requieran credenciales de larga duración con usuarios de IAM, recomendamos rotar las claves de acceso. Para más información, consulte Rotar las claves de acceso periódicamente para casos de uso que requieran credenciales de larga duración en la Guía del usuario de IAM.

Un grupo de IAM es una identidad que especifica un conjunto de usuarios de IAM. No puede iniciar sesión como grupo. Puede usar los grupos para especificar permisos para varios usuarios a la vez. Los grupos facilitan la administración de los permisos de grandes conjuntos de usuarios. Por ejemplo, podría tener un grupo cuyo nombre fuese IAMAdmins y conceder permisos a dicho grupo para administrar los recursos de IAM.

Los usuarios son diferentes de los roles. Un usuario se asocia exclusivamente a una persona o aplicación, pero la intención es que cualquier usuario pueda asumir un rol que necesite. Los usuarios tienen credenciales permanentes a largo plazo y los roles proporcionan credenciales temporales. Para más información, consulte Cuándo crear un usuario de IAM (en lugar de un rol) en la Guía del usuario de IAM.

Roles de IAM

Un rol de IAM es una identidad de la Cuenta de AWSque dispone de permisos específicos. Es similar a un usuario de IAM, pero no está asociado a una determinada persona. Puede asumir temporalmente un rol de IAM en la AWS Management Console cambiando de roles. Puede asumir un rol llamando a una operación de la AWS CLI o de la API de AWS, o utilizando una URL personalizada. Para más información sobre los métodos para el uso de roles, consulte Uso de roles de IAM en la Guía del usuario de IAM.

Los roles de IAM con credenciales temporales son útiles en las siguientes situaciones:

- Acceso de usuario federado: para asignar permisos a una identidad federada, puede crear un rol y definir sus permisos. Cuando se autentica una identidad federada, se asocia la identidad al rol y se le conceden los permisos define el rol. Para obtener información acerca de roles para federación, consulte Creación de un rol para un proveedor de identidades de terceros en la Guía del usuario de IAM. Si utiliza IAM Identity Center, debe configurar un conjunto de permisos. IAM Identity Center correlaciona el conjunto de permisos con un rol en IAM para controlar a qué pueden acceder las identidades después de autenticarse. Para obtener información acerca de los conjuntos de permisos, consulte Conjuntos de permisos en la Guía del usuario de AWS IAM Identity Center.
- Permisos de usuario de IAM temporales: un usuario de IAM puede asumir un rol de IAM para recibir temporalmente permisos distintos que le permitan realizar una tarea concreta.
- Acceso entre cuentas: puede utilizar un rol de IAM para permitir que alguien (una entidad principal de confianza) de otra cuenta acceda a los recursos de la cuenta. Los roles son la forma principal de conceder acceso entre cuentas. No obstante, con algunos Servicios de AWS se puede adjuntar una política directamente a un recurso (en lugar de utilizar un rol como representante). Para obtener información acerca de la diferencia entre los roles y las políticas basadas en recursos para el acceso entre cuentas, consulte Cómo los roles de IAM difieren de las políticas basadas en recursos en la Guía del usuario de IAM.
- Acceso entre servicios: algunos Servicios de AWS utilizan características de otros Servicios de AWS. Por ejemplo, cuando realiza una llamada en un servicio, es común que ese servicio ejecute aplicaciones en Amazon EC2 o almacene objetos en Amazon S3. Es posible que un servicio haga esto usando los permisos de la entidad principal, usando un rol de servicio o usando un rol vinculado a servicios.
  - Reenviar sesiones de acceso (FAS): cuando utiliza un rol o un usuario de IAM para llevar a cabo acciones en AWS, se le considera una entidad principal. Cuando utiliza algunos servicios, es posible que realice una acción que desencadene otra acción en un servicio diferente. FAS utiliza los permisos de la entidad principal para llamar a un Servicio de AWS, combinados con el Servicio de AWS solicitante para realizar solicitudes a servicios posteriores. Las solicitudes de FAS solo se realizan cuando un servicio recibe una solicitud que requiere interacciones con otros Servicios de AWS o recursos para completarse. En este caso, debe tener permisos para realizar ambas acciones. Para obtener información sobre las políticas a la hora de realizar solicitudes de FAS, consulte Reenviar sesiones de acceso.
  - Rol de servicio: un rol de servicio es un rol de IAM que adopta un servicio para realizar acciones en su nombre. Un administrador de IAM puede crear, modificar y eliminar un rol de servicio

desde IAM. Para obtener más información, consulte Creación de un rol para delegar permisos a un Servicio de AWS en la Guía del usuario de IAM.

- Rol vinculado a los servicios: un rol vinculado a servicios es un tipo de rol de servicio que está vinculado a un Servicio de AWS. El servicio puede asumir el rol para realizar una acción en su nombre. Los roles vinculados a servicios aparecen en la Cuenta de AWS y son propiedad del servicio. Un administrador de IAM puede ver, pero no editar, los permisos de los roles vinculados a servicios.

- Aplicaciones que se ejecutan en Amazon EC2: puede utilizar un rol de IAM que le permita administrar credenciales temporales para las aplicaciones que se ejecutan en una instancia de EC2 y realizan solicitudes a la AWS CLI o a la API de AWS. Es preferible hacerlo de este modo a almacenar claves de acceso en la instancia de EC2. Para asignar un rol de AWS a una instancia de EC2 y ponerla a disposición de todas las aplicaciones, cree un perfil de instancia adjuntado a la instancia. Un perfil de instancia contiene el rol y permite a los programas que se ejecutan en la instancia de EC2 obtener credenciales temporales. Para más información, consulte Uso de un rol de IAM para conceder permisos a aplicaciones que se ejecutan en instancias Amazon EC2 en la Guía del usuario de IAM.

Para obtener información sobre el uso de los roles de IAM, consulte Cuándo crear un rol de IAM (en lugar de un usuario) en la Guía del usuario de IAM.

Administración de acceso mediante políticas

Para controlar el acceso en AWS, se crean políticas y se adjuntan a identidades o recursos de AWS. Una política es un objeto de AWS que, cuando se asocia a una identidad o un recurso, define sus permisos. AWS evalúa estas políticas cuando una entidad principal (sesión de rol, usuario o usuario raíz) realiza una solicitud. Los permisos en las políticas determinan si la solicitud se permite o se deniega. La mayoría de las políticas se almacenan en AWS como documentos JSON. Para obtener más información sobre la estructura y el contenido de los documentos de política JSON, consulte Información general de políticas JSON en la Guía del usuario de IAM.

Los administradores pueden utilizar las políticas JSON de AWS para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones.

De forma predeterminada, los usuarios y los roles no tienen permisos. Un administrador de IAM puede crear políticas de IAM para conceder permisos a los usuarios para realizar acciones en los recursos que necesitan. A continuación, el administrador puede añadir las políticas de IAM a roles y los usuarios pueden asumirlos.

Las políticas de IAM definen permisos para una acción independientemente del método que se utilice para realizar la operación. Por ejemplo, suponga que dispone de una política que permite la acción `iam:GetRole`. Un usuario con dicha política puede obtener información del usuario de la AWS Management Console, la AWS CLI o la API de AWS.

Políticas basadas en identidades

Las políticas basadas en identidad son documentos de políticas de permisos JSON que puede asociar a una identidad, como un usuario de IAM, un grupo de usuarios o un rol. Estas políticas controlan qué acciones pueden realizar los usuarios y los roles, en qué recursos y en qué condiciones. Para obtener más información sobre cómo crear una política basada en identidad, consulte Creación de políticas de IAM en la Guía del usuario de IAM.

Las políticas basadas en identidades pueden clasificarse además como políticas insertadas o políticas administradas. Las políticas insertadas se integran directamente en un único usuario, grupo o rol. Las políticas administradas son políticas independientes que puede adjuntar a varios usuarios, grupos y roles de su Cuenta de AWS. Las políticas administradas incluyen las políticas administradas por AWS y las políticas administradas por el cliente. Para más información sobre cómo elegir una política administrada o una política insertada, consulte Elegir entre políticas administradas y políticas insertadas en la Guía del usuario de IAM.

Políticas basadas en recursos

Las políticas basadas en recursos son documentos de política JSON que se asocian a un recurso. Ejemplos de políticas basadas en recursos son las políticas de confianza de roles de IAM y las políticas de bucket de Amazon S3. En los servicios que admiten políticas basadas en recursos, los administradores de servicios pueden utilizarlos para controlar el acceso a un recurso específico. Para el recurso al que se asocia la política, la política define qué acciones puede realizar una entidad principal especificada en ese recurso y en qué condiciones. Debe especificar una entidad principal en una política en función de recursos. Las entidades principales pueden incluir cuentas, usuarios, roles, usuarios federados o Servicios de AWS.

Las políticas basadas en recursos son políticas insertadas que se encuentran en ese servicio. No se puede utilizar políticas de IAM administradas por AWS en una política basada en recursos.

Listas de control de acceso (ACL)

Las listas de control de acceso (ACL) controlan qué entidades principales (miembros de cuentas, usuarios o roles) tienen permisos para acceder a un recurso. Las ACL son similares a las políticas basadas en recursos, aunque no utilizan el formato de documento de políticas JSON.

Amazon S3, AWS WAF y Amazon VPC son ejemplos de servicios que admiten las ACL. Para obtener más información sobre las ACL, consulte Información general de Lista de control de acceso (ACL) en la Guía para desarrolladores de Amazon Simple Storage Service.

Otros tipos de políticas

AWS admite otros tipos de políticas adicionales menos frecuentes. Estos tipos de políticas pueden establecer el máximo de permisos que los tipos de políticas más frecuentes le conceden.

- Límites de permisos: un límite de permisos es una característica avanzada que le permite establecer los permisos máximos que una política basada en identidad puede conceder a una entidad de IAM (usuario o rol de IAM). Puede establecer un límite de permisos para una entidad. Los permisos resultantes son la intersección de las políticas basadas en la identidad de la entidad y los límites de permisos. Las políticas basadas en recursos que especifiquen el usuario o rol en el campo `Principal` no estarán restringidas por el límite de permisos. Una denegación explícita en cualquiera de estas políticas anulará el permiso. Para obtener más información sobre los límites de los permisos, consulte Límites de permisos para las entidades de IAM en la Guía del usuario de IAM.
- Políticas de control de servicio (SCP): las SCP son políticas de JSON que especifican los permisos máximos de una organización o una unidad organizativa en AWS Organizations. AWS Organizations es un servicio que le permite agrupar y administrar de manera centralizada varias Cuentas de AWS que posea su empresa. Si habilita todas las características en una empresa, entonces podrá aplicar políticas de control de servicio (SCP) a una o todas sus cuentas. Una SCP limita los permisos para las entidades de las cuentas de miembros, incluido cada Usuario raíz de la cuenta de AWS. Para obtener más información acerca de Organizations y las SCP, consulte Funcionamiento de las SCP en la Guía del usuario de AWS Organizations.
- Políticas de sesión: las políticas de sesión son políticas avanzadas que se pasan como parámetro cuando se crea una sesión temporal mediante programación para un rol o un usuario federado. Los permisos de la sesión resultantes son la intersección de las políticas basadas en identidades del rol y las políticas de la sesión. Los permisos también pueden proceder de una política en función de recursos. Una denegación explícita en cualquiera de estas políticas anulará el permiso. Para más información, consulte Políticas de sesión en la Guía del usuario de IAM.

Varios tipos de políticas

Cuando se aplican varios tipos de políticas a una solicitud, los permisos resultantes son más complicados de entender. Para obtener información acerca de cómo AWS decide si permitir o no una

solicitud cuando hay varios tipos de políticas implicados, consulte Lógica de evaluación de políticas en la Guía del usuario de IAM.

Cómo funciona Amazon CloudWatch con IAM

Antes de utilizar IAM para administrar el acceso a CloudWatch, obtenga información sobre qué características de IAM se pueden utilizar con CloudWatch.

Características de IAM que puede utilizar con Amazon CloudWatch

Característica de IAM	Compatibilidad con CloudWatch
Políticas basadas en identidades	Sí
Políticas basadas en recursos	No
Acciones de políticas	Sí
Recursos de políticas	Sí
Claves de condición de política (específicas del servicio)	Sí
ACL	No
ABAC (etiquetas en políticas)	Parcial
Credenciales temporales	Sí
Permisos de entidades principales	Sí
Roles de servicio	Sí
Roles vinculados al servicio	No

Para obtener una perspectiva general sobre cómo funcionan CloudWatch y otros servicios de AWS con la mayoría de características de IAM, consulte Servicios de AWS que funcionan con IAM en la Guía del usuario de IAM.

Políticas basadas en identidades para CloudWatch

Compatibilidad con las políticas basadas en identidad	Sí

Las políticas basadas en identidad son documentos de políticas de permisos JSON que puede asociar a una identidad, como un usuario de IAM, un grupo de usuarios o un rol. Estas políticas controlan qué acciones pueden realizar los usuarios y los roles, en qué recursos y en qué condiciones. Para obtener más información sobre cómo crear una política basada en identidad, consulte Creación de políticas de IAM en la Guía del usuario de IAM.

Con las políticas basadas en identidades de IAM, puede especificar las acciones y los recursos permitidos o denegados, así como las condiciones en las que se permiten o deniegan las acciones. No es posible especificar la entidad principal en una política basada en identidad porque se aplica al usuario o rol al que está adjunto. Para más información sobre los elementos que puede utilizar en una política de JSON, consulte Referencia de los elementos de las políticas de JSON de IAM en la Guía del usuario de IAM.

Ejemplos de políticas basadas en identidades para CloudWatch

Para ver ejemplos de políticas basadas en identidad de CloudWatch, consulte Ejemplos de políticas basadas en identidades para Amazon CloudWatch.

Políticas basadas en recursos dentro de CloudWatch

Compatibilidad con las políticas basadas en recursos	No

Las políticas basadas en recursos son documentos de política JSON que se asocian a un recurso. Ejemplos de políticas basadas en recursos son las políticas de confianza de roles de IAM y las políticas de bucket de Amazon S3. En los servicios que admiten políticas basadas en recursos, los administradores de servicios pueden utilizarlos para controlar el acceso a un recurso específico. Para el recurso al que se asocia la política, la política define qué acciones puede realizar una entidad principal especificada en ese recurso y en qué condiciones. Debe especificar una entidad principal en una política en función de recursos. Las entidades principales pueden incluir cuentas, usuarios, roles, usuarios federados o Servicios de AWS.

Para habilitar el acceso entre cuentas, puede especificar toda una cuenta o entidades de IAM de otra cuenta como la entidad principal de una política en función de recursos. Añadir a una política en función de recursos una entidad principal entre cuentas es solo una parte del establecimiento de una relación de confianza. Cuando la entidad principal y el recurso se encuentran en Cuentas de AWS diferentes, un administrador de IAM de la cuenta de confianza también debe conceder a la entidad principal (usuario o rol) permiso para acceder al recurso. Para conceder el permiso, adjunte la entidad a una política basada en identidad. Sin embargo, si la política en función de recursos concede el acceso a una entidad principal de la misma cuenta, no es necesaria una política basada en identidad adicional. Para más información, consulte Cómo los roles de IAM difieren de las políticas basadas en recursos en la Guía del usuario de IAM.

Acciones de política para CloudWatch

Admite acciones de política	Sí

Los administradores pueden utilizar las políticas JSON de AWS para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones.

El elemento `Action` de una política JSON describe las acciones que puede utilizar para conceder o denegar el acceso en una política. Las acciones de la política generalmente tienen el mismo nombre que la operación de API de AWS asociada. Hay algunas excepciones, como acciones de solo permiso que no tienen una operación de API coincidente. También hay algunas operaciones que requieren varias acciones en una política. Estas acciones adicionales se denominan acciones dependientes.

Incluya acciones en una política para conceder permisos y así llevar a cabo la operación asociada.

Para ver una lista de acciones de CloudWatch, consulte Acciones definidas por Amazon CloudWatch en la Referencia de autorizaciones de servicio.

Las acciones de políticas de CloudWatch utilizan el siguiente prefijo antes de la acción:

```
cloudwatch
```

Para especificar varias acciones en una única instrucción, sepárelas con comas.

```
"Action": [
```

```
      "cloudwatch:action1",
      "cloudwatch:action2"
         ]
```

Para ver ejemplos de políticas basadas en identidad de CloudWatch, consulte Ejemplos de políticas basadas en identidades para Amazon CloudWatch.

Recursos de política para CloudWatch

Admite recursos de políticas	Sí

Los administradores pueden utilizar las políticas JSON de AWS para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones.

El elemento `Resource` de la política JSON especifica el objeto u objetos a los que se aplica la acción. Las instrucciones deben contener un elemento `Resource` o `NotResource`. Como práctica recomendada, especifique un recurso utilizando el Nombre de recurso de Amazon (ARN). Puede hacerlo para acciones que admitan un tipo de recurso específico, conocido como permisos de nivel de recurso.

Para las acciones que no admiten permisos de nivel de recurso, como las operaciones de descripción, utilice un carácter comodín (*) para indicar que la instrucción se aplica a todos los recursos.

```
"Resource": "*"
```

Para ver una lista de tipos de recursos de CloudWatch y los ARN, consulte Recursos definidos por Amazon CloudWatch en la Referencia de autorizaciones de servicio. Para obtener información sobre las acciones con las que puede especificar el ARN de cada recurso, consulte Acciones definidas por Amazon CloudWatch.

Para ver ejemplos de políticas basadas en identidad de CloudWatch, consulte Ejemplos de políticas basadas en identidades para Amazon CloudWatch.

Claves de condición de política para CloudWatch

Admite claves de condición de políticas específicas del servicio	Sí

Los administradores pueden utilizar las políticas JSON de AWS para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones.

El elemento `Condition` (o bloque de `Condition`) permite especificar condiciones en las que entra en vigor una instrucción. El elemento `Condition` es opcional. Puede crear expresiones condicionales que utilicen operadores de condición, tales como igual o menor que, para que la condición de la política coincida con los valores de la solicitud.

Si especifica varios elementos de `Condition` en una instrucción o varias claves en un único elemento de `Condition`, AWS las evalúa mediante una operación `AND` lógica. Si especifica varios valores para una única clave de condición, AWS evalúa la condición con una operación lógica `OR`. Se deben cumplir todas las condiciones antes de que se concedan los permisos de la instrucción.

También puede utilizar variables de marcador de posición al especificar condiciones. Por ejemplo, puede conceder un permiso de usuario de IAM para acceder a un recurso solo si está etiquetado con su nombre de usuario de IAM. Para más información, consulte Elementos de la política de IAM: variables y etiquetas en la Guía del usuario de IAM.

AWS admite claves de condición globales y claves de condición específicas del servicio. Para ver todas las claves de condición globales de AWS, consulte Claves de contexto de condición globales de AWSen la Guía del usuario de IAM.

Para ver una lista de las claves de condición de CloudWatch, consulte Claves de condición para Amazon CloudWatch en la Referencia de autorizaciones de servicio. Para obtener más información acerca de las acciones y los recursos con los que puede utilizar una clave de condición, consulte Acciones definidas por Amazon CloudWatch.

Para ver ejemplos de políticas basadas en identidad de CloudWatch, consulte Ejemplos de políticas basadas en identidades para Amazon CloudWatch.

ACL en CloudWatch

Admite las ACL	No

Las listas de control de acceso (ACL) controlan qué entidades principales (miembros de cuentas, usuarios o roles) tienen permisos para acceder a un recurso. Las ACL son similares a las políticas basadas en recursos, aunque no utilizan el formato de documento de políticas JSON.

ABAC con CloudWatch

Admite ABAC (etiquetas en las políticas)	Parcial

El control de acceso basado en atributos (ABAC) es una estrategia de autorización que define permisos en función de atributos. En AWS, estos atributos se denominan etiquetas. Puede adjuntar etiquetas a entidades de IAM (usuarios o roles) y a muchos recursos de AWS. El etiquetado de entidades y recursos es el primer paso de ABAC. A continuación, designa las políticas de ABAC para permitir operaciones cuando la etiqueta de la entidad principal coincida con la etiqueta del recurso al que se intenta acceder.

ABAC es útil en entornos que crecen con rapidez y ayuda en situaciones en las que la administración de las políticas resulta engorrosa.

Para controlar el acceso en función de etiquetas, debe proporcionar información de las etiquetas en el elemento de condición de una política utilizando las claves de condición `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` o `aws:TagKeys`.

Si un servicio admite las tres claves de condición para cada tipo de recurso, el valor es Sí para el servicio. Si un servicio admite las tres claves de condición solo para algunos tipos de recursos, el valor es Parcial.

Para obtener más información sobre ABAC, consulte ¿Qué es ABAC? en la Guía del usuario de IAM. Para ver un tutorial con los pasos para configurar ABAC, consulte Uso del control de acceso basado en atributos (ABAC) en la Guía del usuario de IAM.

Uso de credenciales temporales con CloudWatch

Compatible con el uso de credenciales temporales	Sí

Algunos Servicios de AWS no funcionan cuando inicia sesión con credenciales temporales. Para obtener información adicional, incluida la información sobre qué Servicios de AWS funcionan con

credenciales temporales, consulte Servicios de AWS que funcionan con IAM en la Guía del usuario de IAM.

Utiliza credenciales temporales si inicia sesión en la AWS Management Console con cualquier método, excepto un nombre de usuario y una contraseña. Por ejemplo, cuando accede a AWS utilizando el enlace de inicio de sesión único (SSO) de la empresa, ese proceso crea automáticamente credenciales temporales. También crea automáticamente credenciales temporales cuando inicia sesión en la consola como usuario y luego cambia de rol. Para más información sobre el cambio de roles, consulte Cambio a un rol (consola) en la Guía del usuario de IAM.

Puede crear credenciales temporales de forma manual mediante la AWS CLI o la API de AWS. A continuación, puede usar esas credenciales temporales para acceder a AWS. AWS recomienda generar credenciales temporales de forma dinámica en lugar de usar claves de acceso a largo plazo. Para más información, consulte Credenciales de seguridad temporales en IAM.

Permisos de entidades principales entre servicios para CloudWatch

Admite Forward access sessions (FAS)	Sí

Cuando utiliza un usuario o un rol de IAM para llevar a cabo acciones en AWS, se lo considera una entidad principal. Cuando utiliza algunos servicios, es posible que realice una acción que desencadene otra acción en un servicio diferente. FAS utiliza los permisos de la entidad principal para llamar a un Servicio de AWS, combinados con el Servicio de AWS solicitante para realizar solicitudes a servicios posteriores. Las solicitudes de FAS solo se realizan cuando un servicio recibe una solicitud que requiere interacciones con otros Servicios de AWS o recursos para completarse. En este caso, debe tener permisos para realizar ambas acciones. Para obtener información sobre las políticas a la hora de realizar solicitudes de FAS, consulte Reenviar sesiones de acceso.

Roles de servicio para CloudWatch

Compatible con roles de servicio	Sí

Un rol de servicio es un rol de IAM que asume un servicio para realizar acciones en su nombre. Un administrador de IAM puede crear, modificar y eliminar un rol de servicio desde IAM. Para obtener más información, consulte Creación de un rol para delegar permisos a un Servicio de AWS en la Guía del usuario de IAM.

> **Warning**
>
> Es posible que cambiar los permisos de un rol de servicio interrumpa la funcionalidad de CloudWatch. Edite los roles de servicio solo cuando CloudWatch proporcione orientación para hacerlo.

Ejemplos de políticas basadas en identidades para Amazon CloudWatch

De forma predeterminada, los usuarios y roles no tienen permiso para crear o modificar recursos de CloudWatch. Tampoco pueden realizar tareas mediante la AWS Management Console, la AWS Command Line Interface (AWS CLI) o la API de AWS. Un administrador de IAM puede crear políticas de IAM para conceder permisos a los usuarios para realizar acciones en los recursos que necesitan. A continuación, el administrador puede añadir las políticas de IAM a roles y los usuarios pueden asumirlos.

Para obtener información acerca de cómo crear una política basada en identidades de IAM mediante el uso de estos documentos de políticas JSON de ejemplo, consulte Creación de políticas de IAM en la Guía del usuario de IAM.

Para obtener más información sobre las acciones y los tipos de recursos definidos por CloudWatch, incluido el formato de los ARN para cada tipo de recurso, consulte Acciones, recursos y claves de condición de Amazon CloudWatch en la Referencia de autorizaciones de servicio.

Temas

Prácticas recomendadas sobre las políticas

Las políticas basadas en identidades determinan si alguien puede crear, eliminar o acceder a los recursos de CloudWatch de la cuenta. Estas acciones pueden generar costos adicionales para su Cuenta de AWS. Siga estas directrices y recomendaciones al crear o editar políticas basadas en identidades:

- Comience con las políticas administradas por AWSy continúe con los permisos de privilegio mínimo: a fin de comenzar a conceder permisos a los usuarios y las cargas de trabajo, utilice las políticas administradas por AWS, que conceden permisos para muchos casos de uso comunes.

Están disponibles en su Cuenta de AWS. Se recomienda definir políticas administradas por el cliente de AWS específicas para sus casos de uso a fin de reducir aún más los permisos. Con el fin de obtener más información, consulte las políticas administradas por AWS o las políticas administradas por AWS para funciones de trabajo en la Guía de usuario de IAM.

- Aplique permisos de privilegio mínimo: cuando establezca permisos con políticas de IAM, conceda solo los permisos necesarios para realizar una tarea. Para ello, debe definir las acciones que se pueden llevar a cabo en determinados recursos en condiciones específicas, también conocidos como permisos de privilegios mínimos. Con el fin de obtener más información sobre el uso de IAM para aplicar permisos, consulte Políticas y permisos en IAM en la Guía del usuario de IAM.
- Utilice condiciones en las políticas de IAM para restringir aún más el acceso: puede agregar una condición a sus políticas para limitar el acceso a las acciones y los recursos. Por ejemplo, puede escribir una condición de políticas para especificar que todas las solicitudes deben enviarse utilizando SSL. También puede usar condiciones para conceder acceso a acciones de servicios si se emplean a través de un Servicio de AWS determinado como, por ejemplo, AWS CloudFormation. Para obtener más información, consulte Elementos de la política de JSON de IAM: Condición en la Guía del usuario de IAM.
- Utilice el analizador de acceso de IAM para validar las políticas de IAM con el fin de garantizar la seguridad y funcionalidad de los permisos: el analizador de acceso de IAM valida políticas nuevas y existentes para que respeten el lenguaje (JSON) de las políticas de IAM y las prácticas recomendadas de IAM. El analizador de acceso de IAM proporciona más de 100 verificaciones de políticas y recomendaciones procesables para ayudar a crear políticas seguras y funcionales. Para más información, consulte Política de validación de Analizador de acceso de IAM en la Guía de usuario de IAM.
- Solicite la autenticación multifactor (MFA): si se encuentra en una situación en la que necesita usuarios raíz o de IAM en su Cuenta de AWS, active la MFA para mayor seguridad. Para solicitar la MFA cuando se invocan las operaciones de la API, agregue las condiciones de la MFA a sus políticas. Para más información, consulte Configuración del acceso a una API protegido por MFA en la Guía de usuario de IAM.

Para obtener más información sobre las prácticas recomendadas de IAM, consulte las Prácticas recomendadas de seguridad en IAM en la Guía del usuario de IAM.

Uso de la consola de CloudWatch

Para acceder a la consola de Amazon CloudWatch, debe tener un conjunto mínimo de permisos. Estos permisos deben permitirle registrar y consultar los detalles acerca de los recursos de

CloudWatch en su Cuenta de AWS. Si crea una política basada en identidades que sea más restrictiva que el mínimo de permisos necesarios, la consola no funcionará del modo esperado para las entidades (usuarios o roles) que tengan esa política.

No es necesario que conceda permisos mínimos para la consola a los usuarios que solo realizan llamadas a la AWS CLI o a la API de AWS. En su lugar, permite acceso únicamente a las acciones que coincidan con la operación de API que intentan realizar.

Para asegurarse de que los usuarios y los roles puedan seguir utilizando la consola de CloudWatch, asocie también el `ConsoleAccess` de CloudWatch o la política administrada `ReadOnly` AWS a las entidades. Para más información, consulte Adición de permisos a un usuario en la Guía del usuario de IAM:

Permisos necesarios para la consola CloudWatch

El conjunto completo de permisos necesarios para trabajar con la consola de CloudWatch se enumera a continuación. Estos permisos proporcionan acceso total de escritura y lectura a la consola de CloudWatch.

- application-autoscaling:DescribeScalingPolicies
- autoscaling:DescribeAutoScalingGroups
- autoscaling:DescribePolicies
- cloudtrail:DescribeTrails
- cloudwatch:DeleteAlarms
- cloudwatch:DescribeAlarmHistory
- cloudwatch:DescribeAlarms
- cloudwatch:GetMetricData
- cloudwatch:GetMetricStatistics
- cloudwatch:ListMetrics
- cloudwatch:PutMetricAlarm
- cloudwatch:PutMetricData
- ec2:DescribeInstances
- ec2:DescribeTags
- ec2:DescribeVolumes

- es:DescribeElasticsearchDomain
- es:ListDomainNames
- events:DeleteRule
- events:DescribeRule
- events:DisableRule
- events:EnableRule
- events:ListRules
- events:PutRule
- iam:AttachRolePolicy
- iam:CreateRole
- iam:GetPolicy
- iam:GetPolicyVersion
- iam:GetRole
- iam:ListAttachedRolePolicies
- iam:ListRoles
- kinesis:DescribeStream
- kinesis:ListStreams
- lambda:AddPermission
- lambda:CreateFunction
- lambda:GetFunctionConfiguration
- lambda:ListAliases
- lambda:ListFunctions
- lambda:ListVersionsByFunction
- lambda:RemovePermission
- logs:CancelExportTask
- logs:CreateExportTask
- logs:CreateLogGroup
- logs:CreateLogStream

- logs:DeleteLogGroup
- logs:DeleteLogStream
- logs:DeleteMetricFilter
- logs:DeleteRetentionPolicy
- logs:DeleteSubscriptionFilter
- logs:DescribeExportTasks
- logs:DescribeLogGroups
- logs:DescribeLogStreams
- logs:DescribeMetricFilters
- logs:DescribeQueries
- logs:DescribeSubscriptionFilters
- logs:FilterLogEvents
- logs:GetLogGroupFields
- logs:GetLogRecord
- logs:GetLogEvents
- logs:GetQueryResults
- logs:PutMetricFilter
- logs:PutRetentionPolicy
- logs:PutSubscriptionFilter
- logs:StartQuery
- logs:StopQuery
- logs:TestMetricFilter
- s3:CreateBucket
- s3:ListBucket
- sns:CreateTopic
- sns:GetTopicAttributes
- sns:ListSubscriptions
- sns:ListTopics
- sns:SetTopicAttributes

- sns:Subscribe
- sns:Unsubscribe
- sqs:GetQueueAttributes
- sqs:GetQueueUrl
- sqs:ListQueues
- sqs:SetQueueAttributes
- swf:CreateAction
- swf:DescribeAction
- swf:ListActionTemplates
- swf:RegisterAction
- swf:RegisterDomain
- swf:UpdateAction

Además, para ver el mapa de seguimiento de X-Ray, necesita `AWSXrayReadOnlyAccess`

Resolución de problemas de identidad y acceso de Amazon CloudWatch

Utilice la siguiente información para diagnosticar y solucionar los problemas comunes que pueden surgir al trabajar con CloudWatch e IAM.

Temas

No tengo autorización para realizar una acción en CloudWatch

Si recibe un error que indica que no tiene autorización para realizar una acción, las políticas se deben actualizar para permitirle realizar la acción.

En el siguiente ejemplo, el error se produce cuando el usuario de IAM `mateojackson` intenta utilizar la consola para consultar los detalles acerca de un recurso ficticio *`my-example-widget`*, pero no tiene los permisos ficticios `cloudwatch:`*`GetWidget`*.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 cloudwatch:GetWidget on resource: my-example-widget
```

En este caso, la política del usuario `mateojackson` debe actualizarse para permitir el acceso al recurso *`my-example-widget`* mediante la acción `cloudwatch:`*`GetWidget`*.

Si necesita ayuda, póngase en contacto con su administrador de AWS. El administrador es la persona que le proporcionó las credenciales de inicio de sesión.

No tengo autorización para realizar la operación iam:PassRole

Si recibe un error que indica que no tiene autorización para realizar la acción `iam:PassRole`, las políticas se deben actualizar a fin de permitirle pasar un rol a CloudWatch.

Algunos servicios de Servicios de AWS le permiten transferir un rol existente a dicho servicio en lugar de crear un nuevo rol de servicio o uno vinculado al servicio. Para ello, debe tener permisos para transferir el rol al servicio.

En el siguiente ejemplo, el error se produce cuando un usuario de IAM denominado `marymajor` intenta utilizar la consola para realizar una acción en CloudWatch. Sin embargo, la acción requiere que el servicio cuente con permisos que otorguen un rol de servicio. Mary no tiene permisos para transferir el rol al servicio.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

En este caso, las políticas de Mary se deben actualizar para permitirle realizar la acción `iam:PassRole`.

Si necesita ayuda, póngase en contacto con su administrador de AWS. El administrador es la persona que le proporcionó las credenciales de inicio de sesión.

Quiero permitir a personas externas a mi Cuenta de AWS el acceso a mis recursos de CloudWatch

Puede crear un rol que los usuarios de otras cuentas o las personas externas a la organización puedan utilizar para acceder a sus recursos. Puede especificar una persona de confianza para que asuma el rol. En el caso de los servicios que admitan las políticas basadas en recursos o las listas de control de acceso (ACL), puede utilizar dichas políticas para conceder a las personas acceso a sus recursos.

Para más información, consulte lo siguiente:

- Para obtener información acerca de si CloudWatch admite estas características, consulte Cómo funciona Amazon CloudWatch con IAM.
- Para obtener información acerca de cómo proporcionar acceso a los recursos de las Cuenta de AWS de su propiedad, consulte Proporcionar acceso a un usuario de IAM a otra Cuentas de AWS de la que es propietario en la Guía del usuario de IAM.
- Para obtener información acerca de cómo proporcionar acceso a tus recursos a Cuentas de AWS de terceros, consulte Proporcionar acceso a Cuentas de AWS que son propiedad de terceros en la Guía del usuario de IAM.
- Para obtener información sobre cómo proporcionar acceso mediante una federación de identidades, consulte Proporcionar acceso a usuarios autenticados externamente (identidad federada) en la Guía del usuario de IAM.
- Para obtener información sobre la diferencia entre los roles y las políticas basadas en recursos para el acceso entre cuentas, consulte Cómo los roles de IAM difieren de las políticas basadas en recursos en la Guía del usuario de IAM.

Actualización de permisos del panel de CloudWatch

El 1 de mayo de 2018, AWS cambió los permisos necesarios para obtener acceso a los paneles de CloudWatch. Ahora, el acceso a los paneles de la consola de CloudWatch requiere los permisos que se introdujeron en 2017 para admitir las operaciones de la API en paneles:

- cloudwatch:GetDashboard
- cloudwatch:ListDashboards
- cloudwatch:PutDashboard
- cloudwatch:DeleteDashboards

Para acceder a los paneles de CloudWatch, necesita uno de los siguientes:

- La política AdministratorAccess.
- La política CloudWatchFullAccess.
- Una política personalizada que incluya uno o varios de estos permisos específicos:
  - `cloudwatch:GetDashboard` y `cloudwatch:ListDashboards` para poder ver los paneles
  - `cloudwatch:PutDashboard` para poder crear o modificar paneles

- `cloudwatch:DeleteDashboards` para poder eliminar paneles

Para obtener más información sobre cómo se cambian los permisos de un usuario de IAM mediante políticas, consulte Cambio de los permisos de un usuario de IAM.

Para obtener más información acerca de los permisos de CloudWatch, consulte Referencia de permisos de Amazon CloudWatch.

Para obtener más información sobre las operaciones de la API en paneles, consulte PutDashboard en la referencia de la API de Amazon CloudWatch.

Políticas administradas (predefinidas) de AWS para CloudWatch

AWS aborda muchos casos de uso comunes dando políticas de IAM independientes creadas y administradas por AWS. Estas políticas administradas por AWS conceden los permisos necesarios para casos de uso comunes, lo que le evita tener que investigar los permisos necesarios. Para obtener más información, consulte Políticas administradas por AWS en la Guía del usuario de IAM.

Las siguientes políticas administradas de AWS, que se pueden adjuntar a los usuarios de la cuenta, son específicas de CloudWatch:

Temas

- CloudWatchFullAccessV2
- CloudWatchFullAccess
- CloudWatchReadOnlyAccess
- CloudWatchActionsEC2Access
- CloudWatchAutomaticDashboardsAccess
- CloudWatchAgentServerPolicy
- CloudWatchAgentAdminPolicy
- Políticas administradas (predefinidas) de AWS para la observabilidad entre cuentas de CloudWatch
- Políticas administradas (predefinidas) de AWS para CloudWatch Synthetics
- Políticas administradas (predefinidas) de AWS para Amazon CloudWatch RUM
- Políticas administradas (predefinidas) de AWS para CloudWatch Evidently
- Política administrada de AWS para Systems Manager Incident Manager de AWS

CloudWatchFullAccessV2

AWS recientemente agregó la política de IAM administrada de CloudWatchFullAccessV2. Esta política otorga acceso total a las acciones y los recursos de CloudWatch y, al mismo tiempo, delimita más adecuadamente los permisos concedidos a otros servicios, como Amazon SNS y Amazon EC2 Auto Scaling. Le recomendamos que comience a usar esta política en lugar de CloudWatchFullAccess.AWS planea dejar obsoleto CloudWatchFullAccess en un futuro próximo.

Incluye permisos de `application-signals:` para que los usuarios puedan acceder a todas las funciones desde la consola de CloudWatch en Application Signals. Incluye algunos permisos de `autoscaling:Describe` para que los usuarios con esta política puedan ver las acciones de escalado automático asociadas a las alarmas de CloudWatch. Incluye algunos permisos de `sns` para que los usuarios con esta política puedan recuperar temas de Amazon SNS creados y asociarlos a las alarmas de CloudWatch. Incluye permisos de IAM para que los usuarios con esta política puedan ver información sobre las funciones vinculadas a servicios asociados a CloudWatch. Incluye los permisos de `oam:ListSinks` y `oam:ListAttachedLinks` para que los usuarios con esta política puedan usar la consola para ver los datos compartidos desde las cuentas de origen en la observabilidad entre cuentas de CloudWatch.

Incluye `rum`, `synthetics` y permisos de `xray` para que los usuarios puedan tener acceso completo a CloudWatch Synthetics, AWS X-Ray y CloudWatch RUM, todos ellos bajo el servicio de CloudWatch.

Los contenidos de CloudWatchFullAccessV2 son los siguientes:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "CloudWatchFullAccessPermissions",
            "Effect": "Allow",
            "Action": [
                "application-autoscaling:DescribeScalingPolicies",
                "application-signals:*",
                "autoscaling:DescribeAutoScalingGroups",
                "autoscaling:DescribePolicies",
                "cloudwatch:*",
                "logs:*",
                "sns:CreateTopic",
                "sns:ListSubscriptions",
                "sns:ListSubscriptionsByTopic",
```

```
                "sns:ListTopics",
                "sns:Subscribe",
                "iam:GetPolicy",
                "iam:GetPolicyVersion",
                "iam:GetRole",
                "oam:ListSinks",
                "rum:*",
                "synthetics:*",
                "xray:*"
            ],
            "Resource": "*"
        },
        {
            "Sid": "CloudWatchApplicationSignalsServiceLinkedRolePermissions",
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "arn:aws:iam::*:role/aws-service-role/application-
signals.cloudwatch.amazonaws.com/AWSServiceRoleForCloudWatchApplicationSignals",
            "Condition": {
                "StringLike": {
                    "iam:AWSServiceName": "application-
signals.cloudwatch.amazonaws.com"
                }
            }
        },
        {
            "Sid": "EventsServicePermissions",
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "arn:aws:iam::*:role/aws-service-role/events.amazonaws.com/
AWSServiceRoleForCloudWatchEvents*",
            "Condition": {
                "StringLike": {
                    "iam:AWSServiceName": "events.amazonaws.com"
                }
            }
        },
         {
            "Sid": "OAMReadPermissions",
            "Effect": "Allow",
            "Action": [
                "oam:ListAttachedLinks"
            ],
            "Resource": "arn:aws:oam:*:*:sink/*"
```

```
        }
    ]
}
```

CloudWatchFullAccess

La política CloudWatchFullAccess está en vías de quedar obsoleta. Le recomendamos que deje de usarla y utilice CloudWatchFullAccessV2 en su lugar.

Los contenidos de CloudWatchFullAccess son los siguientes:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "autoscaling:Describe*",
                "cloudwatch:*",
                "logs:*",
                "sns:*",
                "iam:GetPolicy",
                "iam:GetPolicyVersion",
                "iam:GetRole",
                "oam:ListSinks"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "arn:aws:iam::*:role/aws-service-role/events.amazonaws.com/
AWSServiceRoleForCloudWatchEvents*",
            "Condition": {
                "StringLike": {
                    "iam:AWSServiceName": "events.amazonaws.com"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "oam:ListAttachedLinks"
```

```
            ],
            "Resource": "arn:aws:oam:*:*:sink/*"
        }
    ]
}
```

CloudWatchReadOnlyAccess

La política CloudWatchReadOnlyAccess concede acceso de solo lectura a CloudWatch.

La política incluye algunos permisos de `logs:`, para que los usuarios con esta política puedan usar la consola para ver la información de los Registros de CloudWatch y las consultas de Información de registros de CloudWatch. Incluye `autoscaling:Describe*`, para que los usuarios con esta política puedan ver las acciones de escalado automático asociadas a las alarmas de CloudWatch. Incluye los permisos de `application-signals:` para que los usuarios puedan usar Application Signals para supervisar el estado de sus servicios. Incluye `application-autoscaling:DescribeScalingPolicies`, para que los usuarios con esta política puedan acceder a la información sobre las políticas de escalado automático de la aplicación. Incluye `sns:Get*` y `sns:List*`, para que los usuarios con esta política puedan recuperar información sobre los temas de Amazon SNS que reciben notificaciones sobre las alarmas de CloudWatch. Incluye los permisos de `oam:ListSinks` y `oam:ListAttachedLinks`, para que los usuarios con esta política puedan usar la consola para ver los datos compartidos desde las cuentas de origen en la observabilidad entre cuentas de CloudWatch. Incluye los permisos de `iam:GetRole` para que los usuarios puedan comprobar si se ha configurado CloudWatch Application Signals.

Incluye `rum`, `synthetics` y permisos de `xray` para que los usuarios puedan tener acceso solo de lectura a CloudWatch Synthetics, AWS X-Ray y CloudWatch RUM, todos ellos bajo el servicio de CloudWatch.

A continuación se detalla el contenido de la política CloudWatchReadOnlyAccess:

```
{
"Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "CloudWatchReadOnlyAccessPermissions",
            "Effect": "Allow",
            "Action": [
                "application-autoscaling:DescribeScalingPolicies",
                "application-signals:BatchGet*",
                "application-signals:Get*",
```

```
                "application-signals:List*",
                "autoscaling:Describe*",
                "cloudwatch:BatchGet*",
                "cloudwatch:Describe*",
                "cloudwatch:GenerateQuery",
                "cloudwatch:Get*",
                "cloudwatch:List*",
                "logs:Get*",
                "logs:List*",
                "logs:StartQuery",
                "logs:StopQuery",
                "logs:Describe*",
                "logs:TestMetricFilter",
                "logs:FilterLogEvents",
                "logs:StartLiveTail",
                "logs:StopLiveTail",
                "oam:ListSinks",
                "sns:Get*",
                "sns:List*",
                "rum:BatchGet*",
                "rum:Get*",
                "rum:List*",
                "synthetics:Describe*",
                "synthetics:Get*",
                "synthetics:List*",
                "xray:BatchGet*",
                "xray:Get*"
            ],
            "Resource": "*"
        },
        {
            "Sid": "OAMReadPermissions",
            "Effect": "Allow",
            "Action": [
                "oam:ListAttachedLinks"
            ],
            "Resource": "arn:aws:oam:*:*:sink/*"
        },
        {
            "Sid": "CloudWatchReadOnlyGetRolePermissions",
            "Effect": "Allow",
            "Action": "iam:GetRole",
            "Resource": "arn:aws:iam::*:role/aws-service-role/application-
signals.cloudwatch.amazonaws.com/AWSServiceRoleForCloudWatchApplicationSignals"
```

```
        }
     ]
}
```

CloudWatchActionsEC2Access

La política CloudWatchActionsEC2Access concede acceso de solo lectura a alarmas y métricas de CloudWatch, además de a los metadatos de Amazon EC2. Concede acceso a las acciones de la API de detener, terminar y reiniciar para instancias EC2.

A continuación se detalla el contenido de la política CloudWatchActionsEC2Access:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "cloudwatch:Describe*",
        "ec2:Describe*",
        "ec2:RebootInstances",
        "ec2:StopInstances",
        "ec2:TerminateInstances"
      ],
      "Resource": "*"
    }
  ]
}
```

CloudWatchAutomaticDashboardsAccess

El rol de IAM CloudWatch-CrossAccountSharingRole utiliza la política administrada CloudWatch-CrossAccountAccess. Este rol y política permiten a los usuarios de paneles de cuentas cruzadas visualizar paneles automáticos en cada cuenta que comparta paneles.

A continuación se detalla el contenido de CloudWatchAutomaticDashboardsAccess:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
```

```
        "autoscaling:DescribeAutoScalingGroups",
        "cloudfront:GetDistribution",
        "cloudfront:ListDistributions",
        "dynamodb:DescribeTable",
        "dynamodb:ListTables",
        "ec2:DescribeInstances",
        "ec2:DescribeVolumes",
        "ecs:DescribeClusters",
        "ecs:DescribeContainerInstances",
        "ecs:ListClusters",
        "ecs:ListContainerInstances",
        "ecs:ListServices",
        "elasticache:DescribeCacheClusters",
        "elasticbeanstalk:DescribeEnvironments",
        "elasticfilesystem:DescribeFileSystems",
        "elasticloadbalancing:DescribeLoadBalancers",
        "kinesis:DescribeStream",
        "kinesis:ListStreams",
        "lambda:GetFunction",
        "lambda:ListFunctions",
        "rds:DescribeDBClusters",
        "rds:DescribeDBInstances",
        "resource-groups:ListGroupResources",
        "resource-groups:ListGroups",
        "route53:GetHealthCheck",
        "route53:ListHealthChecks",
        "s3:ListAllMyBuckets",
        "s3:ListBucket",
        "sns:ListTopics",
        "sqs:GetQueueAttributes",
        "sqs:GetQueueUrl",
        "sqs:ListQueues",
        "synthetics:DescribeCanariesLastRun",
        "tag:GetResources"
      ],
      "Effect": "Allow",
      "Resource": "*"
    },
    {
      "Action": [
        "apigateway:GET"
      ],
      "Effect": "Allow",
      "Resource": [
```

```
        "arn:aws:apigateway:*::/restapis*"
      ]
    }
  ]
```

CloudWatchAgentServerPolicy

La política CloudWatchAgentServerPolicy se puede utilizar en roles de IAM adjuntados a instancias de Amazon EC2 a fin de permitir que el agente de CloudWatch lea información de la instancia y la registre en CloudWatch. El contenido es el siguiente.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "CWACloudWatchServerPermissions",
            "Effect": "Allow",
            "Action": [
                "cloudwatch:PutMetricData",
                "ec2:DescribeVolumes",
                "ec2:DescribeTags",
                "logs:PutLogEvents",
                "logs:PutRetentionPolicy",
                "logs:DescribeLogStreams",
                "logs:DescribeLogGroups",
                "logs:CreateLogStream",
                "logs:CreateLogGroup",
                "xray:PutTraceSegments",
                "xray:PutTelemetryRecords",
                "xray:GetSamplingRules",
                "xray:GetSamplingTargets",
                "xray:GetSamplingStatisticSummaries"
            ],
            "Resource": "*"
        },
        {
            "Sid": "CWASSMServerPermissions",
            "Effect": "Allow",
            "Action": [
                "ssm:GetParameter"
            ],
            "Resource": "arn:aws:ssm:*:*:parameter/AmazonCloudWatch-*"
        }
```

```
    ]
}
```

CloudWatchAgentAdminPolicy

La política CloudWatchAgentAdminPolicy se puede utilizar en funciones de IAM adjuntadas a instancias de Amazon EC2. Esta política le permite al agente de CloudWatch leer información de la instancia y registrarla en CloudWatch, así como registrar información en el almacén de parámetros. El contenido es el siguiente.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "CWACloudWatchPermissions",
            "Effect": "Allow",
            "Action": [
                "cloudwatch:PutMetricData",
                "ec2:DescribeTags",
                "logs:PutLogEvents",
                "logs:PutRetentionPolicy",
                "logs:DescribeLogStreams",
                "logs:DescribeLogGroups",
                "logs:CreateLogStream",
                "logs:CreateLogGroup",
                "xray:PutTraceSegments",
                "xray:PutTelemetryRecords",
                "xray:GetSamplingRules",
                "xray:GetSamplingTargets",
                "xray:GetSamplingStatisticSummaries"
            ],
            "Resource": "*"
        },
        {
            "Sid": "CWASSMPermissions",
            "Effect": "Allow",
            "Action": [
                "ssm:GetParameter",
                "ssm:PutParameter"
            ],
            "Resource": "arn:aws:ssm:*:*:parameter/AmazonCloudWatch-*"
        }
    ]
```

```
}
```

> **Note**
>
> Para consultar estas políticas de permisos, inicie sesión en la consola de IAM y busque las políticas específicas.

También puede crear sus propias políticas de IAM personalizadas para conceder permisos a las acciones y recursos de CloudWatch. Puede asociar estas políticas personalizadas a los grupos o usuarios de IAM que requieran esos permisos.

Políticas administradas (predefinidas) de AWS para la observabilidad entre cuentas de CloudWatch

Las políticas de esta sección otorgan permisos relacionados con la observabilidad entre cuentas de CloudWatch. Para obtener más información, consulte Observabilidad entre cuentas de CloudWatch.

CloudWatchCrossAccountSharingConfiguration

La política de CloudWatchCrossAccountSharingConfiguration permite crear, administrar y ver los enlaces de Observability Access Manager para compartir los recursos de CloudWatch entre cuentas. Para obtener más información, consulte Observabilidad entre cuentas de CloudWatch. El contenido es el siguiente:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "cloudwatch:Link",
                "oam:ListLinks"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "oam:DeleteLink",
                "oam:GetLink",
```

```
                "oam:TagResource"
            ],
            "Resource": "arn:aws:oam:*:*:link/*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "oam:CreateLink",
                "oam:UpdateLink"
            ],
            "Resource": [
                "arn:aws:oam:*:*:link/*",
                "arn:aws:oam:*:*:sink/*"
            ]
        }
    ]
}
```

OAMFullAccess

La política OAMFullAccess otorga acceso para crear, administrar y ver los sumideros y enlaces de Observability Access Manager, que se utilizan para la observabilidad entre cuentas de CloudWatch.

La política OAMFullAccess por sí sola no permite compartir datos de observabilidad entre enlaces. Para crear un enlace para compartir las métricas de CloudWatch, también necesita CloudWatchFullAccess o CloudWatchCrossAccountSharingConfiguration. Para crear un enlace para compartir los grupos de registro de CloudWatch Logs, también necesita CloudWatchLogsFullAccess o CloudWatchLogsCrossAccountSharingConfiguration. Para crear un enlace para compartir los seguimientos de X-Ray, también necesita AWSXRayFullAccess o AWSXRayCrossAccountSharingConfiguration.

Para obtener más información, consulte Observabilidad entre cuentas de CloudWatch. El contenido es el siguiente:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "oam:*"
            ],
```

```
            "Resource": "*"
        }
    ]
}
```

OAMReadOnlyAccess

La política OAMReadOnlyAccess otorga acceso de solo lectura a los recursos de Observability Access Manager, que se utilizan para la observabilidad entre cuentas de CloudWatch. Para obtener más información, consulte Observabilidad entre cuentas de CloudWatch. El contenido es el siguiente:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "oam:Get*",
                "oam:List*"
            ],
            "Resource": "*"
        }
    ]
}
```

Políticas administradas (predefinidas) de AWS para CloudWatch Synthetics

Las políticas administradas de AWS CloudWatchSyntheticsFullAccess y CloudWatchSyntheticsReadOnlyAccess están disponibles para asignarlas a usuarios que administrarán o usarán CloudWatch Synthetics. También son relevantes las siguientes políticas adicionales:

- AmazonS3ReadOnlyAccess y CloudWatchReadOnlyAccess: estas son necesarias para poder leer todos los datos de Synthetics en la consola de CloudWatch.
- AWSLambdaReadOnlyAccess: para poder ver el código fuente que utilizan los canaries.
- CloudWatchSyntheticsFullAccess le permite crear un valor controlado; además, para crear un valor controlado que tendrá un rol de IAM nuevo creado para este, también necesita la siguiente declaración de política anexa:

```
{
```

```
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iam:CreateRole",
                "iam:DeleteRole",
                "iam:CreatePolicy",
                "iam:DeletePolicy",
                "iam:AttachRolePolicy",
                "iam:DetachRolePolicy",
            ],
            "Resource": [
                "arn:aws:iam::*:role/service-role/CloudWatchSyntheticsRole*",
                "arn:aws:iam::*:policy/service-role/CloudWatchSyntheticsPolicy*"
            ]
        }
    ]
}
```

Important

Conceder a un usuario los permisos `iam:CreateRole`, `iam:DeleteRole`, `iam:CreatePolicy`, `iam:DeletePolicy`, `iam:AttachRolePolicy` y `iam:DetachRolePolicy` proporciona a ese usuario acceso administrativo completo para crear, adjuntar y eliminar roles y políticas que tienen ARN que hagan coincidir `arn:aws:iam::*:role/service-role/CloudWatchSyntheticsRole*` y `arn:aws:iam::*:policy/service-role/CloudWatchSyntheticsPolicy*`. Por ejemplo, un usuario con estos permisos puede crear una política que tenga permisos completos para todos los recursos y asociarla a cualquier rol que coincida con ese patrón de ARN. Sea muy cauteloso en lo referente a la persona a la que concede estos permisos.

Para obtener información acerca de cómo asociar políticas y conceder permisos a los usuarios, consulte Cambio de los permisos de un usuario de IAM y Para integrar una política en línea de un usuario o un rol.

CloudWatchSyntheticsFullAccess

El contenido de la política CloudWatchSyntheticsFullAccess es el siguiente:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "synthetics:*"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:CreateBucket",
                "s3:PutEncryptionConfiguration"
            ],
            "Resource": [
                "arn:aws:s3:::cw-syn-results-*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:ListRoles",
                "s3:ListAllMyBuckets",
                "xray:GetTraceSummaries",
                "xray:BatchGetTraces",
                "apigateway:GET"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetBucketLocation"
            ],
            "Resource": "arn:aws:s3:::*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:ListBucket"
```

```
            ],
            "Resource": "arn:aws:s3:::cw-syn-*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObjectVersion"
            ],
            "Resource": "arn:aws:s3:::aws-synthetics-library-*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:PassRole"
            ],
            "Resource": [
                "arn:aws:iam::*:role/service-role/CloudWatchSyntheticsRole*"
            ],
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": [
                        "lambda.amazonaws.com",
                        "synthetics.amazonaws.com"
                    ]
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:GetRole",
                "iam:ListAttachedRolePolicies"
            ],
            "Resource": [
                "arn:aws:iam::*:role/service-role/CloudWatchSyntheticsRole*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "cloudwatch:GetMetricData",
                "cloudwatch:GetMetricStatistics"
            ],
            "Resource": "*"
```

```
        },
        {
            "Effect": "Allow",
            "Action": [
                "cloudwatch:PutMetricAlarm",
                "cloudwatch:DeleteAlarms"
            ],
            "Resource": [
                "arn:aws:cloudwatch:*:*:alarm:Synthetics-*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "cloudwatch:DescribeAlarms"
            ],
            "Resource": [
                "arn:aws:cloudwatch:*:*:alarm:*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "lambda:CreateFunction",
                "lambda:AddPermission",
                "lambda:PublishVersion",
                "lambda:UpdateFunctionCode",
                "lambda:UpdateFunctionConfiguration",
                "lambda:GetFunctionConfiguration",
                "lambda:DeleteFunction"
            ],
            "Resource": [
                "arn:aws:lambda:*:*:function:cwsyn-*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "lambda:GetLayerVersion",
                "lambda:PublishLayerVersion",
                "lambda:DeleteLayerVersion"
            ],
            "Resource": [
                "arn:aws:lambda:*:*:layer:cwsyn-*",
```

```
                "arn:aws:lambda:*:*:layer:Synthetics:*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeVpcs",
                "ec2:DescribeSubnets",
                "ec2:DescribeSecurityGroups"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "sns:ListTopics"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "sns:CreateTopic",
                "sns:Subscribe",
                "sns:ListSubscriptionsByTopic"
            ],
            "Resource": [
                "arn:*:sns:*:*:Synthetics-*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:ListAliases"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
```

```
                "kms:DescribeKey"
            ],
            "Resource": "arn:aws:kms:*:*:key/*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt"
            ],
            "Resource": "arn:aws:kms:*:*:key/*",
            "Condition": {
                "StringLike": {
                    "kms:ViaService": [
                        "s3.*.amazonaws.com"
                    ]
                }
            }
        }
    ]
}
```

CloudWatchSyntheticsReadOnlyAccess

El contenido de la política CloudWatchSyntheticsReadOnlyAccess es el siguiente:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "synthetics:Describe*",
                "synthetics:Get*",
                "synthetics:List*",
                "lambda:GetFunctionConfiguration"
            ],
            "Resource": "*"
        }
    ]
}
```

Políticas administradas (predefinidas) de AWS para Amazon CloudWatch RUM

Las políticas administradas AmazonCloudWatchRUMFullAccess (Acceso completo a Amazon CloudWatch RUM) y AmazonCloudWatchRUMReadOnlyAccess (Acceso de solo lectura a Amazon CloudWatch RUM) de AWS están disponibles para que las asigne a los usuarios que administrarán o usarán CloudWatch RUM.

AmazonCloudWatchRUMFullAccess

A continuación se detalla el contenido de la política AmazonCloudWatchRUMFullAccess.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "rum:*"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:GetRole",
                "iam:CreateServiceLinkedRole"
            ],
            "Resource": [
                "arn:aws:iam::*:role/aws-service-role/rum.amazonaws.com/
AWSServiceRoleForRealUserMonitoring"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:PassRole"
            ],
            "Resource": [
                "arn:aws:iam::*:role/RUM-Monitor*"
            ],
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": [
```

```
                    "cognito-identity.amazonaws.com"
                ]
            }
        }
    },
    {
        "Effect": "Allow",
        "Action": [
            "cloudwatch:GetMetricData",
            "cloudwatch:GetMetricStatistics",
            "cloudwatch:ListMetrics"
        ],
        "Resource": "*"
    },
    {
        "Effect": "Allow",
        "Action": [
            "cloudwatch:DescribeAlarms"
        ],
        "Resource": "arn:aws:cloudwatch:*:*:alarm:*"
    },
    {
        "Effect": "Allow",
        "Action": [
            "cognito-identity:CreateIdentityPool",
            "cognito-identity:ListIdentityPools",
            "cognito-identity:DescribeIdentityPool",
            "cognito-identity:GetIdentityPoolRoles",
            "cognito-identity:SetIdentityPoolRoles"
        ],
        "Resource": "arn:aws:cognito-identity:*:*:identitypool/*"
    },
    {
        "Effect": "Allow",
        "Action": [
            "logs:CreateLogGroup",
            "logs:DeleteLogGroup",
            "logs:PutRetentionPolicy",
            "logs:CreateLogStream"
        ],
        "Resource": "arn:aws:logs:*:*:log-group:*RUMService*"
    },
    {
        "Effect": "Allow",
```

```
            "Action": [
                "logs:CreateLogDelivery",
                "logs:GetLogDelivery",
                "logs:UpdateLogDelivery",
                "logs:DeleteLogDelivery",
                "logs:ListLogDeliveries",
                "logs:DescribeResourcePolicies"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "logs:DescribeLogGroups"
            ],
            "Resource": "arn:aws:logs:*:*:log-group::log-stream:*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "synthetics:describeCanaries",
                "synthetics:describeCanariesLastRun"
            ],
            "Resource": "arn:aws:synthetics:*:*:canary:*"
        }
    ]
}
```

AmazonCloudWatchRUMReadOnlyAccess

A continuación se detalla el contenido de la política AmazonCloudWatchRUMReadOnlyAccess.

```
{

 "Version": "2012-10-17",
 "Statement": [
  {
   "Effect": "Allow",
   "Action": [
    "rum:GetAppMonitor",
    "rum:GetAppMonitorData",
    "rum:ListAppMonitors",
    "rum:ListRumMetricsDestinations",
    "rum:BatchGetRumMetricDefinitions"
```

```
    ],
    "Resource": "*"
   }
  ]
}
```

AmazonCloudWatchRUMServiceRolePolicy

No puede adjuntar AmazonCloudWatchRUMServiceRolePolicy (Política de roles de servicio de Amazon Cloud Watch RUM) a sus entidades de IAM. Esta política se adjunta a un rol vinculado a servicios que permite que CloudWatch RUM publique datos de supervisión en otros servicios relevantes de AWS. Para obtener más información sobre este rol vinculado a servicios, consulte Uso de roles vinculados a servicios para CloudWatch RUM.

A continuación se detalla el contenido completo de la política AmazonCloudWatchRUMServiceRolePolicy.

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Effect": "Allow",
   "Action": [
    "xray:PutTraceSegments"
   ],
   "Resource": [
    "*"
   ]
  },
  {
   "Effect": "Allow",
   "Action": "cloudwatch:PutMetricData",
   "Resource": "*",
   "Condition": {
    "StringLike": {
     "cloudwatch:namespace": [
      "RUM/CustomMetrics/*",
      "AWS/RUM"
     ]
    }
   }
  }
 ]
```

```
}
```

Políticas administradas (predefinidas) de AWS para CloudWatch Evidently

Las políticas administradas CloudWatchEvidentlyFullAccess y CloudWatchEvidentlyReadOnlyAccess de AWS están disponibles para asignarlas a usuarios que administrarán o usarán CloudWatch Evidently.

CloudWatchEvidentlyFullAccess

A continuación se detalla el contenido de la política CloudWatchEvidentlyFullAccess.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "evidently:*"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:ListRoles"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:GetRole"
            ],
            "Resource": [
                "arn:aws:iam::*:role/service-role/CloudWatchRUMEvidentlyRole-*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetBucketLocation",
                "s3:ListAllMyBuckets"
```

```
            ],
            "Resource": "arn:aws:s3:::*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "cloudwatch:GetMetricData",
                "cloudwatch:GetMetricStatistics",
                "cloudwatch:DescribeAlarmHistory",
                "cloudwatch:DescribeAlarmsForMetric",
                "cloudwatch:ListTagsForResource"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "cloudwatch:DescribeAlarms",
                "cloudwatch:TagResource",
                "cloudwatch:UnTagResource"
            ],
            "Resource": [
                "arn:aws:cloudwatch:*:*:alarm:*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "cloudtrail:LookupEvents"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "cloudwatch:PutMetricAlarm"
            ],
            "Resource": [
                "arn:aws:cloudwatch:*:*:alarm:Evidently-Alarm-*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
```

```
                "sns:ListTopics"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "sns:CreateTopic",
                "sns:Subscribe",
                "sns:ListSubscriptionsByTopic"
            ],
            "Resource": [
                "arn:*:sns:*:*:Evidently-*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "logs:DescribeLogGroups"
            ],
            "Resource": [
                "*"
            ]
        }
    ]
}
```

CloudWatchEvidentlyReadOnlyAccess

A continuación se detalla el contenido de la política CloudWatchEvidentlyReadOnlyAccess.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "evidently:GetExperiment",
                "evidently:GetFeature",
                "evidently:GetLaunch",
                "evidently:GetProject",
                "evidently:GetSegment",
```

```
                "evidently:ListExperiments",
                "evidently:ListFeatures",
                "evidently:ListLaunches",
                "evidently:ListProjects",
                "evidently:ListSegments",
                "evidently:ListSegmentReferencs"
            ],
            "Resource": "*"
        }
    ]
}
```

Política administrada de AWS para Systems Manager Incident Manager de AWS

La política AWSCloudWatchAlarms_ActionSSMIncidentsServiceRolePolicy está adjuntada a un rol vinculado a servicios que permite a CloudWatch comenzar incidentes en Systems Manager Incident Manager de AWS en su nombre. Para obtener más información, consulte Permisos de roles vinculados a servicios para acciones de alarmas de CloudWatch Systems Manager Incident Manager.

La política cuenta con el siguiente permiso:

- ssm-incidents:StartIncident

Ejemplos de políticas administradas por el cliente

En esta sección, encontrará ejemplos de políticas de usuario que conceden permisos para diversas acciones de CloudWatch. Estas políticas funcionan cuando se utiliza la API de CloudWatch, los SDK de AWS o la AWS CLI.

Ejemplos

- Ejemplo 1: Permitir al usuario acceso completo a CloudWatch
- Ejemplo 2: Permitir acceso de solo lectura a CloudWatch
- Ejemplo 3: Detener o terminar una instancia de Amazon EC2

Ejemplo 1: Permitir al usuario acceso completo a CloudWatch

Para conceder a un usuario acceso completo a CloudWatch, puede otorgarle la política administrada CloudWatchFullAccess en lugar de crear una política administrada por el cliente. Los contenidos de CloudWatchFullAccess se describen en CloudWatchFullAccess.

Ejemplo 2: Permitir acceso de solo lectura a CloudWatch

La siguiente política le permite a un usuario acceso de solo lectura a CloudWatch y le permite ver las acciones de Amazon EC2 Auto Scaling, las métricas de CloudWatch, los datos de CloudWatch Logs y los datos de Amazon SNS relacionados con alarmas.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "autoscaling:Describe*",
        "cloudwatch:Describe*",
        "cloudwatch:Get*",
        "cloudwatch:List*",
        "logs:Get*",
        "logs:Describe*",
        "logs:StartQuery",
        "logs:StopQuery",
        "logs:TestMetricFilter",
        "logs:FilterLogEvents",
        "logs:StartLiveTail",
        "logs:StopLiveTail",
        "sns:Get*",
        "sns:List*"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

Ejemplo 3: Detener o terminar una instancia de Amazon EC2

La siguiente política le permite a una acción de alarma de CloudWatch detener o terminar una instancia EC2. En el siguiente ejemplo, las acciones GetMetricData, ListMetrics y DescribeAlarms

son opcionales. Se recomienda que incluya estas acciones para asegurarse de haber parado o finalizado la instancia correctamente.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "cloudwatch:PutMetricAlarm",
        "cloudwatch:GetMetricData",
        "cloudwatch:ListMetrics",
        "cloudwatch:DescribeAlarms"
      ],
      "Resource": [
        "*"
      ],
      "Effect": "Allow"
    },
    {
      "Action": [
        "ec2:DescribeInstanceStatus",
        "ec2:DescribeInstances",
        "ec2:StopInstances",
        "ec2:TerminateInstances"
      ],
      "Resource": [
        "*"
      ],
      "Effect": "Allow"
    }
  ]
}
```

Actualizaciones de CloudWatch para las políticas administradas de AWS

Es posible consultar los detalles sobre las actualizaciones de las políticas administradas de AWS para CloudWatch debido a que este servicio comenzó a realizar el seguimiento de estos cambios. Para obtener alertas automáticas sobre cambios en esta página, suscríbase a la fuente RSS en la página de historial de documentos de CloudWatch.

Cambio	Descripción	Fecha
CloudWatchFullAccessV2: actualizar a una política existente	CloudWatch actualizó la política denominada CloudWatchFullAccessV2. Se actualizó el alcance de la política de `CloudWatchFullAccessPermissions` para añadir `application-signals:*` para que los usuarios pudieran usar CloudWatch Application Signals para ver, investigar y diagnosticar problemas relacionados con el estado de sus servicios.	20 de mayo de 2024
CloudWatchReadOnlyAccess: actualizar a una política existente	CloudWatch actualizó la política denominada CloudWatchReadOnlyAccess. El alcance de la política de `CloudWatchReadOnlyAccessPermissions` se actualizó para añadir `application-signals:BatchGet*` , `application-signals:List*` y `application-signals:Get*` para que los usuarios puedan usar CloudWatch Application Signals para ver, investigar y diagnosticar problemas relacionados con el estado de sus servicios. Se actualizó	20 de mayo de 2024

Cambio	Descripción	Fecha
	el alcance de `CloudWatchReadOnlyGetRolePermissions` para añadir la acción `iam:GetRole`, de modo que los usuarios puedan comprobar si CloudWatch Application Signals está configurado.	
CloudWatchApplicationSignalsServiceRolePolicy: actualización a una política existente	CloudWatch actualizó la política denominada CloudWatchApplicationSignalsServiceRolePolicy. El alcance de los permisos `logs:StartQuery` y `logs:GetQueryResults` se modificó para agregar los ARN `arn:aws:logs:*:*:log-group:/aws/appsignals/*:*` y `arn:aws:logs:*:*:log-group:/aws/application-signals/data:*` y permitir Application Signals en más arquitecturas.	18 de abril de 2024

Cambio	Descripción	Fecha
CloudWatchApplicationSignalsServiceRolePolicy: actualización a una política existente	CloudWatch cambió el alcance de un permiso en CloudwatchApplicationSignalsServiceRolePolicy. El alcance del permiso `cloudwatch:GetMetricData` se modificó a * para que Application Signals pueda recuperar métricas de los orígenes de las cuentas vinculadas.	8 de abril de 2024
CloudWatchAgentServerPolicy: actualización a una política existente	CloudWatch agregó permisos a CloudWatchAgentServerPolicy. Los permisos `xray:PutTraceSegments`, `xray:PutTelemetryRecords`, `xray:GetSamplingRules`, `xray:GetSamplingTargets`, `xray:GetSamplingStatisticSummaries` y `logs:PutRetentionPolicy` se agregaron para que el agente de CloudWatch pueda publicar seguimientos de X-Ray y modificar los periodos de retención de los grupos de registros.	12 de febrero de 2024

Cambio	Descripción	Fecha
CloudWatchAgentAdminPolicy: actualización a una política existente	CloudWatch agregó permisos a CloudWatchAgentAdminPolicy. Los permisos `xray:PutTraceSegments` , `xray:PutTelemetryRecords` , `xray:GetSamplingRules` , `xray:GetSamplingTargets` , `xray:GetSamplingStatisticSummaries` y `logs:PutRetentionPolicy` se agregaron para que el agente de CloudWatch pueda publicar seguimientos de X-Ray y modificar los periodos de retención de los grupos de registros.	12 de febrero de 2024

Cambio	Descripción	Fecha
CloudWatchFullAccessV2: actualizar a una política existente	CloudWatch añadió permisos a CloudWatchFullAccessV2. Los permisos existente s para las acciones de CloudWatch Synthetics, X-Ray y CloudWatch RUM y los nuevos permisos para CloudWatch Application Signals se añadieron para que los usuarios con esta política puedan administr ar CloudWatch Application Signals. Se añadió el permiso para crear el rol vinculado al servicio de CloudWatch Application Signals para permitir que CloudWatch Application Signals detecte los datos de telemetría en registros, métricas, seguimien tos y etiquetas.	5 de diciembre de 2023

Cambio	Descripción	Fecha
CloudWatchReadOnlyAccess: actualizar a una política existente	CloudWatch agregó permisos a CloudWatchReadOnly Access. Los permisos existentes de solo lectura para las acciones de CloudWatch Synthetics, X-Ray y CloudWatch RUM y los nuevos permisos de solo lectura para CloudWatch Application Signals se añadieron para que los usuarios con esta política puedan clasificar y diagnosticar los problemas en el estado del servicio notificados por CloudWatch Application Signals. El permiso de `cloudwatch:GenerateQuery` se añadió para que los usuarios con esta política puedan generar una cadena de consulta de Información de métricas de CloudWatch a partir de una petición en lenguaje natural.	5 de diciembre de 2023

Cambio	Descripción	Fecha
CloudWatchApplicationSignalsServiceRolePolicy: nueva política	CloudWatch agregó una nueva política CloudWatchApplicationSignalsServiceRolePolicy. La política CloudWatchApplicationSignalsServiceRolePolicy otorga permisos para que una próxima característica recopile datos de registros de CloudWatch, datos de registros de seguimiento de X-Ray, datos de métricas de CloudWatch y datos de etiquetado.	9 de noviembre de 2023
AWSServiceRoleForCloudWatchMetrics_DBPerfinSightsServiceRolePolicy: nueva política	CloudWatch agregó una nueva política, llamada AWSServiceRoleForCloudWatchMetrics_DbPerfInsightsServiceRolePolicy. La política AWSServiceRoleForCloudWatchMetrics_DBPerfInsightsServiceRolePolicy concede permiso a CloudWatch para obtener métricas de Performance Insights de las bases de datos en su nombre.	20 de septiembre de 2023

Cambio	Descripción	Fecha
CloudWatchReadOnlyAccess: actualizar a una política existente	CloudWatch agregó un permiso a CloudWatchReadOnlyAccess. El permiso `application-autoscaling:DescribeScalingPolicies` se añadió para que los usuarios con esta política puedan acceder a la información sobre las políticas de escalado automático de aplicaciones.	14 de septiembre de 2023
CloudWatchFullAccessV2: nueva política	CloudWatch agregó una nueva política CloudWatchFullAccessV2. CloudWatchFullAccessV2 otorga acceso total a las acciones y los recursos de CloudWatch y, al mismo tiempo, amplía el alcance de los permisos concedidos a otros servicios, como Amazon SNS y Amazon EC2 Auto Scaling. Para obtener más información, consulte CloudWatchFullAccessV2.	1 de agosto de 2023

Cambio	Descripción	Fecha
AWSServiceRoleForInternetMonitor: actualización a una política existente	Amazon CloudWatch Internet Monitor agregó nuevos permisos para supervisar los recursos del Equilibrador de carga de red. Los permisos `elasticloadbalancing:DescribeLoadBalancers` y `ec2:DescribeNetworkInterfaces` son necesarios para que Internet Monitor pueda supervisar el tráfico del Equilibrador de carga de red de los clientes mediante el análisis de los registros de flujo de los recursos de NLB. Para obtener más información, consulte Uso de Amazon CloudWatch Internet Monitor.	15 de julio de 2023

Cambio	Descripción	Fecha
CloudWatchReadOnlyAccess: actualizar a una política existente	CloudWatch agregó permisos a CloudWatchReadOnly Access. Se agregaron los permisos `logs:StartLiveTail` y `logs:StopLiveTail` a fin de que los usuarios con esta política puedan usar la consola para iniciar y detener las sesiones de seguimiento en tiempo real de Registros de CloudWatch. Para obtener más información, consulte Use live tail to view logs in near real time.	6 de junio de 2023
CloudWatchCrossAccountSharingConfiguration: política nueva	CloudWatch agregó una política nueva para permitir administrar los enlaces de observabilidad entre cuentas de CloudWatch que comparten las métricas de CloudWatch. Para obtener más información, consulte Observabilidad entre cuentas de CloudWatch.	27 de noviembre de 2022

Cambio	Descripción	Fecha
OAMFullAccess: política nueva	CloudWatch agregó una política nueva para permitir la administración completa de los enlaces y sumideros de observabilidad de entre cuentas de CloudWatch. Para obtener más información, consulte Observabilidad entre cuentas de CloudWatch.	27 de noviembre de 2022
OamReadOnlyAccess: política nueva	CloudWatch agregó una política nueva para permitir ver información sobre los enlaces y los sumideros de observabilidad de entre cuentas de CloudWatch. Para obtener más información, consulte Observabilidad entre cuentas de CloudWatch.	27 de noviembre de 2022
CloudWatchFullAccess: actualizar a una política existente	CloudWatch agregó permisos a CloudWatchFullAccess. Los permisos de `oam:ListSinks` y `oam:ListAttachedLinks` se agregaron para que los usuarios con esta política puedan usar la consola para ver los datos compartidos desde las cuentas de origen en la observabilidad entre cuentas de CloudWatch.	27 de noviembre de 2022

Cambio	Descripción	Fecha
CloudWatchReadOnlyAccess: actualizar a una política existente	CloudWatch agregó permisos a CloudWatchReadOnly Access. Los permisos de `oam:ListSinks` y `oam:ListAttachedLinks` se agregaron para que los usuarios con esta política puedan usar la consola para ver los datos compartidos desde las cuentas de origen en la observabilidad entre cuentas de CloudWatch.	27 de noviembre de 2022

Cambio	Descripción	Fecha
AmazonCloudWatchRUMServiceRolePolicy: actualización de una política existente	CloudWatch RUM actualizó una clave de condición en AmazonCloudWatchRUMServiceRolePolicy. La clave de condición `"Condition": { "StringEquals": { "cloudwatch:namespace": "AWS/RUM" } }` se ha cambiado por la siguiente para que CloudWatch RUM pueda enviar métricas personalizadas a espacios de nombres de métricas personalizadas. `"Condition": { "StringLike": { "cloudwatch:namespace": [ "RUM/CustomMetrics/*", "AWS/RUM" ] } }`	2 de febrero de 2023

Cambio	Descripción	Fecha
AmazonCloudWatchRUMReadOnlyAccess: política actualizada	CloudWatch agregó permisos a la política AmazonCloudWatchRUMReadOnlyAccess. Se agregaron los permisos `rum:ListRumMetricsDestinations` y `rum:BatchGetRumMetricsDefinitions` para que CloudWatch RUM pueda enviar métricas ampliadas a CloudWatch y Evidently.	27 de octubre de 2022
AmazonCloudWatchRUMServiceRolePolicy: actualización de una política existente	CloudWatch RUM agregó permisos a la política AmazonCloudWatchRUMServiceRolePolicy. Se agregó el permiso `cloudwatch:PutMetricData` para que CloudWatch RUM pueda enviar métricas ampliadas a CloudWatch.	26 de octubre de 2022

Cambio	Descripción	Fecha
CloudWatchEvidentlyReadOnlyAccess: actualización de una política existente	CloudWatch Evidently agregó permisos a CloudWatchEvidentlyReadOnlyAccess. Los permisos `evidently:GetSegment`, `evidently:ListSegments` y `evidently:ListSegmentReferences` se agregaron para que los usuarios con esta política puedan ver los segmentos de audiencia de Evidently que se han creado.	12 de agosto de 2022
CloudWatchSyntheticsFullAccess: actualización de una política existente	CloudWatch Synthetics agregó permisos a CloudWatchSyntheticsFullAccess. Los permisos `lambda:DeleteFunction` y `lambda:DeleteLayerVersion` se han agregado permisos para que CloudWatch Synthetics pueda eliminar recursos relacionados cuando se elimina un valor controlado. Se ha agregado la `iam:ListAttachedRolePolicies` para que los clientes puedan ver las políticas asociadas a un rol de IAM de un valor controlado.	6 de mayo de 2022

Cambio	Descripción	Fecha
AmazonCloudWatchRUMFullAccess: nueva política	CloudWatch agregó una nueva política para permitir la administración completa de CloudWatch RUM. CloudWatch RUM le permite llevar a cabo una supervisión real de usuarios de su aplicación web. Para obtener más información, consulte Uso de CloudWatch RUM.	29 de noviembre de 2021
AmazonCloudWatchRUMReadOnlyAccess: nueva política	CloudWatch agregó una nueva política para permitir el acceso de solo lectura a CloudWatch RUM. CloudWatch RUM le permite llevar a cabo una supervisión real de usuarios de su aplicación web. Para obtener más información, consulte Uso de CloudWatch RUM.	29 de noviembre de 2021

Cambio	Descripción	Fecha
CloudWatchEvidentlyFullAccess: nueva política	CloudWatch agregó una nueva política para permitir la administración completa de CloudWatch Evidently. CloudWatch Evidently le permite realizar experimentos A/B de sus aplicaciones web e implementarlos de forma gradual. Para obtener más información, consulte Realice lanzamientos y experimentos A/B con CloudWatch Evidently .	29 de noviembre de 2021
CloudWatchEvidentlyReadOnlyAccess: nueva política	CloudWatch agregó una nueva política para permitir el acceso de solo lectura a CloudWatch Evidently. CloudWatch Evidently le permite realizar experimentos A/B de sus aplicaciones web e implementarlos de forma gradual. Para obtener más información, consulte Realice lanzamientos y experimentos A/B con CloudWatch Evidently .	29 de noviembre de 2021

Cambio	Descripción	Fecha
AWSServiceRoleForCloudWatchRUM: nueva política administrada	CloudWatch agregó una política para un nuevo rol vinculado a servicios para permitir que CloudWatch RUM publique los datos de supervisión en otros servicios relevantes de AWS.	29 de noviembre de 2021

Cambio	Descripción	Fecha
CloudWatchSyntheticsFullAccess: actualización de una política existente	CloudWatch Synthetics agregó permisos aCloudWatchSyntheticsFullAccess (Acceso completo a CloudWatch Synthetics) y también cambió el alcance de un permiso. Se agregó el permiso de `kms:ListAliases` para que los usuarios puedan publicar las claves de AWS KMS disponibles que se pueden utilizar para cifrar artefactos de valor controlado. Se agregó el permiso de `kms:DescribeKey` para que los usuarios puedan ver los detalles de las claves que se utilizarán para cifrar los artefactos de valor controlado. Finalmente, se agregó el permiso de `kms:Decrypt` para permitir que los usuarios descifren artefactos de valor controlado. Esta capacidad de descifrado se limita al uso en los recursos dentro de los buckets de Amazon S3. El ámbito del `Resource` (recurso) del permiso de `s3:GetBucketLocation` se modificó de `*` a `arn:aws:s3:::*` .	29 de septiembre de 2021

Cambio	Descripción	Fecha
CloudWatchSyntheticsFullAccess: actualización de una política existente	CloudWatch Synthetics agregó un permiso a CloudWatchSyntheticsFullAccess. Se agregó el permiso `lambda:UpdateFunctionCode` para que los usuarios con esta política puedan cambiar la versión de tiempo de ejecución de los canaries.	20 de julio de 2021
AWSCloudWatchAlarms_ActionSSMIncidentsServiceRolePolicy: nueva política administrada	CloudWatch agregó una nueva política de IAM administrada para permitir que CloudWatch cree incidentes en Incident Manager de AWS Systems Manager.	10 de mayo de 2021
CloudWatchAutomaticDashboardsAccess: actualización de una política existente	CloudWatch agregó un permiso a la política administrada CloudWatchAutomaticDashboardsAccess El permiso `synthetics:DescribeCanariesLastRun` se agregó a esta política a fin de permitir que los usuarios del panel de las cuentas cruzadas visualicen los detalles sobre las ejecuciones de los valores controlados de CloudWatch Synthetics.	20 de abril de 2021

Cambio	Descripción	Fecha
CloudWatch comenzó a realizar seguimientos de los cambios	CloudWatch comenzó a realizar seguimientos de los cambios para las Políticas administradas de AWS.	14 de abril de 2021

Uso de claves de condición para limitar el acceso a los espacios de nombres de CloudWatch

Utilice claves de condición de IAM para que los usuarios solo puedan publicar métricas en los espacios de nombres de CloudWatch que usted especifique.

Permiso sobre la publicación en un solo espacio de nombres

La siguiente política limita la capacidad de publicación del usuario a solo el espacio de nombres denominado `MyCustomNamespace`.

```
{
    "Version": "2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Resource": "*",
        "Action": "cloudwatch:PutMetricData",
        "Condition": {
            "StringEquals": {
                "cloudwatch:namespace": "MyCustomNamespace"
            }
        }
    }
}
```

Exclusión de la publicación de un espacio de nombres

La siguiente política permite al usuario publicar métricas en cualquier espacio de nombres excepto en `CustomNamespace2`.

```
{
    "Version": "2012-10-17",
```

```
    "Statement": [
        {
            "Effect": "Allow",
            "Resource": "*",
            "Action": "cloudwatch:PutMetricData"
        },
        {
            "Effect": "Deny",
            "Resource": "*",
            "Action": "cloudwatch:PutMetricData",
            "Condition": {
                "StringEquals": {
                    "cloudwatch:namespace": "CustomNamespace2"
                }
            }
        }
    ]
}
```

Uso de claves de condición para limitar el acceso de los usuarios de Contributor Insights a los grupos de registro

Para crear una regla en Contributor Insights y ver los resultados, un usuario debe tener el permiso `cloudwatch:PutInsightRule`. De forma predeterminada, un usuario con este permiso puede crear una regla de Contributor Insights que evalúe cualquier grupo de registros en CloudWatch Logs y, a continuación, ver los resultados. Los resultados pueden contener datos de colaborador para esos grupos de registro.

Puede crear políticas de IAM con claves de condición para conceder a los usuarios el permiso para registrar reglas de Contributor Insights para algunos grupos de registro, al tiempo que les impide registrar reglas para ver estos datos de otros grupos de registro.

Para obtener más información sobre el elemento `Condition` en la política de IAM, consulte IAM JSON policy elements: Condition (Elementos de las políticas JSON de IAM: Condición).

Permitir el acceso a reglas de escritura y ver resultados solo para determinados grupos de registro

La siguiente política permite al usuario acceder al registro de reglas y ver resultados para el grupo de registro denominado `AllowedLogGroup` y todos los grupos de registro que tienen nombres que comienzan por `AllowedWildCard`. No concede acceso a las reglas de escritura ni a ver resultados de reglas para ningún otro grupo de registros.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowCertainLogGroups",
            "Effect": "Allow",
            "Action": "cloudwatch:PutInsightRule",
            "Resource": "arn:aws:cloudwatch:*:*:insight-rule/*",
            "Condition": {
                "ForAllValues:StringEqualsIgnoreCase": {
                    "cloudwatch:requestInsightRuleLogGroups": [
                        "AllowedLogGroup",
                        "AllowedWildcard*"
                    ]
                }
            }
        }
    ]
}
```

Denegar reglas de escritura para grupos de registro específicos pero permitir reglas de escritura para todos los demás grupos de registro

La siguiente política deniega explícitamente al usuario el acceso para registrar reglas y ver resultados de reglas para el grupo de registro denominado `ExplicitlyDeniedLogGroup`, pero permite registrar reglas y ver resultados de reglas para todos los demás grupos de registro.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowInsightRulesOnLogGroupsByDefault",
            "Effect": "Allow",
            "Action": "cloudwatch:PutInsightRule",
            "Resource": "arn:aws:cloudwatch:*:*:insight-rule/*"

        },
        {
            "Sid": "ExplicitDenySomeLogGroups",
            "Effect": "Deny",
            "Action": "cloudwatch:PutInsightRule",
            "Resource": "arn:aws:cloudwatch:*:*:insight-rule/*",
```

```
            "Condition": {
                "ForAllValues:StringEqualsIgnoreCase": {
                    "cloudwatch:requestInsightRuleLogGroups": [
                        "/test/alpine/ExplicitlyDeniedLogGroup"
                    ]
                }
            }
        }
    ]
}
```

Uso de claves de condición para limitar las acciones de la alarma

Cuando las alarmas de CloudWatch cambian de estado, pueden realizar diferentes acciones, como detener y terminar instancias EC2 y realizar acciones de Systems Manager. Estas acciones se pueden iniciar cuando la alarma cambia a cualquier estado, incluidos los estados ALARM (ALARMA), OK o INSUFICIENT_DATA (DATOS INSUFICIENTES).

Uso de la clave de condición `cloudwatch:AlarmActions` para permitir que un usuario cree alarmas que solo pueden realizar las acciones especificadas cuando cambia el estado de la alarma. Por ejemplo, puede permitir que un usuario cree alarmas que solo pueden realizar acciones que no sean acciones de EC2.

Permitir a un usuario crear alarmas que solo puedan enviar notificaciones de Amazon SNS o realizar acciones de Systems Manager

La siguiente política limita al usuario a crear alarmas que solo pueden enviar notificaciones de Amazon SNS y realizar acciones de Systems Manager. El usuario no puede crear alarmas que realicen acciones de EC2.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "CreateAlarmsThatCanPerformOnlySNSandSSMActions",
            "Effect": "Allow",
            "Action": "cloudwatch:PutMetricAlarm",
            "Resource": "*",
            "Condition": {
                "ForAllValues:StringLike": {
                    "cloudwatch:AlarmActions": [
                        "arn:aws:sns:*",
```

```
                    "arn:aws:ssm:*"
                ]
            }
        }
    }
  ]
}
```

Uso de roles vinculados a servicios para CloudWatch

Amazon CloudWatch utiliza roles vinculados a servicios de AWS Identity and Access Management (IAM). Un rol vinculado a un servicio es un tipo único de rol de IAM que está vinculado directamente a CloudWatch. Los roles vinculados a servicios están predefinidos por CloudWatch e incluyen todos los permisos que el servicio requiere para llamar a otros servicios de AWS en su nombre.

El rol vinculado a un servicio en CloudWatch le permite configurar alarmas de CloudWatch que puedan terminar, detener o reiniciar instancias de Amazon EC2 sin que tenga que agregar manualmente los permisos necesarios. Otro rol vinculado a servicios permite que una cuenta de supervisión tenga acceso a los datos de CloudWatch de otras cuentas que haya especificado, con el fin de crear paneles para cuentas y regiones cruzadas.

CloudWatch define los permisos de estos roles vinculados a servicios y, a menos que esté definido de otra manera, solo CloudWatch puede asumir el rol. Los permisos definidos incluyen las políticas de confianza y de permisos, y que la política de permisos no se pueda asociar a ninguna otra entidad de IAM.

Las funciones se pueden eliminar únicamente después de eliminar primero sus recursos relacionados. Esta restricción protege los recursos de CloudWatch, ya que evita que se puedan quitar accidentalmente permisos de acceso a ellos.

Para obtener información sobre otros servicios que son compatibles con los roles vinculados a servicios, consulte Servicios de AWS que funcionan con IAM y busque los servicios que muestran Yes (Sí) en la columna Service-Linked Role (Rol vinculado a servicios). Elija una opción Sí con un enlace para ver la documentación acerca del rol vinculado a servicios en cuestión.

Permisos de roles vinculados a servicios para acciones de alarmas de EC2 de CloudWatch

CloudWatch usa el rol vinculado al servicio denominado AWSServiceRoleForCloudWatchEvents: CloudWatch utiliza este rol vinculado al servicio para realizar acciones de alarma de Amazon EC2.

El rol vinculado al servicio AWSServiceRoleForCloudWatchEvents confía en el servicio de CloudWatch Events para asumir el rol: CloudWatch Events invoca las acciones de terminar, detener o reiniciar acciones de instancia cuando la alarma las llama.

La política de permisos del rol vinculado al servicio AWSServiceRoleForCloudWatchEvents permite que CloudWatch Events realice las siguientes acciones en instancias de Amazon EC2:

- `ec2:StopInstances`
- `ec2:TerminateInstances`
- `ec2:RecoverInstances`
- `ec2:DescribeInstanceRecoveryAttribute`
- `ec2:DescribeInstances`
- `ec2:DescribeInstanceStatus`

La política de permisos del rol vinculado al servicio AWSServiceRoleForCloudWatchCrossAccount permite que CloudWatch realice las siguientes acciones:

- `sts:AssumeRole`

Permisos de roles vinculados a un servicio para CloudWatch Application Signals

CloudWatch Application Signals utiliza el rol vinculado a un servicio denominado AWSServiceRoleForCloudWatchApplicationSignals: CloudWatch utiliza este rol vinculado a un servicio para recopilar datos de los Registros de CloudWatch, datos de seguimiento de X-Ray, datos de las métricas de CloudWatch y datos de etiquetado de las aplicaciones que haya habilitado para CloudWatch Application Signals.

El rol vinculado al servicio AWSServiceRoleForCloudWatchApplicationSignals confía en que CloudWatch Application Signals asuma el rol. Application Signals recopila los datos de registros, seguimientos, métricas y etiquetas de su cuenta.

AWSServiceRoleForCloudWatchApplicationSignals tiene una política de IAM asociada denominada CloudWatchApplicationSignalsServiceRolePolicy. Esta política otorga permiso a CloudWatch Application Signals para recopilar datos de supervisión y etiquetado de otros servicios de AWS relevantes. Incluye permisos para que Application Signals complete las siguientes acciones:

- `xray:GetServiceGraph`

- logs:StartQuery
- logs:GetQueryResults
- cloudwatch:GetMetricData
- cloudwatch:ListMetrics
- tag:GetResources

El contenido completo de la política CloudWatchApplicationSignalsServiceRolePolicy es el siguiente:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "XRayPermission",
            "Effect": "Allow",
            "Action": [
                "xray:GetServiceGraph"
            ],
            "Resource": [
                "*"
            ],
            "Condition": {
                "StringEquals": {
                    "aws:ResourceAccount": "${aws:PrincipalAccount}"
                }
            }
        },
        {
            "Sid": "CWLogsPermission",
            "Effect": "Allow",
            "Action": [
                "logs:StartQuery",
                "logs:GetQueryResults"
            ],
            "Resource": [
                "arn:aws:logs:*:*:log-group:/aws/appsignals/*:*",
                "arn:aws:logs:*:*:log-group:/aws/application-signals/data:*"
            ],
            "Condition": {
                "StringEquals": {
                    "aws:ResourceAccount": "${aws:PrincipalAccount}"
                }
```

```
            }
        },
        {
            "Sid": "CWListMetricsPermission",
            "Effect": "Allow",
            "Action": [
                "cloudwatch:ListMetrics"
            ],
            "Resource": [
                "*"
            ],
            "Condition": {
                "StringEquals": {
                    "aws:ResourceAccount": "${aws:PrincipalAccount}"
                }
            }
        },
        {
            "Sid": "CWGetMetricDataPermission",
            "Effect": "Allow",
            "Action": [
                "cloudwatch:GetMetricData"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Sid": "TagsPermission",
            "Effect": "Allow",
            "Action": [
                "tag:GetResources"
            ],
            "Resource": [
                "*"
            ],
            "Condition": {
                "StringEquals": {
                    "aws:ResourceAccount": "${aws:PrincipalAccount}"
                }
            }
        }
    ]
```

```
}
```

Permisos de roles vinculados a servicios para las acciones de alarmas de CloudWatch Systems Manager OpsCenter

CloudWatch usa el rol vinculado al servicio denominado AWSServiceRoleForCloudWatchAlarms_ActionSSM: CloudWatch utiliza este rol vinculado a servicios para realizar acciones de OpsCenter de Systems Manager cuando una alarma de CloudWatch entra en el estado ALARM (ALARMA).

El rol vinculado al servicio AWSServiceRoleForCloudWatchAlarms_ActionSSM confía en el servicio de CloudWatch para asumir el rol. Las alarmas de CloudWatch invocan las acciones de OpsCenter de Systems Manager cuando la alarma las llama.

La política de permisos del rol vinculado al servicio AWSServiceRoleForCloudWatchAlarms_ActionSSM permite que Systems Manager realice las siguientes acciones:

- `ssm:CreateOpsItem`

Permisos de roles vinculados a servicios para acciones de alarmas de CloudWatch Systems Manager Incident Manager

CloudWatch utiliza el rol vinculado al servicio denominado AWSServiceRoleForCloudWatchAlarms_ActionSSMIncidents: CloudWatch utiliza este rol vinculado a servicios para comenzar incidentes de Incident Manager cuando una alarma de CloudWatch entra en el estado ALARM (ALARMA).

El rol vinculado al servicio AWSServiceRoleForCloudWatchAlarms_ActionSSMIncidents confía en que los servicios de CloudWatch asuman el rol. Las alarmas de CloudWatch invocan la acción de Incident Manager de Systems Manager cuando la alarma la llama.

La política de permisos del rol vinculado al servicio AWSServiceRoleForCloudWatchAlarms_ActionSSMIncidents permite que Systems Manager realice las siguientes acciones:

- `ssm-incidents:StartIncident`

Permisos de roles vinculados a servicios para las cuentas y Regiones cruzadas de CloudWatch

CloudWatch utiliza el rol vinculado al servicio denominado AWSServiceRoleForCloudWatchCrossAccount: CloudWatch utiliza este rol para tener acceso a los datos de CloudWatch de otras cuentas de AWS que especifique. El SLR solo proporciona el permiso de asumir el rol para permitir que el servicio de CloudWatch asuma el rol en la cuenta de uso compartido. Es el rol de uso compartido que proporciona acceso a los datos.

La política de permisos del rol vinculado al servicio AWSServiceRoleForCloudWatchCrossAccount permite que CloudWatch realice las siguientes acciones:

- `sts:AssumeRole`

El rol vinculado al servicio AWSServiceRoleForCloudWatchCrossAccount confía en que el servicio CloudWatch asuma el rol.

Permisos de roles vinculados a un servicio para la base de datos CloudWatch Performance Insights

CloudWatch utiliza el rol vinculado a un servicio denominado AWSServiceRoleForCloudWatchMetrics_DbPerfInsights. CloudWatch usa esta función para recuperar las métricas de Performance Insights para crear alarmas e instantáneas.

El rol vinculado al servicio AWSServiceRoleForCloudWatchMetrics_DbPerfInsights tiene adjunta la política de IAM de `AWSServiceRoleForCloudWatchMetrics_DbPerfInsightsServiceRolePolicy`. El contenido de esta política se detalla a continuación:

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Effect": "Allow",
   "Action": [
    "pi:GetResourceMetrics"
   ],
   "Resource": "*",
   "Condition": {
    "StringEquals": {
```

```
      "aws:ResourceAccount": "${aws:PrincipalAccount}"
     }
    }
   }
  ]
}
```

El rol vinculado al servicio AWSServiceRoleForCloudWatchMetrics_DbPerfInsights confía en que el servicio CloudWatch asuma el rol.

Creación de un rol vinculado al servicio para CloudWatch

No es necesario que se cree manualmente ninguno de estos roles vinculados a servicios. La primera vez que crea una alarma en la AWS Management Console, la CLI de IAM o en la API de IAM, CloudWatch crea AWSServiceRoleForCloudWatchEvents y AWSServiceRoleForCloudWatchAlarms_ActionSSM.

La primera vez que habilita el detector de servicios y topología, Application Signals crea AWSServiceRoleForCloudWatchApplicationSignals por usted.

Cuando habilita por primera vez una cuenta como cuenta de supervisión para la funcionalidad entre cuentas y regiones, CloudWatch crea AWSServiceRoleForCloudWatchCrossAccount.

La primera vez que crea una alarma que utiliza la función matemática `DB_PERF_INSIGHTS` métrica, CloudWatch crea AWSServiceRoleForCloudWatchMetrics_DBPerfInsights por usted.

Para obtener más información, consulte Crear un rol vinculado a un servicio en la Guía del usuario de IAM.

Modificación de un rol vinculado a un servicio para CloudWatch

CloudWatch no permite que se editen los roles AWSServiceRoleForCloudWatchEvents, AWSServiceRoleForCloudWatchAlarms_ActionSSM, AWSServiceRoleForCloudWatchCrossAccount oAWSServiceRoleForCloudWatchMetrics_DbPerfInsights. Después de crear estos roles, no puede cambiar sus nombres, porque diversas entidades pueden hacer referencia a ellos. Sin embargo, puede editar la descripción de estos roles mediante IAM.

Edición de la descripción de un rol vinculado a un servicio (consola de IAM)

Puede utilizar la consola de IAM para editar la descripción de un rol vinculado a un servicio.

Para editar la descripción de un rol vinculado a un servicio (consola)

1. En el panel de navegación de la consola de IAM, elija Roles.
2. Seleccione el nombre del rol que desea modificar.
3. En el extremo derecho de Role description, seleccione Edit.
4. Escriba una nueva descripción en el cuadro y elija Save (Guardar).

Edición de la descripción de un rol vinculado a servicio (AWS CLI)

Puede utilizar comandos de IAM desde la AWS Command Line Interface para editar la descripción de un rol vinculado a un servicio.

Para cambiar la descripción de un rol vinculado a un servicio (AWS CLI)

1. (Opcional) Para ver la descripción actual de un rol, ejecute uno de los siguientes comandos:

   ```
   $ aws iam get-role --role-name role-name
   ```

   Utilice el nombre del rol, no el ARN, para hacer referencia a los roles con los comandos de AWS CLI. Por ejemplo, si un rol tiene el ARN `arn:aws:iam::123456789012:role/myrole`, debe referirse a él como **`myrole`**.
2. Para actualizar la descripción de un rol vinculado a un servicio, ejecute el siguiente comando:

   ```
   $ aws iam update-role-description --role-name role-name --description description
   ```

Edición de la descripción de un rol vinculado a un servicio (API de IAM)

Puede utilizar la API de IAM para editar la descripción de un rol vinculado a un servicio.

Para cambiar la descripción de un rol vinculado a un servicio (API)

1. (Opcional) Para ver la descripción actual de una función, ejecute el siguiente comando:

   GetRole
2. Para actualizar la descripción de una función, use el siguiente comando:

   UpdateRoleDescription

Eliminación de un rol vinculado a un servicio para CloudWatch

Si ya no tiene alarmas que detengan, terminen o reinicien instancias EC2, le recomendamos que elimine el rol AWSServiceRoleForCloudWatchEvents.

Si ya no cuenta con alarmas que realicen acciones OpsCenter de Systems Manager, se recomienda que elimine el rol AWSServiceRoleForCloudWatchAlarms_ActionSSM.

Si elimina todas las alarmas que utilizan la función matemática de `DB_PERF_INSIGHTS` métricas, le recomendamos que elimine el rol vinculado a un servicio AWSServiceRoleForCloudWatchMetrics_DBPerfInsights.

De esta forma no tiene una entidad no utilizada que no se supervise ni mantenga de forma activa. Sin embargo, debe limpiar el rol vinculado al servicio antes de eliminarlo.

Limpiar un rol vinculado a servicios

Antes de poder utilizar IAM para eliminar un rol vinculado a un servicio, primero debe confirmar que dicho rol no tiene sesiones activas y eliminar los recursos que utiliza.

Para comprobar si el rol vinculado a un servicio tiene una sesión activa en la consola de IAM

1. Abra la consola de IAM en https://console.aws.amazon.com/iam/.
2. Seleccione Roles en el panel de navegación. Elija el nombre (no la casilla de verificación) del rol AWSServiceRoleForCloudWatchEvents.
3. En la página Summary del rol seleccionado, elija Access Advisor y revise la actividad reciente del rol vinculado al servicio.

> **Note**
>
> Si no está seguro acerca de si CloudWatch utiliza el rol AWSServiceRoleForCloudWatchEvents, intente eliminar el rol para verificarlo. Si el servicio está utilizando el rol, este no podrá eliminarse y podrá ver las regiones en las que se está utilizando. Si el rol se está utilizando, debe esperar que la sesión finalice para poder eliminarlo. No se puede revocar la sesión de un rol vinculado a servicios.

Eliminación de un rol vinculado a un servicio (consola de IAM)

Puede utilizar la consola de IAM para eliminar un rol vinculado a un servicio.

Para eliminar un rol vinculado a un servicio (consola)

1. Abra la consola de IAM en https://console.aws.amazon.com/iam/.
2. Seleccione Roles en el panel de navegación. Seleccione la casilla junto al nombre del rol que desea eliminar, no el nombre ni la fila.
3. En Role actions, elija Delete role.
4. En el cuadro de diálogo de confirmación, revise los datos del último acceso al servicio, que muestra cuándo cada uno de los roles seleccionados tuvo acceso a un servicio de AWS por última vez. Esto lo ayuda a confirmar si el rol está actualmente activo. Para continuar, elija Yes, Delete.
5. Consulte las notificaciones de la consola de IAM para supervisar el progreso de la eliminación del rol vinculado al servicio. Debido a que el proceso de eliminación del rol vinculado al servicio de IAM es asíncrono, la tarea de eliminación puede realizarse correcta o incorrectamente después de que envía la solicitud de eliminación. Si la tarea no se realiza correctamente, elija View details o View Resources desde las notificaciones para obtener información acerca de por qué no se pudo eliminar el rol. Si la eliminación no pudo producirse porque hay recursos en el servicio que está utilizando el rol, entonces el motivo del error incluye una lista de recursos.

Eliminar un rol vinculado a un servicio (AWS CLI)

Puede utilizar los comandos de IAM desde la AWS Command Line Interface para eliminar un rol vinculado a un servicio.

Para eliminar un rol vinculado a un servicio (AWS CLI)

1. Como los roles vinculados a servicios no se puede eliminar si están en uso o tienen recursos asociados, debe enviar una solicitud de eliminación. Esta solicitud puede denegarse si no se cumplen estas condiciones. Debe apuntar el valor `deletion-task-id` de la respuesta para comprobar el estado de la tarea de eliminación. Escriba el siguiente comando para enviar una solicitud de eliminación de un rol vinculado a un servicio:

```
$ aws iam delete-service-linked-role --role-name service-linked-role-name
```

2. Escriba el siguiente comando para comprobar el estado de la tarea de eliminación:

```
$ aws iam get-service-linked-role-deletion-status --deletion-task-id deletion-task-id
```

El estado de la tarea de eliminación puede ser `NOT_STARTED`, `IN_PROGRESS`, `SUCCEEDED` o `FAILED`. Si ocurre un error durante la eliminación, la llamada devuelve el motivo del error para que pueda resolver el problema.

Eliminación de un rol vinculado a un servicio (API de IAM)

Puede utilizar la API de IAM para eliminar un rol vinculado a un servicio.

Para eliminar un rol vinculado a un servicio (API)

1. Para enviar una solicitud de eliminación de un rol vinculado a un servicio, realice una llamada a DeleteServiceLinkedRole. En la solicitud, especifique el nombre de rol que desea eliminar.

   Como los roles vinculados a servicios no se puede eliminar si están en uso o tienen recursos asociados, debe enviar una solicitud de eliminación. Esta solicitud puede denegarse si no se cumplen estas condiciones. Debe apuntar el valor `DeletionTaskId` de la respuesta para comprobar el estado de la tarea de eliminación.
2. Para comprobar el estado de la tarea de eliminación, realice una llamada a GetServiceLinkedRoleDeletionStatus. En la solicitud, especifique el valor de `DeletionTaskId`.

   El estado de la tarea de eliminación puede ser `NOT_STARTED`, `IN_PROGRESS`, `SUCCEEDED` o `FAILED`. Si ocurre un error durante la eliminación, la llamada devuelve el motivo del error para que pueda resolver el problema.

Actualizaciones de CloudWatch para roles vinculados a servicios de AWS

Es posible consultar los detalles sobre las actualizaciones de las políticas administradas de AWS para CloudWatch debido a que este servicio comenzó a realizar el seguimiento de estos cambios. Para obtener alertas automáticas sobre cambios en esta página, suscríbase a la fuente RSS en la página de historial de documentos de CloudWatch.

Cambio	Descripción	Fecha
AWSServiceRoleForCloudWatchApplicationSignals:	CloudWatch agrega más grupos de registros al alcance	24 de abril de 2024

Cambio	Descripción	Fecha
actualización de los permisos de la política de roles vinculados a servicios	de los permisos `logs:StartQuery` y `logs:GetQueryResults` otorgados por este rol.	
AWSServiceRoleForCloudWatchApplicationSignals: nuevo rol vinculado a un servicio	CloudWatch añadió este nuevo rol vinculado a servicios para permitir a CloudWatch Application Signals recopilar datos de Registros de CloudWatch, datos del seguimiento de X-Ray, datos de métricas de CloudWatch y datos del etiquetado de las aplicaciones que haya habilitado para CloudWatch Application Signals.	9 de noviembre de 2023
AWSServiceRoleForCloudWatchMetrics_DBPerfInsights: nuevo rol vinculado a un servicio	CloudWatch agregó este nuevo rol vinculado a servicios para permitir que CloudWatch obtenga métricas de Performance Insights para alarmas e instantáneas. Se adjunta una política de IAM a este rol, y la política concede permiso a CloudWatch para obtener métricas de Performance Insights en su nombre.	13 de septiembre de 2023

Cambio	Descripción	Fecha
AWSServiceRoleForCloudWatchAlarms_ActionSSMIncidents: nuevo rol vinculado a un servicio	CloudWatch agregó un nuevo rol vinculado a un servicio para permitir que CloudWatch cree incidentes en Incident Manager de AWS Systems Manager.	26 de abril de 2021
CloudWatch comenzó a realizar seguimientos de los cambios	CloudWatch comenzó a realizar seguimientos de los cambios para los roles vinculados al servicio.	26 de abril de 2021

Uso de roles vinculados a servicios para CloudWatch RUM

CloudWatch RUM utiliza un rol vinculado al servicio AWS Identity and Access Management (IAM). Un rol vinculado a un servicio es un tipo único de rol de IAM que está vinculado directamente a RUM. El rol vinculado al servicio está predefinido por RUM e incluye todos los permisos que el servicio requiere para llamar a otros servicios de AWS en su nombre.

RUM define los permisos de estos roles vinculados al servicio y, a menos que esté definido de otra manera, solo RUM puede asumir el rol. Los permisos definidos incluyen las políticas de confianza y de permisos, y que la política de permisos no se pueda adjuntar a ninguna otra entidad de IAM.

Los roles solo se pueden eliminar después de eliminar primero sus recursos relacionados. Esta restricción protege los recursos de RUM, ya que evita que se puedan quitar permisos de acceso a ellos de forma accidental.

Para obtener información sobre otros servicios que admiten roles vinculados a servicios, consulte Servicios de AWS que funcionan con IAM y busque los servicios que muestran Sí en la columna Roles vinculados a servicios. Elija una opción Sí con un enlace para ver la documentación acerca del rol vinculado al servicio en cuestión.

Permisos de roles vinculados a servicios para RUM

RUM usa el rol vinculado a un servicio denominado AWSServiceRoleForCloudWatchRUM: esta función permite que RUM envíe datos de seguimiento de AWS X-Ray a su cuenta, para monitores de aplicaciones para los que habilita el seguimiento de X-Ray.

El rol vinculado al servicio AWSServiceRoleForCloudWatchCrossAccount confía en que el servicio de X-Ray asuma el rol. X-Ray envía los datos de seguimiento a su cuenta.

El rol vinculado al servicio AWSServiceRoleForCloudWatchRUM tiene una política de IAM adjunta denominadaAmazonCloudWatchRUMServiceRolePolicy (Política de rol de servicio de Amazon CloudWatch RUM). Esta política le concede permiso a CloudWatch RUM para publicar datos de supervisión en otros servicios relevantes de AWS. Incluye permisos que permiten que RUM complete las siguientes acciones:

- `xray:PutTraceSegments`
- `cloudwatch:PutMetricData`

A continuación se detalla el contenido completo de la política AmazonCloudWatchRUMServiceRolePolicy.

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Effect": "Allow",
   "Action": [
    "xray:PutTraceSegments"
   ],
   "Resource": [
    "*"
   ]
  },
  {
   "Effect": "Allow",
   "Action": "cloudwatch:PutMetricData",
   "Resource": "*",
   "Condition": {
    "StringLike": {
     "cloudwatch:namespace": [
      "RUM/CustomMetrics/*",
```

```
        "AWS/RUM"
      ]
     }
    }
   }
  ]
}
```

Creación de un rol vinculado a un servicio para RUM

No es necesario crear un rol vinculado a un servicio de forma manual para CloudWatch RUM. La primera vez que crea un monitor de aplicaciones con el seguimiento de X-Ray habilitado o actualiza un monitor de aplicaciones para utilizar el seguimiento de X-Ray, RUM le crea un AWSServiceRoleForCloudWatchRUM.

Para obtener más información, consulte Crear un rol vinculado a un servicio en la Guía del usuario de IAM.

Edición de un rol vinculado a un servicio para RUM

CloudWatch RUM no le permite editar el rol AWSServiceRoleForCloudWatchRUM. Después de crear estos roles, no puede cambiar sus nombres, porque diversas entidades pueden hacer referencia a ellos. Sin embargo, puede editar la descripción de estos roles mediante IAM.

Edición de la descripción de un rol vinculado a un servicio (consola de IAM)

Puede utilizar la consola de IAM para editar la descripción de un rol vinculado a un servicio.

Para editar la descripción de un rol vinculado a un servicio (consola)

1. En el panel de navegación de la consola de IAM, elija Roles.
2. Seleccione el nombre del rol que desea modificar.
3. En el extremo derecho de Role description, seleccione Edit.
4. Escriba una nueva descripción en el cuadro y elija Save (Guardar).

Edición de la descripción de un rol vinculado a servicio (AWS CLI)

Puede utilizar comandos de IAM desde la AWS Command Line Interface para editar la descripción de un rol vinculado a un servicio.

Para cambiar la descripción de un rol vinculado a un servicio (AWS CLI)

1. (Opcional) Para ver la descripción actual de un rol, ejecute uno de los siguientes comandos:

   ```
   $ aws iam get-role --role-name role-name
   ```

   Utilice el nombre del rol, no el ARN, para hacer referencia a los roles con los comandos de AWS CLI. Por ejemplo, si un rol tiene el ARN `arn:aws:iam::123456789012:role/myrole`, debe referirse a él como **`myrole`**.
2. Para actualizar la descripción de un rol vinculado a un servicio, ejecute el siguiente comando:

   ```
   $ aws iam update-role-description --role-name role-name --description description
   ```

Edición de la descripción de un rol vinculado a un servicio (API de IAM)

Puede utilizar la API de IAM para editar la descripción de un rol vinculado a un servicio.

Para cambiar la descripción de un rol vinculado a un servicio (API)

1. (Opcional) Para ver la descripción actual de una función, ejecute el siguiente comando:

   GetRole
2. Para actualizar la descripción de una función, use el siguiente comando:

   UpdateRoleDescription

Eliminación de un rol vinculado a un servicio para RUM

Si ya no tiene monitores de aplicación con X-Ray habilitado, le recomendamos que elimine el rol AWSServiceRoleForCloudWatchRUM.

De esta forma no tiene una entidad no utilizada que no se supervise ni mantenga de forma activa. Sin embargo, debe limpiar el rol vinculado al servicio antes de eliminarlo.

Limpiar un rol vinculado a servicios

Antes de poder utilizar IAM para eliminar un rol vinculado a un servicio, primero debe confirmar que dicho rol no tiene sesiones activas y eliminar los recursos que utiliza.

Para comprobar si el rol vinculado a un servicio tiene una sesión activa en la consola de IAM

1. Abra la consola de IAM en https://console.aws.amazon.com/iam/.
2. Seleccione Roles en el panel de navegación. Elija el nombre (no el casillero) del rol AWSServiceRoleForCloudWatchRUM.
3. En la página Summary del rol seleccionado, elija Access Advisor y revise la actividad reciente del rol vinculado al servicio.

> **Note**
>
> Si no está seguro acerca de si RUM utiliza el rol AWSServiceRoleForCloudWatchRUM, intente eliminar el rol. Si el servicio está utilizando el rol, este no podrá eliminarse y podrá ver las regiones en las que se está utilizando. Si el rol se está utilizando, debe esperar que la sesión finalice para poder eliminarlo. No se puede revocar la sesión de un rol vinculado a servicios.

Eliminación de un rol vinculado a un servicio (consola de IAM)

Puede utilizar la consola de IAM para eliminar un rol vinculado a un servicio.

Para eliminar un rol vinculado a un servicio (consola)

1. Abra la consola de IAM en https://console.aws.amazon.com/iam/.
2. Seleccione Roles en el panel de navegación. Seleccione la casilla junto al nombre del rol que desea eliminar, no el nombre ni la fila.
3. En Role actions, elija Delete role.
4. En el cuadro de diálogo de confirmación, revise los datos del último acceso al servicio, que muestra cuándo cada uno de los roles seleccionados tuvo acceso a un servicio de AWS por última vez. Esto lo ayuda a confirmar si el rol está actualmente activo. Para continuar, elija Yes, Delete.
5. Consulte las notificaciones de la consola de IAM para supervisar el progreso de la eliminación del rol vinculado al servicio. Debido a que el proceso de eliminación del rol vinculado al servicio de IAM es asíncrono, la tarea de eliminación puede realizarse correcta o incorrectamente después de que envía la solicitud de eliminación. Si la tarea no se realiza correctamente, elija View details o View Resources desde las notificaciones para obtener información acerca de por

qué no se pudo eliminar el rol. Si la eliminación no pudo producirse porque hay recursos en el servicio que está utilizando el rol, entonces el motivo del error incluye una lista de recursos.

Eliminar un rol vinculado a un servicio (AWS CLI)

Puede utilizar los comandos de IAM desde la AWS Command Line Interface para eliminar un rol vinculado a un servicio.

Para eliminar un rol vinculado a un servicio (AWS CLI)

1. Como los roles vinculados a servicios no se puede eliminar si están en uso o tienen recursos asociados, debe enviar una solicitud de eliminación. Esta solicitud puede denegarse si no se cumplen estas condiciones. Debe apuntar el valor `deletion-task-id` de la respuesta para comprobar el estado de la tarea de eliminación. Escriba el siguiente comando para enviar una solicitud de eliminación de un rol vinculado a un servicio:

   ```
   $ aws iam delete-service-linked-role --role-name service-linked-role-name
   ```

2. Escriba el siguiente comando para comprobar el estado de la tarea de eliminación:

   ```
   $ aws iam get-service-linked-role-deletion-status --deletion-task-id deletion-task-id
   ```

   El estado de la tarea de eliminación puede ser `NOT_STARTED`, `IN_PROGRESS`, `SUCCEEDED` o `FAILED`. Si ocurre un error durante la eliminación, la llamada devuelve el motivo del error para que pueda resolver el problema.

Eliminación de un rol vinculado a un servicio (API de IAM)

Puede utilizar la API de IAM para eliminar un rol vinculado a un servicio.

Para eliminar un rol vinculado a un servicio (API)

1. Para enviar una solicitud de eliminación de un rol vinculado a un servicio, realice una llamada a DeleteServiceLinkedRole. En la solicitud, especifique el nombre de rol que desea eliminar.

   Como los roles vinculados a servicios no se puede eliminar si están en uso o tienen recursos asociados, debe enviar una solicitud de eliminación. Esta solicitud puede denegarse si no se

cumplen estas condiciones. Debe apuntar el valor `DeletionTaskId` de la respuesta para comprobar el estado de la tarea de eliminación.

2. Para comprobar el estado de la tarea de eliminación, realice una llamada a GetServiceLinkedRoleDeletionStatus. En la solicitud, especifique el valor de `DeletionTaskId`.

   El estado de la tarea de eliminación puede ser `NOT_STARTED`, `IN_PROGRESS`, `SUCCEEDED` o `FAILED`. Si ocurre un error durante la eliminación, la llamada devuelve el motivo del error para que pueda resolver el problema.

Uso de roles vinculados a un servicio para CloudWatch Application Insights

CloudWatch Application Insights utiliza roles vinculados a un servicio (roles de AWS Identity and Access Management [IAM]). Un rol vinculado a un servicio es un tipo único de rol de IAM que está vinculado directamente a CloudWatch Application Insights. Los roles vinculados a un servicio están predefinidos por CloudWatch Application Insights e incluyen todos los permisos que el servicio requiere para llamar a otros servicios de AWS en su nombre.

Un rol vinculado a un servicio simplifica la configuración de CloudWatch Application Insights debido a que ya no tendrá que agregar manualmente los permisos necesarios. CloudWatch Application Insights define los permisos de los roles vinculados a un servicio y, a menos que esté definido de otra manera, solo CloudWatch Application Insights puede asumir los roles. Los permisos definidos incluyen las políticas de confianza y de permisos y que la política de permisos no se pueda adjuntar a ninguna otra entidad de IAM.

Para obtener información acerca de otros servicios que admiten roles vinculados a servicios, consulte Servicios de AWS que funcionan con IAM y busque los servicios que muestran Yes (Sí) en la columna Service Linked Role (Rol vinculado a servicios). Elija el vínculo Yes (Sí) para ver la documentación acerca del rol vinculado a un servicio en cuestión.

Permisos de roles vinculados a un servicio para CloudWatch Application Insights

CloudWatch Application Insights utiliza el rol vinculado a un servicio denominado AWSServiceRoleForApplicationInsights. Application Insights utiliza este rol para realizar operaciones, como analizar los grupos de recursos del cliente, crear pilas de CloudFormation para crear alarmas en las métricas y configurar el agente de CloudWatch en instancias EC2. Este rol vinculado a servicios cuenta con una política de IAM que se llama `CloudwatchApplicationInsightsServiceLinkedRolePolicy`. Para obtener actualizaciones

de esta política, consulte Actualizaciones de Application Insights para las políticas administradas de AWS.

La política de permisos del rol permite que CloudWatch Application Insights realice las siguientes acciones en los recursos.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "cloudwatch:DescribeAlarmHistory",
        "cloudwatch:DescribeAlarms",
        "cloudwatch:GetMetricData",
        "cloudwatch:ListMetrics",
        "cloudwatch:PutMetricAlarm",
        "cloudwatch:DeleteAlarms",
        "cloudwatch:PutAnomalyDetector",
        "cloudwatch:DeleteAnomalyDetector",
        "cloudwatch:DescribeAnomalyDetectors"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "logs:FilterLogEvents",
        "logs:GetLogEvents",
        "logs:DescribeLogStreams",
        "logs:DescribeLogGroups"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "events:DescribeRule"
      ],
```

```
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "cloudFormation:CreateStack",
        "cloudFormation:UpdateStack",
        "cloudFormation:DeleteStack",
        "cloudFormation:DescribeStackResources"
      ],
      "Resource": [
        "arn:aws:cloudformation:*:*:stack/ApplicationInsights-*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "cloudFormation:DescribeStacks",
        "cloudFormation:ListStackResources",
        "cloudFormation:ListStacks"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "tag:GetResources"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "resource-groups:ListGroupResources",
        "resource-groups:GetGroupQuery",
        "resource-groups:GetGroup"
      ],
      "Resource": [
```

```
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "resource-groups:CreateGroup",
        "resource-groups:DeleteGroup"
      ],
      "Resource": [
        "arn:aws:resource-groups:*:*:group/ApplicationInsights-*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "elasticloadbalancing:DescribeLoadBalancers",
        "elasticloadbalancing:DescribeTargetGroups",
        "elasticloadbalancing:DescribeTargetHealth"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "autoscaling:DescribeAutoScalingGroups"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "ssm:PutParameter",
        "ssm:DeleteParameter",
        "ssm:AddTagsToResource",
        "ssm:RemoveTagsFromResource",
        "ssm:GetParameters"
      ],
      "Resource": "arn:aws:ssm:*:*:parameter/AmazonCloudWatch-ApplicationInsights-*"
    },
```

```
    {
      "Effect": "Allow",
      "Action": [
        "ssm:CreateAssociation",
        "ssm:UpdateAssociation",
        "ssm:DeleteAssociation",
        "ssm:DescribeAssociation"
      ],
      "Resource": [
        "arn:aws:ec2:*:*:instance/*",
        "arn:aws:ssm:*:*:association/*",
        "arn:aws:ssm:*:*:managed-instance/*",
        "arn:aws:ssm:*:*:document/AWSEC2-
ApplicationInsightsCloudwatchAgentInstallAndConfigure",
        "arn:aws:ssm:*:*:document/AWS-ConfigureAWSPackage",
        "arn:aws:ssm:*:*:document/AmazonCloudWatch-ManageAgent"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "ssm:GetOpsItem",
        "ssm:CreateOpsItem",
        "ssm:DescribeOpsItems",
        "ssm:UpdateOpsItem",
        "ssm:DescribeInstanceInformation"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "ssm:AddTagsToResource"
      ],
      "Resource": "arn:aws:ssm:*:*:opsitem/*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "ssm:ListCommandInvocations",
        "ssm:GetCommandInvocation"
      ],
```

```
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": "ssm:SendCommand",
      "Resource": [
        "arn:aws:ec2:*:*:instance/*",
        "arn:aws:ssm:*:*:document/AWSEC2-CheckPerformanceCounterSets",
        "arn:aws:ssm:*:*:document/AWS-ConfigureAWSPackage",
        "arn:aws:ssm:*:*:document/AWSEC2-DetectWorkload",
        "arn:aws:ssm:*:*:document/AmazonCloudWatch-ManageAgent"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeInstances",
        "ec2:DescribeVolumes",
        "ec2:DescribeVolumeStatus",
        "ec2:DescribeVpcs",
        "ec2:DescribeVpcAttribute",
        "ec2:DescribeNatGateways"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "rds:DescribeDBInstances",
        "rds:DescribeDBClusters"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "lambda:ListFunctions",
        "lambda:GetFunctionConfiguration",
```

```
        "lambda:ListEventSourceMappings"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "events:PutRule",
        "events:PutTargets",
        "events:RemoveTargets",
        "events:DeleteRule"
      ],
      "Resource": [
        "arn:aws:events:*:*:rule/AmazonCloudWatch-ApplicationInsights-*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "xray:GetServiceGraph",
        "xray:GetTraceSummaries",
        "xray:GetTimeSeriesServiceStatistics",
        "xray:GetTraceGraph"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "dynamodb:ListTables",
        "dynamodb:DescribeTable",
        "dynamodb:DescribeContributorInsights",
        "dynamodb:DescribeTimeToLive"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
```

```
      "Action": [
        "application-autoscaling:DescribeScalableTargets"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:ListAllMyBuckets",
        "s3:GetMetricsConfiguration",
        "s3:GetReplicationConfiguration"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "states:ListStateMachines",
        "states:DescribeExecution",
        "states:DescribeStateMachine",
        "states:GetExecutionHistory"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "apigateway:GET"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "ecs:DescribeClusters",
        "ecs:DescribeContainerInstances",
```

```
        "ecs:DescribeServices",
        "ecs:DescribeTaskDefinition",
        "ecs:DescribeTasks",
        "ecs:DescribeTaskSets",
        "ecs:ListClusters",
        "ecs:ListContainerInstances",
        "ecs:ListServices",
        "ecs:ListTasks"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "ecs:UpdateClusterSettings"
      ],
      "Resource": [
        "arn:aws:ecs:*:*:cluster/*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "eks:DescribeCluster",
        "eks:DescribeFargateProfile",
        "eks:DescribeNodegroup",
        "eks:ListClusters",
        "eks:ListFargateProfiles",
        "eks:ListNodegroups",
        "fsx:DescribeFileSystems",
        "fsx:DescribeVolumes"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "sns:GetSubscriptionAttributes",
        "sns:GetTopicAttributes",
        "sns:GetSMSAttributes",
```

```
        "sns:ListSubscriptionsByTopic",
        "sns:ListTopics"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "sqs:ListQueues"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "logs:DeleteSubscriptionFilter"
      ],
      "Resource": [
        "arn:aws:logs:*:*:log-group:*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "logs:PutSubscriptionFilter"
      ],
      "Resource": [
        "arn:aws:logs:*:*:log-group:*",
        "arn:aws:logs:*:*:destination:AmazonCloudWatch-ApplicationInsights-
LogIngestionDestination*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "elasticfilesystem:DescribeFileSystems"
      ],
      "Resource": [
        "*"
      ]
    },
    {
```

```
      "Effect": "Allow",
      "Action": [
        "route53:GetHostedZone",
        "route53:GetHealthCheck",
        "route53:ListHostedZones",
        "route53:ListHealthChecks",
        "route53:ListQueryLoggingConfigs"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "route53resolver:ListFirewallRuleGroupAssociations",
        "route53resolver:GetFirewallRuleGroup",
        "route53resolver:ListFirewallRuleGroups",
        "route53resolver:ListResolverEndpoints",
        "route53resolver:GetResolverQueryLogConfig",
        "route53resolver:ListResolverQueryLogConfigs",
        "route53resolver:ListResolverQueryLogConfigAssociations",
        "route53resolver:GetResolverEndpoint",
        "route53resolver:GetFirewallRuleGroupAssociation"
      ],
      "Resource": [
        "*"
      ]
    }
  ]
}
```

Debe configurar permisos para permitir a una entidad de IAM (como un usuario, grupo o rol) crear, editar o eliminar un rol vinculado a servicios. Para obtener más información, consulte Permisos de roles vinculados a servicios en la Guía del usuario de IAM.

Creación de un rol vinculado a un servicio para CloudWatch Application Insights

No necesita crear manualmente un rol vinculado a servicios. Al crear una nueva aplicación de Application Insights en la consola, CloudWatch Application Insights le crea el rol vinculado a un servicio. AWS Management Console

Si elimina este rol vinculado al servicio y desea crearlo de nuevo, puede utilizar el mismo proceso para volver a crear el rol en su cuenta. Al crear una nueva aplicación de Application Insights, CloudWatch Application Insights vuelve a crear el rol vinculado a un servicio.

Modificación de un rol vinculado a un servicio para CloudWatch Application Insights

CloudWatch Application Insights no le permite editar el rol vinculado a un servicio AWSServiceRoleForApplicationInsights. Después de crear un rol vinculado a servicios, no puede cambiarle el nombre, ya que varias entidades pueden hacer referencia al mismo. Sin embargo, puede editar la descripción del rol mediante IAM. Para obtener más información, consulte Editar un rol vinculado a servicios en la Guía del usuario de IAM..

Eliminación de un rol vinculado a un servicio para CloudWatch Application Insights

Si ya no necesita utilizar una característica o servicio que requiere un rol vinculado a servicios, recomendamos que elimine dicho rol. De esta forma, evitará tener una entidad no utilizada que no se monitorice ni mantenga de forma activa. Sin embargo, debe eliminar todas las aplicaciones de Application Insights antes de poder eliminar manualmente el rol.

> **Note**
>
> Si el servicio de CloudWatch Application Insights está utilizando el rol cuando intenta eliminar los recursos, es posible que no se pueda borrar. En tal caso, espere unos minutos e intente de nuevo la operación.

Para eliminar recursos de CloudWatch Application Insights utilizados por AWSServiceRoleForApplicationInsights

- Eliminación de todas las aplicaciones de CloudWatch Application Insights. Para obtener más información, consulte ‘Eliminación de aplicaciones’ en la Guía del usuario de CloudWatch Application Insights.

Para eliminar manualmente el rol vinculado a servicios mediante IAM

Utilice la consola de IAM, AWS CLI o la API de AWS para eliminar el rol vinculado a un servicio AWSServiceRoleForApplicationInsights (Rol vinculado a un servicio de AWS para Application Insights). Para obtener más información, consulte Eliminación de un rol vinculado a servicios en la Guía del usuario de IAM.

Regiones admitidas para roles vinculados a un servicio de CloudWatch Application Insights

CloudWatch Application Insights admite el uso de roles vinculados a un servicio en todas las Regiones de AWS en las que el servicio está disponible. Para obtener más información, consulte CloudWatch Application Insights Regions and Endpoints (Regiones y puntos de enlace de CloudWatch Application Insights).

Políticas administradas de AWS para Información de aplicaciones de Amazon CloudWatch

Una política administrada de AWS es una política independiente que AWS crea y administra. Las políticas administradas de AWS se diseñan para ofrecer permisos para muchos casos de uso comunes, por lo que puede empezar a asignar permisos a los usuarios, grupos y roles.

Tenga presente que es posible que las políticas administradas de AWS no concedan permisos de privilegio mínimo para los casos de uso concretos, ya que están disponibles para que las utilicen todos los clientes de AWS. Se recomienda definir políticas administradas por el cliente para los casos de uso a fin de reducir aún más los permisos.

No puede cambiar los permisos definidos en las políticas administradas por AWS. Si AWS actualiza los permisos definidos en un política administrada de AWS, la actualización afecta a todas las identidades de entidades principales (usuarios, grupos y roles) a las que está adjunta la política. Lo más probable es que AWS actualice una política administrada de AWS cuando se lance un nuevo Servicio de AWS o las operaciones de la API nuevas estén disponibles para los servicios existentes.

Para obtener más información, consulte Políticas administradas de AWS en la Guía del usuario de IAM.

Política administrada de AWS: CloudWatchApplicationInsightsFullAccess

Puede adjuntar la política `CloudWatchApplicationInsightsFullAccess` a las identidades de IAM.

Esta política otorga permisos administrativos que brindan acceso completo a la funcionalidad de Application Insights.

Detalles sobre los permisos

Esta política incluye los siguientes permisos.

- `applicationinsights`: permite el acceso completo a la funcionalidad de Application Insights.
- `iam`: permite que Application Insights cree el rol vinculado a un servicio AWSServiceRoleForApplicationInsights. Esto es necesario para que Application Insights pueda realizar operaciones, como analizar los grupos de recursos de un cliente, crear pilas de CloudFormation para crear alarmas en las métricas y configurar el agente de CloudWatch en instancias EC2. Para obtener más información, consulte Uso de roles vinculados a un servicio para CloudWatch Application Insights.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "applicationinsights:*",
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeInstances",
        "ec2:DescribeVolumes",
        "rds:DescribeDBInstances",
        "rds:DescribeDBClusters",
        "sqs:ListQueues",
        "elasticloadbalancing:DescribeLoadBalancers",
        "elasticloadbalancing:DescribeTargetGroups",
        "elasticloadbalancing:DescribeTargetHealth",
        "autoscaling:DescribeAutoScalingGroups",
        "lambda:ListFunctions",
        "dynamodb:ListTables",
```

```
        "s3:ListAllMyBuckets",
        "sns:ListTopics",
        "states:ListStateMachines",
        "apigateway:GET",
        "ecs:ListClusters",
        "ecs:DescribeTaskDefinition",
        "ecs:ListServices",
        "ecs:ListTasks",
        "eks:ListClusters",
        "eks:ListNodegroups",
        "fsx:DescribeFileSystems",
        "logs:DescribeLogGroups",
        "elasticfilesystem:DescribeFileSystems"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "iam:CreateServiceLinkedRole"
      ],
      "Resource": [
        "arn:aws:iam::*:role/aws-service-role/application-insights.amazonaws.com/
AWSServiceRoleForApplicationInsights"
      ],
      "Condition": {
        "StringEquals": {
          "iam:AWSServiceName": "application-insights.amazonaws.com"
        }
      }
    }
  ]
}
```

Política administrada de AWS: CloudWatchApplicationInsightsReadOnlyAccess

Puede adjuntar la política `CloudWatchApplicationInsightsReadOnlyAccess` a las identidades de IAM.

Esta política otorga permisos administrativos que brindan acceso de solo lectura a todas las funcionalidades de Application Insights.

Detalles sobre los permisos

Esta política incluye los siguientes permisos.

- `applicationinsights`: permite el acceso de solo lectura a la funcionalidad de Application Insights.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "applicationinsights:Describe*",
                "applicationinsights:List*"
            ],
            "Resource": "*"
        }
    ]
}
```

Política administrada de AWS: CloudwatchApplicationInsightsServiceLinkedRolePolicy

No puede adjuntar CloudwatchApplicationInsightsServiceLinkedRolePolicy a las entidades de IAM. Esta política está adjuntada a un rol vinculado a un servicio que permite que Application Insights monitoree los recursos de los clientes. Para obtener más información, consulte Uso de roles vinculados a un servicio para CloudWatch Application Insights.

Actualizaciones de Application Insights para las políticas administradas de AWS

Es posible consultar los detalles sobre las actualizaciones de las políticas administradas de AWS para Application Insights debido a que este servicio comenzó a realizar el seguimiento de estos cambios. Para obtener alertas automáticas sobre cambios en esta página, suscríbase a la fuente RSS en la página Document history (Historial de documentos) de Application Insights.

Cambio	Descripción	Fecha
CloudwatchApplicationInsightsServiceLinkedRolePolicy: Actualización de una política existente	Application Insights ha agregado nuevos permisos para enumerar las pilas de CloudFormation. Estos permisos son necesarios para que Información de aplicaciones de Amazon CloudWatch analice y monitoree los recursos de AWS anidados en la pila de CloudFormation.	24 de abril de 2023
CloudwatchApplicationInsightsServiceLinkedRolePolicy: actualización de una política existente	Application Insights agregó nuevos permisos para obtener una lista de los recursos de Amazon VPC y Route 53. Estos permisos son necesarios para que Información de aplicaciones de Amazon CloudWatch configure el monitoreo de la red de la práctica recomendada con Amazon CloudWatch.	23 de enero de 2023
CloudwatchApplicationInsightsServiceLinkedRolePolicy: actualización de una política existente	Application Insights agregó nuevos permisos para obtener los resultados de la invocación de los comandos SSM. Estos permisos son necesarios para que Información de aplicaciones de Amazon CloudWatch detecte y monitoree automáticamente	19 de diciembre de 2022

Cambio	Descripción	Fecha
	las cargas de trabajo que se ejecutan en las instancias de Amazon EC2.	
CloudwatchApplicationInsightsServiceLinkedRolePolicy: actualización de una política existente	Application Insights agregó nuevos permisos para describir los recursos de Amazon VPC y Route 53. Estos permisos son necesarios para que Información de aplicaciones de Amazon CloudWatch lea las configuraciones de los recursos de los clientes de Amazon VPC y Route 53, y para ayudar a los clientes a configurar de manera automática las prácticas recomendadas de monitoreo con Amazon CloudWatch.	19 de diciembre de 2022
CloudwatchApplicationInsightsServiceLinkedRolePolicy: actualización de una política existente	Application Insights agregó nuevos permisos para describir los recursos de EFS. Estos permisos son necesarios para que Información de aplicaciones de Amazon CloudWatch lea las configuraciones de recursos de Amazon EFS de los clientes y los ayude a configurar automáticamente las prácticas recomendadas de monitoreo para EFS con CloudWatch.	3 de octubre de 2022

Cambio	Descripción	Fecha
CloudwatchApplicationInsightsServiceLinkedRolePolicy: actualización de una política existente	Application Insights agregó nuevos permisos para describir el sistema de archivos EFS. Estos permisos son necesarios para que Información de aplicaciones de Amazon CloudWatch cree aplicaciones basadas en cuentas mediante consultas a todos los recursos compatibles de una cuenta.	3 de octubre de 2022
CloudwatchApplicationInsightsServiceLinkedRolePolicy: actualización de una política existente	Application Insights agregó nuevos permisos para recuperar la información sobre los recursos de FSx. Estos permisos son necesarios para que Información de aplicaciones de Amazon CloudWatch monitoree las cargas de trabajo al recuperar la información suficiente sobre los volúmenes subyacentes de FSx.	12 de septiembre de 2022

Cambio	Descripción	Fecha
Política administrada de AWS: CloudWatchApplicationInsightsFullAccess: actualización de una política actual	Application Insights agregó un nuevo permiso para describir los grupos de registros. Estos permisos son necesarios para Amazon CloudWatch Application Insights a fin de garantizar que los permisos correctos para supervisar los grupos de registros estén en una cuenta cuando se crea una nueva aplicación.	24 de enero de 2022
CloudwatchApplicationInsightsServiceLinkedRolePolicy: actualización de una política existente	Application Insights agregó nuevos permisos para crear y eliminar los filtros de suscripción de CloudWatch Log. Estos permisos son necesarios para que Amazon CloudWatch Application Insights cree filtros de suscripción para facilitar la supervisión de los registros de los recursos dentro de las aplicaciones configuradas.	24 de enero de 2022

Cambio	Descripción	Fecha
CloudwatchApplicationInsightsServiceLinkedRolePolicy: actualización de una política existente	Application Insights agregó nuevos permisos para describir los grupos de destino y el estado de destino para Elastic Load Balancers. Estos permisos son necesarios para que Amazon CloudWatch Application Insights cree aplicaciones basadas en cuentas mediante consultas a todos los recursos compatibles de una cuenta.	4 de noviembre de 2021
CloudwatchApplicationInsightsServiceLinkedRolePolicy: actualización de una política existente	Application Insights agregó nuevos permisos para ejecutar el documento de SSM de `AmazonCloudWatch-ManageAgent` en instancias de Amazon EC2. Estos permisos son necesarios para que Amazon CloudWatch Application Insights limpie los archivos de configuración del agente de CloudWatch creados por Application Insights.	30 de septiembre de 2021

Cambio	Descripción	Fecha
CloudwatchApplicationInsightsServiceLinkedRolePolicy: actualización de una política existente	Application Insights agregó nuevos permisos para admitir el monitoreo de aplicaciones basado en cuentas para incorporar y monitorear todos los recursos compatibles de su cuenta. Estos permisos son necesarios para que Amazon CloudWatch Application Insights consulte, etiquete recursos y cree grupos para estos recursos. Application Insights agregó nuevos permisos para admitir el monitoreo de temas de SNS. Estos permisos son necesarios para que Amazon CloudWatch Application Insights recopile metadatos de los recursos de SNS para configurar el monitoreo de los temas de SNS.	15 de septiembre de 2021

Cambio	Descripción	Fecha
Política administrada de AWS: CloudWatchApplicationInsightsFullAccess: actualización de una política actual	Application Insights agregó nuevos permisos para describir y enumerar los recursos compatibles. Estos permisos son necesarios para que Amazon CloudWatch Application Insights cree aplicaciones basadas en cuentas mediante consultas a todos los recursos compatibles de una cuenta.	15 de septiembre de 2021
CloudwatchApplicationInsightsServiceLinkedRolePolicy: actualización de una política existente	Application Insights agregó nuevos permisos para describir los recursos de FSx. Estos permisos son necesarios para que Amazon CloudWatch Application Insights lea las configuraciones de recursos de FSx de los clientes y los ayude a configurar automáticamente el monitoreo de FSx de la práctica recomendada con CloudWatch.	31 de agosto de 2021

Cambio	Descripción	Fecha
CloudwatchApplicationInsightsServiceLinkedRolePolicy: actualización de una política existente	Application Insights agregó nuevos permisos para describir y enumerar los recursos de los servicios ECS y EKS. Este permiso es necesario para que Amazon CloudWatch Application Insights lea la configuración de los recursos del contenedor del cliente y los ayude a configurar automáticamente el monitoreo de contenedores de la práctica recomendada con CloudWatch.	18 de mayo de 2021
CloudwatchApplicationInsightsServiceLinkedRolePolicy: actualización de una política existente	Application Insights agregó nuevos permisos para permitir que OpsCenter etiquete OpsItems mediante la acción `ssm:AddTagsToResource` en los recursos con el tipo de recurso `opsitem`.. OpsCenter requiere este permiso. Amazon CloudWatch Application Insights crea OpsItems para que el cliente pueda resolver problemas mediante AWSSSM OpsCenter.	13 de abril de 2021

Cambio	Descripción	Fecha
Aplicación Insights ha comenzado a realizar seguimientos de los cambios	Application Insights ha comenzado a realizar seguimientos de los cambios para las políticas administr adas de AWS.	13 de abril de 2021

Referencia de permisos de Amazon CloudWatch

En la siguiente tabla figuran las operaciones de la API de CloudWatch y las acciones correspondientes para las que usted puede conceder permisos para realizar la acción. Las acciones se especifican en el campo `Action` de la política, y el valor del recurso se especifica como un carácter comodín (*) en el campo `Resource` de la política.

Puede utilizar claves de condición generales de AWS en las políticas de CloudWatch para expresar condiciones. Para ver una lista completa de claves generales de AWS, consulte Claves de contexto de condición de IAM y globales de AWS en la Guía del usuario de IAM.

> **Note**
>
> Para especificar una acción, use el prefijo `cloudwatch:` seguido del nombre de operación de API. Por ejemplo: `cloudwatch:GetMetricData`, `cloudwatch:ListMetrics` o `cloudwatch:*` (para todas las acciones de CloudWatch).

Temas

- Operaciones de la API y permisos necesarios de CloudWatch para acciones
- Operaciones de la API de CloudWatch Contributor Insights y permisos necesarios para realizar acciones
- Operaciones de la API de CloudWatch Events y permisos necesarios para realizar acciones
- Operaciones de la API de CloudWatch Logs y permisos necesarios para realizar acciones
- Operaciones de la API de Amazon EC2 y permisos necesarios para realizar acciones
- Operaciones de la API de Amazon EC2 Auto Scaling y permisos necesarios para realizar acciones

Operaciones de la API y permisos necesarios de CloudWatch para acciones

Operaciones de la API de CloudWatch	Permisos necesarios (acciones de API)
DeleteAlarms	`cloudwatch:DeleteAlarms` Necesario para eliminar una alarma.
DeleteDashboards	`cloudwatch:DeleteDashboards` Necesario para eliminar un panel.
DeleteMetricStream	`cloudwatch:DeleteMetricStream` Necesario para eliminar un flujo métrico.
DescribeAlarmHistory	`cloudwatch:DescribeAlarmHistory` Necesario para ver el historial de alarmas. Para recuperar información acerca de las alarmas compuestas, el permiso `cloudwatch:DescribeAlarmHistory` debe tener alcance `*`. No puede devolver información sobre las alarmas compuestas si el permiso `cloudwatch:DescribeAlarmHistory` tiene un alcance más limitado.
DescribeAlarms	`cloudwatch:DescribeAlarms` Se necesita para recuperar información sobre las alarmas. Para recuperar información acerca de las alarmas compuestas, el permiso `cloudwatch:DescribeAlarms` debe tener alcance `*`. No puede devolver información sobre las

Operaciones de la API de CloudWatch	Permisos necesarios (acciones de API)
	alarmas compuestas si el permiso `cloudwatch:DescribeAlarms` tiene un alcance más limitado.
DescribeAlarmsForMetric	`cloudwatch:DescribeAlarmsForMetric` Necesario para ver alarmas para una métrica.
DisableAlarmActions	`cloudwatch:DisableAlarmActions` Necesario para desactivar una acción de alarma.
EnableAlarmActions	`cloudwatch:EnableAlarmActions` Necesario para habilitar una acción de alarma.
GetDashboard	`cloudwatch:GetDashboard` Necesario para mostrar datos acerca de paneles existentes.
GetMetricData	`cloudwatch:GetMetricData` Es necesario para representar los datos métricos en un gráfico de la consola de CloudWatch a fin de recuperar grandes lotes de datos métricos y llevar a cabo cálculos métricos en función de dichos datos.

Operaciones de la API de CloudWatch	Permisos necesarios (acciones de API)
GetMetricStatistics	`cloudwatch:GetMetricStatistics` Es necesario para visualizar gráficos en otras partes de la consola de CloudWatch y en widgets de paneles.
GetMetricStream	`cloudwatch:GetMetricStream` Es necesario para ver información acerca de un flujo métrico.
GetMetricWidgetImage	`cloudwatch:GetMetricWidgetImage` Es necesario para recuperar un gráfico de instantáneas de una o varias métricas de CloudWatch como imagen de mapa de bits.
ListDashboards	`cloudwatch:ListDashboards` Es necesario para visualizar la lista de paneles de CloudWatch en su cuenta.
ListMetrics	`cloudwatch:ListMetrics` Es necesario para visualizar o buscar nombres de métricas dentro de la consola de CloudWatch y en la CLI. Necesario para seleccionar métricas en widgets de paneles.
ListMetricStreams	`cloudwatch:ListMetricStreams` Es necesario para visualizar o buscar la lista de flujos métricos en la cuenta.

Operaciones de la API de CloudWatch	Permisos necesarios (acciones de API)
PutCompositeAlarm	`cloudwatch:PutCompositeAlarm` Es necesario para crear una alarma compuesta Para crear una alarma compuesta, el permiso `cloudwatch:PutCompositeAlarm` debe tener un alcance `*`. No puede devolver información sobre alarmas compuestas si el permiso `cloudwatch:PutCompositeAlarm` tiene un alcance más limitado.
PutDashboard	`cloudwatch:PutDashboard` Necesario para crear un panel o actualizar un panel existente.
PutMetricAlarm	`cloudwatch:PutMetricAlarm` Necesario para crear o actualizar una alarma.
PutMetricData	`cloudwatch:PutMetricData` Necesario para crear métricas.
PutMetricStream	`cloudwatch:PutMetricStream` Necesario para crear flujos métricos.
SetAlarmState	`cloudwatch:SetAlarmState` Necesario para configurar manualmente el estado de una alarma.

Operaciones de la API de CloudWatch	Permisos necesarios (acciones de API)
StartMetricStreams	`cloudwatch:StartMetricStreams` Necesario para comenzar el flujo de las métricas en un flujo métrico.
StopMetricStreams	`cloudwatch:StopMetricStreams` Necesario para detener temporalmente el flujo de las métricas en un flujo métrico.
TagResource	`cloudwatch:TagResource` Necesario para agregar o actualizar etiquetas en recursos de CloudWatch, como alarmas y reglas de Contributor Insights.
UntagResource	`cloudwatch:UntagResource` Necesario para eliminar etiquetas de los recursos de CloudWatch.

Operaciones de la API de CloudWatch Contributor Insights y permisos necesarios para realizar acciones

Important

Cuando le concede el permiso `cloudwatch:PutInsightRule` a un usuario, de forma predeterminada, ese usuario puede crear una regla que evalúe cualquier grupo de registros en CloudWatch Logs. Puede agregar condiciones de políticas de IAM que limiten estos permisos para que un usuario incluya y excluya grupos de registros específicos. Para obtener más información, consulte Uso de claves de condición para limitar el acceso de los usuarios de Contributor Insights a los grupos de registro.

Operaciones de la API de CloudWatch Contributor Insights	Permisos necesarios (acciones de API)
DeleteInsightRules	`cloudwatch:DeleteInsightRules` Necesario para eliminar reglas de Contributor Insights.
DescribeInsightRules	`cloudwatch:DescribeInsightRules` Necesario para visualizar las reglas de Contributor Insights en la cuenta.
EnableInsightRules	`cloudwatch:EnableInsightRules` Necesario para habilitar las reglas de Contribut or Insights.
GetInsightRuleReport	`cloudwatch:GetInsightRuleReport` Necesario para recuperar datos de series temporales y otras estadísticas que recopilan las reglas de Contributor Insights.
PutInsightRule	`cloudwatch:PutInsightRule` Necesario para crear reglas de Contributor Insights. Consulte la nota Importante ubicada al principio de esta tabla.

Operaciones de la API de CloudWatch Events y permisos necesarios para realizar acciones

Operaciones de la API de CloudWatch Events	Permisos necesarios (acciones de API)

Operaciones de la API de CloudWatch Events	Permisos necesarios (acciones de API)
DeleteRule	`events:DeleteRule` Necesario para eliminar una regla.
DescribeRule	`events:DescribeRule` Necesario para mostrar detalles acerca de una regla.
DisableRule	`events:DisableRule` Necesario para deshabilitar una regla.
EnableRule	`events:EnableRule` Necesario para habilitar una regla.
ListRuleNamesByTarget	`events:ListRuleNamesByTarget` Necesario para enumerar reglas asociadas a un destino.
ListRules	`events:ListRules` Necesario para enumerar todas las reglas de su cuenta.
ListTargetsByRule	`events:ListTargetsByRule` Necesario para enumerar todos los destinos asociados a una regla.

Operaciones de la API de CloudWatch Events	Permisos necesarios (acciones de API)
PutEvents	`events:PutEvents` Necesario para agregar eventos personalizados que se pueden asignar a reglas.
PutRule	`events:PutRule` Necesario para crear o actualizar una regla.
PutTargets	`events:PutTargets` Necesario para añadir destinos a una regla.
RemoveTargets	`events:RemoveTargets` Necesario para eliminar un destino de una regla.
TestEventPattern	`events:TestEventPattern` Necesario para probar un patrón de evento con respecto a un evento dado.

Operaciones de la API de CloudWatch Logs y permisos necesarios para realizar acciones

Operaciones de la API de Registros de CloudWatch	Permisos necesarios (acciones de API)
CancelExportTask	`logs:CancelExportTask`

Operaciones de la API de Registros de CloudWatch	Permisos necesarios (acciones de API)
	Necesario para cancelar una tarea de exportación en ejecución o pendiente.
CreateExportTask	`logs:CreateExportTask` Necesario para exportar datos desde un grupo de registros a un bucket de Amazon S3.
CreateLogGroup	`logs:CreateLogGroup` Necesario para crear un nuevo grupo de registros.
CreateLogStream	`logs:CreateLogStream` Necesario para crear un nuevo flujo de registros en un grupo de registros.
DeleteDestination	`logs:DeleteDestination` Necesario para eliminar un destino de registro y deshabilita los filtros de suscripción al mismo.
DeleteLogGroup	`logs:DeleteLogGroup` Necesario para eliminar un grupo de registros y todos los eventos de registro asociados.
DeleteLogStream	`logs:DeleteLogStream` Necesario para eliminar un flujo de registros y todos los eventos de registro asociados.

Operaciones de la API de Registros de CloudWatch	Permisos necesarios (acciones de API)
DeleteMetricFilter	`logs:DeleteMetricFilter` Necesario para eliminar un filtro de métricas asociado con un grupo de registros.
DeleteQueryDefinition	`logs:DeleteQueryDefinition` Necesario para eliminar una definición de consulta guardada en Información de registros de CloudWatch.
DeleteResourcePolicy	`logs:DeleteResourcePolicy` Necesario para eliminar una política de recursos de Registros de CloudWatch.
DeleteRetentionPolicy	`logs:DeleteRetentionPolicy` Necesario para eliminar la política de retención de un grupo de registros.
DeleteSubscriptionFilter	`logs:DeleteSubscriptionFilter` Necesario para eliminar el filtro de suscripción asociado a un grupo de registros.
DescribeDestinations	`logs:DescribeDestinations` Necesario para ver todos los destinos asociados a la cuenta.

Operaciones de la API de Registros de CloudWatch	Permisos necesarios (acciones de API)
DescribeExportTasks	`logs:DescribeExportTasks` Necesario para ver todas las tareas de exportación asociadas a la cuenta.
DescribeLogGroups	`logs:DescribeLogGroups` Necesario para ver todos los grupos de registro asociados a la cuenta.
DescribeLogStreams	`logs:DescribeLogStreams` Necesario para ver todos los flujos de registro asociados a un grupo de registros.
DescribeMetricFilters	`logs:DescribeMetricFilters` Necesario para ver todas las métricas asociadas a un grupo de registros.
DescribeQueryDefinitions	`logs:DescribeQueryDefinitions` Necesario para ver la lista de definiciones de consulta guardadas en Información de registros de CloudWatch.
DescribeQueries	`logs:DescribeQueries` Necesario para ver la lista de consultas de Información de registros de CloudWatch programadas, en proceso de ejecución o ejecutadas recientemente.

Operaciones de la API de Registros de CloudWatch	Permisos necesarios (acciones de API)
DescribeResourcePolicies	`logs:DescribeResourcePolicies` Necesario para ver una lista de políticas de recursos de Registros de CloudWatch.
DescribeSubscriptionFilters	`logs:DescribeSubscriptionFilters` Necesario para ver todos los filtros de suscripción asociados con un grupo de registros.
FilterLogEvents	`logs:FilterLogEvents` Necesario para ordenar los eventos de registros por patrón de filtro de grupo de registros.
GetLogEvents	`logs:GetLogEvents` Necesario para recuperar eventos de registro de un flujo de registros.
GetLogGroupFields	`logs:GetLogGroupFields` Necesario para recuperar la lista de campos que se incluyen en los eventos de registro de un grupo de registros.
GetLogRecord	`logs:GetLogRecord` Necesario para recuperar los detalles de un único evento de registro.

Operaciones de la API de Registros de CloudWatch	Permisos necesarios (acciones de API)
GetQueryResults	`logs:GetQueryResults` Necesario para recuperar los resultados de las consultas de Información de registros de CloudWatch.
ListTagsLogGroup	`logs:ListTagsLogGroup` Necesario para ver las etiquetas asociadas a un grupo de registros.
PutDestination	`logs:PutDestination` Necesario para crear o actualizar un flujo de registros de destino (como, por ejemplo, un flujo de Kinesis).
PutDestinationPolicy	`logs:PutDestinationPolicy` Necesario para crear o actualizar una política de acceso asociada a un destino de registro existente.
PutLogEvents	`logs:PutLogEvents` Necesario para cargar un lote de eventos de registro en un flujo de registros.
PutMetricFilter	`logs:PutMetricFilter` Necesario para crear o actualizar un filtro de métricas y asociarlo a un grupo de registros.

Operaciones de la API de Registros de CloudWatch	Permisos necesarios (acciones de API)
PutQueryDefinition	`logs:PutQueryDefinition` Necesario para guardar una consulta en Información de registros de CloudWatch.
PutResourcePolicy	`logs:PutResourcePolicy` Necesario para crear una política de recursos de Registros de CloudWatch.
PutRetentionPolicy	`logs:PutRetentionPolicy` Necesario para establecer el número de días que conservar los eventos de registro (retenció n) en un grupo de registros.
PutSubscriptionFilter	`logs:PutSubscriptionFilter` Necesario para crear o actualizar un filtro de suscripción y asociarlo a un grupo de registros.
StartQuery	`logs:StartQuery` Necesario para comenzar consultas de Información de registros de CloudWatch.
StopQuery	`logs:StopQuery` Necesario para detener una consulta en curso de Información de registros de CloudWatch.

Operaciones de la API de Registros de CloudWatch	Permisos necesarios (acciones de API)
TagLogGroup	`logs:TagLogGroup` Necesario para añadir o actualizar etiquetas de grupo de registro.
TestMetricFilter	`logs:TestMetricFilter` Necesario para probar un patrón de filtro con respecto a una muestra de mensajes de evento de registro.

Operaciones de la API de Amazon EC2 y permisos necesarios para realizar acciones

Operaciones de la API de Amazon EC2	Permisos necesarios (acciones de API)
DescribeInstanceStatus	`ec2:DescribeInstanceStatus` Necesario para ver los detalles de estado de instancia EC2.
DescribeInstances	`ec2:DescribeInstances` Necesario para ver los detalles de instancia EC2.
RebootInstances	`ec2:RebootInstances` Necesario para reiniciar una instancia EC2.
StopInstances	`ec2:StopInstances` Necesario para parar una instancia EC2.

Operaciones de la API de Amazon EC2	Permisos necesarios (acciones de API)
TerminateInstances	`ec2:TerminateInstances` Necesario para terminar una instancia EC2.

Operaciones de la API de Amazon EC2 Auto Scaling y permisos necesarios para realizar acciones

Operaciones de la API de Amazon EC2 Auto Scaling	Permisos necesarios (acciones de API)
Escalado	`autoscaling:Scaling` Necesario para escalar un grupo de Auto Scaling.
Desencadenador	`autoscaling:Trigger` Necesario para activar una acción de Auto Scaling.

Validación de conformidad de Amazon CloudWatch

Auditores de terceros evalúan la seguridad y la conformidad de Amazon CloudWatch como parte de distintos programas de conformidad de AWS. Estos incluyen SOC, PCI, FedRAMP, HIPAA y otros.

Para obtener una lista de los servicios de AWS en el ámbito de programas de conformidad específicos, consulte AWS Services in Scope by Compliance Program (Servicios en el ámbito de programas de conformidad). Para obtener información general, consulte Programas de conformidad de AWS.

Puede descargar los informes de auditoría de terceros utilizando AWS Artifact. Para obtener más información, consulte Descarga de informes en AWS Artifact.

La responsabilidad de conformidad al utilizar Amazon CloudWatch se determina en función de la confidencialidad de los datos, los objetivos de cumplimiento de la empresa y las leyes y normativas aplicables. AWS proporciona los siguientes recursos para ayudar con la conformidad:

- Security and Compliance Quick Start Guides (Guías de inicio rápido de seguridad y conformidad) (Guías de inicio rápido de seguridad y conformidad): Estas guías de implementación analizan consideraciones sobre arquitectura y proporcionan los pasos para implementar los entornos de referencia centrados en la seguridad y la conformidad en AWS.
- Documento técnico sobre arquitectura para seguridad y conformidad de HIPAA : en este documento técnico, se describe cómo las empresas pueden utilizar AWS para crear aplicaciones conformes con HIPAA.
- AWSRecursos de conformidad de: este conjunto de manuales y guías podría aplicarse a su sector y ubicación.
- Evaluación de recursos con reglas en la Guía para desarrolladores de AWS Config: AWS Config evalúa en qué medida las configuraciones de sus recursos cumplen las prácticas internas, las directrices del sector y las normativas.
- AWS Security Hub: este servicio de AWS proporciona una vista integral de su estado de seguridad en AWS que lo ayuda a verificar la conformidad con los estándares y las prácticas recomendadas del sector de seguridad.

Resiliencia de Amazon CloudWatch

La infraestructura global de AWS está conformada por regiones y zonas de disponibilidad de AWS. Las regiones proporcionan varias zonas de disponibilidad físicamente independientes y aisladas que se encuentran conectadas mediante redes con un alto nivel de rendimiento y redundancia, además de baja latencia. Con las zonas de disponibilidad, puede diseñar y utilizar aplicaciones y bases de datos que realizan una conmutación por error automática entre las zonas sin interrupciones. Las zonas de disponibilidad tienen una mayor disponibilidad, tolerancia a errores y escalabilidad que las infraestructuras tradicionales de centros de datos únicos o múltiples.

Para obtener más información sobre las regiones y zonas de disponibilidad de AWS, consulte Infraestructura global de AWS.

Seguridad de la infraestructura en Amazon CloudWatch

Como se trata de un servicio administrado, Amazon CloudWatch está protegido por la seguridad de red global de AWS. Para obtener información sobre los servicios de seguridad de AWS y cómo AWS protege la infraestructura, consulte Seguridad en la nube de AWS. Para diseñar su entorno de AWS con las prácticas recomendadas de seguridad de infraestructura, consulte Protección de la infraestructura en Portal de seguridad de AWS Well-Architected Framework.

Puede utilizar llamadas a la API publicadas de AWS para obtener acceso a CloudWatch a través de la red. Los clientes deben admitir lo siguiente:

- Seguridad de la capa de transporte (TLS). Nosotros exigimos TLS 1.2 y recomendamos TLS 1.3.
- Conjuntos de cifrado con confidencialidad directa total (PFS) tales como DHE (Ephemeral Diffie-Hellman) o ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). La mayoría de los sistemas modernos como Java 7 y posteriores son compatibles con estos modos.

Además, las solicitudes deben estar firmadas mediante un ID de clave de acceso y una clave de acceso secreta que esté asociada a una entidad de seguridad de IAM. También puede utilizar AWS Security Token Service (AWS STS) para generar credenciales de seguridad temporales para firmar solicitudes.

Aislamiento de red

Una Virtual Private Cloud (VPC) es una red virtual en su propia área, aislada lógicamente en la nube de Amazon Web Services. Una subred es un rango de direcciones IP de una VPC. Puede implementar diversos recursos de AWS en las subredes de las VPC. Por ejemplo, puede implementar instancias de Amazon EC2, clústeres de EMR y tablas de DynamoDB en las subredes. Para obtener más información, consulte la Guía del usuario de Amazon VPC.

Para permitir que CloudWatch se comunique con los recursos en una VPC sin pasar por la Internet pública, utilice AWS PrivateLink. Para obtener más información, consulte Uso de CloudWatch y CloudWatch Synthetics con los puntos de enlace de la VPC de tipo interfaz.

Una subred privada es una subred que no tiene una ruta predeterminada a la Internet pública. La implementación de un recurso de AWS en una subred privada no impide que Amazon CloudWatch recopile las métricas integradas del recurso.

Si necesita publicar métricas personalizadas desde un recurso de AWS en una subred privada, puede hacerlo con un servidor proxy. El servidor proxy reenvía esas solicitudes HTTPS a los puntos de enlace de la API pública para CloudWatch.

AWS Security Hub

Puede supervisar el uso de RDS en relación con las prácticas recomendadas de seguridad con AWS Security Hub. Security Hub utiliza controles de seguridad para evaluar las configuraciones de los recursos y los estándares de seguridad para ayudarlo a cumplir con varios marcos de conformidad. Para obtener más información sobre el uso de Security Hub para evaluar los recursos de CloudWatch, consulte Controles de Amazon CloudWatch en la Guía del usuario de AWS Security Hub.

Uso de CloudWatch y CloudWatch Synthetics con los puntos de enlace de la VPC de tipo interfaz

Si utiliza Amazon Virtual Private Cloud (Amazon VPC) para alojar los recursos de AWS, puede establecer una conexión privada entre la VPC, CloudWatch y CloudWatch Synthetics. Puede utilizar estas conexiones para habilitar CloudWatch y CloudWatch Synthetics para comunicarse con los recursos en la VPC sin pasar por la internet pública.

Amazon VPC es un servicio de AWS que puede utilizar para lanzar recursos de AWS en una red virtual que usted defina. Con una VPC, puede controlar la configuración de la red, como el rango de direcciones IP, las subredes, las tablas de ruteo y las gateways de red. Para conectar la VPC a CloudWatch o a CloudWatch Synthetics, defina un punto de enlace de la VPC de tipo interfaz para conectar la VPC a los servicios de AWS. El punto de enlace ofrece conectividad escalable de confianza con CloudWatch o CloudWatch Synthetics sin necesidad de utilizar una gateway de Internet, una instancia (NAT) de traducción de dirección de red o una conexión de VPN. Para obtener más información, consulte What Is Amazon VPC (¿Qué es Amazon VPC?) en la Guía del usuario de Amazon VPC.

Los puntos de enlace de la VPC de tipo interfaz utilizan la tecnología de AWS PrivateLink, una tecnología de AWS que permite la comunicación privada entre los servicios de AWS mediante una interfaz de red elástica con direcciones IP privadas. Para obtener más información, consulte la publicación de blog New – AWS PrivateLink for AWS Services.

Los siguientes pasos son para usuarios de Amazon VPC. Para obtener más información, consulte Introducción en la Guía del usuario de Amazon VPC.

Punto de enlace de la VPC de CloudWatch

CloudWatch actualmente admite puntos de enlace de la VPC en las siguientes Regiones de AWS:

- Este de EE. UU. (Ohio)
- EE.UU. Este (Norte de Virginia)
- EE.UU. Oeste (Norte de California)
- Oeste de EE. UU. (Oregón)
- Asia-Pacífico (Hong Kong)
- Asia-Pacífico (Mumbai)
- Asia Pacific (Seoul)
- Asia Pacífico (Singapur)
- Asia Pacífico (Sídney)
- Asia Pacífico (Tokio)
- Canada (Central)
- Europe (Frankfurt)
- Europa (Irlanda)
- Europe (London)
- Europa (París)
- Medio Oriente (EAU)
- América del Sur (São Paulo)
- AWS GovCloud (Este de EE. UU.)
- AWS GovCloud (Oeste de EE. UU.)

Creación de un punto de enlace de la VPC para CloudWatch

Para comenzar a utilizar CloudWatch con la VPC, cree un punto de enlace de la VPC de tipo interfaz para CloudWatch. El nombre del servicio que se va a elegir es `com.amazonaws.`*`region`*`.monitoring`. Para obtener más información, consulte Creación de un punto de enlace de interfaz en la Guía del usuario de Amazon VPC.

No necesita cambiar la configuración de CloudWatch. CloudWatch llama a otros servicios de AWS con puntos de enlace públicos o puntos de enlace de la VPC privados de tipo interfaz, lo que esté

en uso. Por ejemplo, si crea punto de enlace de la VPC de tipo interfaz para CloudWatch y ya tiene métricas que circulan por CloudWatch desde los recursos que se encuentran en la VPC, estas métricas comienzan a circular por el punto de enlace de la VPC de tipo interfaz de forma predeterminada.

Control del acceso al punto de enlace de la VPC de CloudWatch

Una política de punto de conexión de VPC es una política de recursos de IAM que puede asociar a un punto de conexión cuando crea o modifica el punto de conexión. Si no adjunta una política al crear un punto de enlace, Amazon VPC adjunta una política predeterminada que le conceda acceso completo al servicio. Una política de punto de conexión no anula ni sustituye a las políticas de usuario de ni las políticas específicas de los servicios. Se trata de una política independiente para controlar el acceso desde el punto de conexión al servicio especificado.

Las políticas de punto de conexión deben escribirse en formato JSON.

Para obtener más información, consulte Controlar el acceso a servicios con puntos de conexión de VPC en la Guía del usuario de Amazon VPC.

A continuación, se muestra un ejemplo de una política de puntos de enlace de CloudWatch. Esta política permite a los usuarios que se conectan a CloudWatch a través de la VPC que envíen datos de métricas a CloudWatch, y les impide que realicen otras acciones de CloudWatch.

```
{
  "Statement": [
    {
      "Sid": "PutOnly",
      "Principal": "*",
      "Action": [
        "cloudwatch:PutMetricData"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

Para editar la política de punto de enlace de la VPC para CloudWatch

1. Abra la consola de Amazon VPC en https://console.aws.amazon.com/vpc/.
2. En el panel de navegación, elija Puntos de conexión.

3. Si todavía no ha creado el punto de enlace para CloudWatch, elija Create Endpoint (Crear punto de enlace). Seleccione com.amazonaws.***region***.monitoring y, a continuación, elija Create endpoint (Crear punto de enlace).
4. Seleccione el punto de enlace com.amazonaws.***region***.monitoring y, a continuación, elija la pestaña Policy (Política).
5. Elija Edit Policy (Editar política) y, a continuación, realice los cambios.

Puntos de enlace de la VPC de CloudWatch Synthetics

Actualmente, CloudWatch Synthetics admite puntos de enlace de la VPC en las siguientes Regiones de AWS:

- Este de EE. UU. (Ohio)
- EE.UU. Este (Norte de Virginia)
- EE.UU. Oeste (Norte de California)
- Oeste de EE. UU. (Oregón)
- Asia-Pacífico (Hong Kong)
- Asia-Pacífico (Mumbai)
- Asia Pacific (Seoul)
- Asia Pacífico (Singapur)
- Asia Pacífico (Sídney)
- Asia Pacífico (Tokio)
- Canada (Central)
- Europe (Frankfurt)
- Europa (Irlanda)
- Europe (London)
- Europa (París)
- América del Sur (São Paulo)

Creación de un punto de enlace de la VPC para CloudWatch Synthetics

Para empezar a utilizar CloudWatch Synthetics con la VPC, cree un punto de enlace de la VPC de tipo interfaz para CloudWatch Synthetics. El nombre del servicio que se va a elegir es

`com.amazonaws.region.synthetics`. Para obtener más información, consulte Creación de un punto de enlace de interfaz en la Guía del usuario de Amazon VPC.

No necesita cambiar la configuración de CloudWatch Synthetics. CloudWatch Synthetics se comunica con otros servicios de AWS con los puntos de enlace públicos o los puntos de enlace de la VPC de tipo interfaz privados, lo que esté en uso. Por ejemplo, si crea un punto de enlace de la VPC de tipo interfaz para CloudWatch Synthetics y ya dispone de un punto de enlace tipo interfaz para Amazon S3, CloudWatch Synthetics inicia la comunicación con Amazon S3 a través del punto de enlace de la VPC de tipo interfaz de forma predeterminada.

Control del acceso a su punto de enlace de la VPC de CloudWatch Synthetics

Una política de punto de conexión de VPC es una política de recursos de IAM que puede asociar a un punto de conexión cuando crea o modifica el punto de conexión. Si no asocia una política al crear un punto de enlace, se asociará automáticamente una política predeterminada que conceda acceso completo al servicio. Una política de punto de conexión no anula ni sustituye a las políticas de usuario de ni las políticas específicas de los servicios. Se trata de una política independiente para controlar el acceso desde el punto de conexión al servicio especificado.

Las políticas de punto de enlace afectan a los canaries que la VPC administra de forma privada. No son necesarios para canaries que se ejecutan en subredes privadas.

Las políticas de punto de conexión deben escribirse en formato JSON.

Para obtener más información, consulte Controlar el acceso a servicios con puntos de conexión de VPC en la Guía del usuario de Amazon VPC.

A continuación, se muestra un ejemplo de una política de punto de enlace de CloudWatch Synthetics. Esta política permite que los usuarios que se conectan a CloudWatch Synthetics a través de la VPC puedan ver información sobre los canaries y las ejecuciones, pero no crear, modificar ni eliminar los canaries.

```
{
    "Statement": [
        {
            "Action": [
                "synthetics:DescribeCanaries",
                "synthetics:GetCanaryRuns"
            ],
            "Effect": "Allow",
            "Resource": "*",
```

```
            "Principal": "*"
        }
    ]
}
```

Para editar la política de punto de enlace de la VPC para CloudWatch Synthetics

1. Abra la consola de Amazon VPC en https://console.aws.amazon.com/vpc/.
2. En el panel de navegación, elija Puntos de conexión.
3. Si todavía no ha creado el punto de enlace para CloudWatch Synthetics, elija Create Endpoint (Crear punto de enlace). Seleccione com.amazonaws.***region***.synthetics y, a continuación, elija Create endpoint (Crear punto de enlace).
4. Seleccione el punto de enlace com.amazonaws.***region***.synthetics y, a continuación, elija la pestaña Policy (Política).
5. Elija Edit Policy (Editar política) y, a continuación, realice los cambios.

Consideraciones de seguridad para los canaries de Synthetics

En las siguientes secciones se explican los problemas de seguridad que debe tener en cuenta al crear y ejecutar canaries en Synthetics.

Use conexiones seguras

Dado que el código de valor controlado y los resultados de las ejecuciones de prueba de valor controlado pueden contener información confidencial, no haga que su valor controlado se conecte a puntos de conexión a través de conexiones no cifradas. Utilice siempre conexiones cifradas, como aquellas que empiezan por `https://`.

Consideraciones de nomenclatura para valores controlados

El nombre de recurso de Amazon (ARN) de un valor controlado se incluye en la cabecera del usuario-agente como parte de las llamadas salientes que se realicen desde el navegador Chromium impulsado por Puppeteer que se incluye como parte de la biblioteca contenedora de CloudWatch Synthetics. Ayuda a identificar el tráfico de valores controlados de CloudWatch Synthetics y a relacionarlo de nuevo con los valores controlados que realizan las llamadas.

El ARN de valor controlado incluye el nombre del valor controlado. Elija nombres de valor controlado que no revelen información privada.

Además, asegúrese de apuntar sus canaries solo a aquellos sitios web y puntos de enlace que controle.

Secretos e información confidencial en código de valor controlado

Si pasa el código de valor controlado directamente al valor controlado usando un archivo zip, el contenido del script se puede ver en los registros de AWS CloudTrail.

Si tiene información confidencial o secretos (como claves de acceso o credenciales de base de datos) en un script de valor controlado, le recomendamos encarecidamente que almacene el script como un objeto con control de versiones en Amazon S3 y pase la ubicación de Amazon S3 al valor controlado, en lugar de pasar el código de valor controlado mediante un archivo zip.

Si utiliza un archivo zip para pasar el script de valor controlado, le recomendamos encarecidamente que no incluya secretos ni información confidencial en el código fuente del valor controlado. Para obtener más información acerca de cómo utilizar AWS Secrets Manager para ayudar a mantener seguros sus secretos, consulte ¿Qué es AWS Secrets Manager?.

Consideraciones de permisos

Se recomienda que restrinja el acceso a los recursos que CloudWatch Synthetics ha creado o utilizado. Utilice permisos estrictos en los buckets de Amazon S3 donde los canaries almacenan los resultados de las ejecuciones de prueba y otros artefactos, como registros y capturas de pantalla.

Del mismo modo, mantenga permisos estrictos en las ubicaciones donde se almacena el código fuente del valor controlado, de modo que ningún usuario elimine accidentalmente o de forma maliciosa las capas o funciones de Lambda que se han utilizado para el valor controlado.

Para ayudar a garantizar la ejecución del código de valor controlado deseado, puede utilizar el control de versiones de objetos en el bucket de Amazon S3 donde se almacena el código del valor controlado. A continuación, cuando especifique este código para ejecutarlo como valor controlado, puede incluir el objeto `versionId` como parte de la ruta, como en los ejemplos siguientes.

```
https://bucket.s3.amazonaws.com/path/object.zip?versionId=version-id
https://s3.amazonaws.com/bucket/path/object.zip?versionId=version-id
https://bucket.s3-region.amazonaws.com/path/object.zip?versionId=version-id
```

Seguimientos de pilas y mensajes de excepción

De forma predeterminada, los valores controlados de CloudWatch Synthetics capturan cualquier excepción que el script del valor controlado produzca, independientemente de si el script es personalizado o proviene de un esquema. CloudWatch Synthetics registra tanto el mensaje de excepción como el seguimiento de la pila en tres ubicaciones:

- Vuelva al servicio de CloudWatch Synthetics para acelerar la depuración al describir las ejecuciones de prueba
- En CloudWatch Logs en función de la configuración con la que se crean las funciones de Lambda
- En el archivo de registro de Synthetics, que es un archivo de texto llano que se carga en la ubicación de Amazon S3 especificada por el valor que se establece para la ubicación de resultados del valor controlado `resultsLocation`

Si desea enviar y almacenar menos información, puede capturar excepciones antes de que regresen a la biblioteca contenedora de CloudWatch Synthetics.

También puede haber URL de solicitud en los errores. CloudWatch Synthetics busca cualquier URL en el error que el script genera y edita los parámetros de URL restringidos de ellas en función de la configuración restrictedUrlParameters. Si está registrando mensajes de error en su script, puede usar getSanitizedErrorMessage para editar direcciones URL antes de registrar.

Limite el alcance de los roles de IAM

Le recomendamos que no configure el valor controlado para visitar URL o puntos de conexión potencialmente malintencionados. Apuntar el valor controlado a sitios web o puntos de conexión desconocidos o que no son de confianza podría exponer el código de la función de Lambda a scripts de un usuario malicioso. Si un sitio web malicioso puede salir de Chromium, podría tener acceso a su código Lambda de una manera similar a si se conectó a él mediante un navegador de Internet.

Ejecute la función de Lambda con un rol de ejecución de IAM que tenga permisos más focalizados. De esta forma, si la función de Lambda se ve comprometida por un script malicioso, tendrá acciones limitadas que puede realizar cuando se ejecute como la cuenta de AWS del valor controlado.

Cuando se utiliza la consola de CloudWatch para crear un valor controlado, se crea con un rol de ejecución de IAM reducido.

Redacción de datos confidenciales

CloudWatch Synthetics captura las direcciones URL, el código de estado, los motivos del error (si existe) y cabeceras y cuerpos de solicitudes y respuestas. Esto permite al usuario del valor controlado a comprender, monitorear y depurar valores controlados.

Las configuraciones descritas en las siguientes secciones se pueden establecer en cualquier punto de la ejecución del valor controlado. También puede optar por aplicar diferentes configuraciones a diferentes pasos de Synthetics.

Direcciones URL de solicitud

De forma predeterminada, CloudWatch Synthetics registra las direcciones URL de solitud, los códigos de estado y el motivo de estado de cada URL en los registros de los valores controlados. Las URL de solicitud también pueden aparecer en informes de ejecución de valores controlados, archivos HAR, etc. La URL de solicitud puede contener parámetros de consulta confidenciales, como tokens de acceso o contraseñas. Puede editar la información confidencial para que CloudWatch Synthetics no la registre.

Para editar la información confidencial, establezca la propiedad de configuración restrictedUrlParameters. Para obtener más información, consulte Clase de SyntheticsConfiguration. Esto hace que CloudWatch Synthetics edite los parámetros de URL, incluidos los valores de los parámetros de ruta y de consulta, basados en restrictedUrlParameters antes de que se realice el registro. Si está registrando direcciones URL en el script, puede usar getSanitizedUrl(url, stepConfig = null) para editar las direcciones URL antes de que se realice el registro. Para obtener más información, consulte Clase de SyntheticSloghelper.

Encabezados

De forma predeterminada, CloudWatch Synthetics no registra las cabeceras de solicitud o respuesta. Para canarios de la UI, este es el comportamiento predeterminado para canaries que usan la versión de tiempo de ejecución `syn-nodejs-puppeteer-3.2` y posteriores.

Si las cabeceras no contienen información confidencial, puede habilitar las cabeceras en los informes de archivos HAR y HTTP al establecer la propiedades includeRequestHeaders y includeResponseHeaders a `true`. Puede habilitar todos las cabeceras pero optar por restringir los valores de las claves de cabecera confidenciales. Por ejemplo, puede elegir solo editar las cabeceras `Authorization` de los artefactos que los canaries producen.

Cuerpo de solicitud y de respuesta

De forma predeterminada, CloudWatch Synthetics no registra el cuerpo de solicitud o de respuesta en registros o informes de valores controlados. Esta información resulta especialmente útil para los canarios de la API. Synthetics captura todas las solicitudes HTTP y puede mostrar cabeceras y cuerpos de solicitud y de respuesta. Para obtener más información, consulte executeHttpStep(stepName, requestOptions, [callback], [stepConfig]). Puede elegir habilitar el cuerpo de solicitud o de respuesta al establecer las propiedades includeRequestBody y includeResponseBody en `true`.

Registro de llamadas a la API de Amazon CloudWatch con AWS CloudTrail

Amazon CloudWatch y CloudWatch Synthetics están integrados con AWS CloudTrail, un servicio que proporciona un registro de las acciones de los usuarios, los roles o un servicio de AWS. CloudTrail captura las llamadas a la API que se realizaron desde su cuenta AWS o en su nombre. Las llamadas capturadas incluyen las llamadas realizadas desde la consola y llamadas de código a las operaciones de la API.

Si crea un seguimiento, puede habilitar la entrega continua de eventos de CloudTrail a un bucket de S3, incluidos los eventos de CloudWatch. Si no configura un registro de seguimiento, puede ver los eventos más recientes en la consola de CloudTrail en el Historial de eventos. Mediante la información que CloudTrail recopila, se puede determinar la petición que se envió a CloudWatch, la dirección IP desde la que se realizó la petición, quién la realizó, cuándo se realizó y otros detalles adicionales.

Para obtener más información acerca de CloudTrail, incluso cómo configurarlo y habilitarlo, consulte la Guía del usuario de AWS CloudTrail.

Cada entrada de registro o evento contiene información sobre quién generó la solicitud. La información de identidad del usuario lo ayuda a determinar lo siguiente:

- Si la solicitud se realizó con credenciales de usuario de AWS Identity and Access Management (IAM) o credenciales de usuario raíz.
- Si la solicitud se realizó con credenciales de seguridad temporales de un rol o fue un usuario federado.
- Si la solicitud la realizó otro servicio de AWS.

Para obtener más información, consulte el Elemento userIdentity de CloudTrail.

Para mantener un registro continuo de los eventos de la cuenta de AWS, incluidos los eventos de CloudWatch y CloudWatch Synthetics, cree un seguimiento. Un registro de seguimiento permite a CloudTrail que pueda enviar archivos de registro a un bucket de S3. De manera predeterminada, cuando se crea un registro de seguimiento en la consola, el registro de seguimiento se aplica a todas las regiones de AWS. El registro de seguimiento registra los eventos de todas las regiones de la partición de AWS y envía los archivos de registro al bucket de S3 especificado. También puede

configurar otros servicios de AWS para analizar y actuar según los datos de eventos recopilados en los registros de CloudTrail. Para más información, consulte los siguientes temas:

- Introducción a la creación de registros de seguimiento
- Servicios e integraciones compatibles con CloudTrail
- Configurar notificaciones de Amazon SNS para CloudTrail
- Recepción de archivos de registro de CloudTrail de varias regiones y Recepción de archivos de registro de CloudTrail de varias cuentas

> **Note**
>
> Para obtener información sobre las llamadas a la API de Registros de CloudWatch que se registran en CloudTrail, consulte la información sobre Registros de CloudWatch en CloudTrail.

Temas

- Información de CloudWatch en CloudTrail
- CloudWatch Internet Monitor en CloudTrail
- Información de CloudWatch Synthetics en CloudTrail

Información de CloudWatch en CloudTrail

CloudWatch admite el registro de las siguientes acciones como eventos en archivos de registros de CloudTrail:

- DeleteAlarms
- DeleteAnomalyDetector
- DeleteDashboards
- DescribeAlarmHistory
- DescribeAlarms
- DescribeAlarmsForMetric
- DescribeAnomalyDetectors
- DisableAlarmActions

- EnableAlarmActions
- GetDashboard
- ListDashboards
- PutAnomalyDetector
- PutDashboard
- PutMetricAlarm
- SetAlarmState

Ejemplo: Entradas de archivos de registros de CloudWatch

En el siguiente ejemplo, se muestra una entrada de registro de CloudTrail que ilustra la acción `PutMetricAlarm`.


```
{
    "Records": [{
        "eventVersion": "1.01",
        "userIdentity": {
            "type": "Root",
            "principalId": "EX_PRINCIPAL_ID",
            "arn": "arn:aws:iam::123456789012:root",
            "accountId": "123456789012",
            "accessKeyId": "EXAMPLE_KEY_ID"
        },
        "eventTime": "2014-03-23T21:50:34Z",
        "eventSource": "monitoring.amazonaws.com",
        "eventName": "PutMetricAlarm",
        "awsRegion": "us-east-1",
        "sourceIPAddress": "127.0.0.1",
        "userAgent": "aws-sdk-ruby2/2.0.0.rc4 ruby/1.9.3 x86_64-linux Seahorse/0.1.0",
        "requestParameters": {
            "threshold": 50.0,
            "period": 60,
            "metricName": "CloudTrail Test",
            "evaluationPeriods": 3,
            "comparisonOperator": "GreaterThanThreshold",
            "namespace": "AWS/CloudWatch",
            "alarmName": "CloudTrail Test Alarm",
            "statistic": "Sum"
        },
        "responseElements": null,
```

```
        "requestID": "29184022-b2d5-11e3-a63d-9b463e6d0ff0",
        "eventID": "b096d5b7-dcf2-4399-998b-5a53eca76a27"
    },
    ..additional entries
  ]
  }
```

La siguiente entrada de archivo de registro muestra que un usuario ha llamado a la acción `PutRule` de CloudWatch Events.

```
{
         "eventVersion":"1.03",
         "userIdentity":{
            "type":"Root",
            "principalId":"123456789012",
            "arn":"arn:aws:iam::123456789012:root",
            "accountId":"123456789012",
            "accessKeyId":"AKIAIOSFODNN7EXAMPLE",
            "sessionContext":{
               "attributes":{
                  "mfaAuthenticated":"false",
                  "creationDate":"2015-11-17T23:56:15Z"
               }
            }
         },
         "eventTime":"2015-11-18T00:11:28Z",
         "eventSource":"events.amazonaws.com",
         "eventName":"PutRule",
         "awsRegion":"us-east-1",
         "sourceIPAddress":"AWS Internal",
         "userAgent":"AWS CloudWatch Console",
         "requestParameters":{
            "description":"",
            "name":"cttest2",
            "state":"ENABLED",
            "eventPattern":"{\"source\":[\"aws.ec2\"],\"detail-type\":[\"EC2 Instance
 State-change Notification\"]}",
            "scheduleExpression":""
         },
         "responseElements":{
            "ruleArn":"arn:aws:events:us-east-1:123456789012:rule/cttest2"
         },
         "requestID":"e9caf887-8d88-11e5-a331-3332aa445952",
```

```
         "eventID":"49d14f36-6450-44a5-a501-b0fdcdfaeb98",
         "eventType":"AwsApiCall",
         "apiVersion":"2015-10-07",
         "recipientAccountId":"123456789012"
}
```

La siguiente entrada del archivo de registro muestra que un usuario ha llamado a la acción `CreateExportTask` de CloudWatch Logs.

```
{
        "eventVersion": "1.03",
        "userIdentity": {
            "type": "IAMUser",
            "principalId": "EX_PRINCIPAL_ID",
            "arn": "arn:aws:iam::123456789012:user/someuser",
            "accountId": "123456789012",
            "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
            "userName": "someuser"
        },
        "eventTime": "2016-02-08T06:35:14Z",
        "eventSource": "logs.amazonaws.com",
        "eventName": "CreateExportTask",
        "awsRegion": "us-east-1",
        "sourceIPAddress": "127.0.0.1",
        "userAgent": "aws-sdk-ruby2/2.0.0.rc4 ruby/1.9.3 x86_64-linux Seahorse/0.1.0",
        "requestParameters": {
            "destination": "yourdestination",
            "logGroupName": "yourloggroup",
            "to": 123456789012,
            "from": 0,
            "taskName": "yourtask"
        },
        "responseElements": {
            "taskId": "15e5e534-9548-44ab-a221-64d9d2b27b9b"
        },
        "requestID": "1cd74c1c-ce2e-12e6-99a9-8dbb26bd06c9",
        "eventID": "fd072859-bd7c-4865-9e76-8e364e89307c",
        "eventType": "AwsApiCall",
        "apiVersion": "20140328",
        "recipientAccountId": "123456789012"
}
```

CloudWatch Internet Monitor en CloudTrail

CloudWatch Internet Monitor admite el registro de las siguientes acciones como eventos en archivos de registro de CloudTrail.

- CreateMonitor
- DeleteMonitor
- GetHealthEvent
- GetMonitor
- GetQueryResults
- GetQueryStatus
- ListHealthEvents
- ListMonitors
- ListTagsForResource
- StartQuery
- StopQuery
- UpdateMonitor

Ejemplo: entradas de archivos de registro de CloudWatch Internet Monitor

En el siguiente ejemplo, se muestra una entrada de registro de CloudTrail Internet Monitor que ilustra la acción `ListMonitors`.

```
{
        "eventVersion": "1.08",
        "userIdentity": {
            "type": "AssumedRole",
            "principalId": "EX_PRINCIPAL_ID",
            "arn": "arn:aws:iam::000000000000:assumed-role/role_name",
            "accountId":"123456789012",
            "accessKeyId":"AKIAIOSFODNN7EXAMPLE",
            "sessionContext": {
                "sessionIssuer": {
                "type": "Role",
                "principalId": "EX_PRINCIPAL_ID",
                "arn": "arn:aws:iam::000000000000:role/Admin",
                "accountId":"123456789012",
```

```
                "userName": "SAMPLE_NAME"
                },
                "webIdFederationData": {},
                "attributes": {
                    "creationDate": "2022-10-11T17:25:41Z",
                    "mfaAuthenticated": "false"
                }
            }
        },
        "eventTime": "2022-10-11T17:30:18Z",
        "eventSource": "internetmonitor.amazonaws.com",
        "eventName": "ListMonitors",
        "awsRegion": "us-east-2",
        "sourceIPAddress": "192.0.2.0",
        "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7)",
        "requestParameters": null,
        "responseElements": null,
        "requestID": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "eventID": "a1b2c3d4-5678-90ab-cdef-EXAMPLEbbbbb",
        "readOnly": true,
        "eventType": "AwsApiCall",
        "managementEvent": true,
        "recipientAccountId": "111122223333",
        "eventCategory": "Management"
    }
```

En el siguiente ejemplo, se muestra una entrada de registro de CloudTrail Internet Monitor que ilustra la acción `CreateMonitor`.

```
{
        "eventVersion": "1.08",
        "userIdentity": {
            "type": "AssumedRole",
            "principalId": "EX_PRINCIPAL_ID",
            "arn": "arn:aws:iam::000000000000:assumed-role/role_name",
            "accountId":"123456789012",
            "accessKeyId":"AKIAIOSFODNN7EXAMPLE",
            "sessionContext": {
                "sessionIssuer": {
                "type": "Role",
                "principalId": "EX_PRINCIPAL_ID",
                "arn": "arn:aws:iam::000000000000:role/Admin",
                "accountId":"123456789012",
```

```
                "userName": "SAMPLE_NAME"
                },
                "webIdFederationData": {},
                "attributes": {
                    "creationDate": "2022-10-11T17:25:41Z",
                    "mfaAuthenticated": "false"
                }
            }
        },
        "eventTime": "2022-10-11T17:30:08Z",
        "eventSource": "internetmonitor.amazonaws.com",
        "eventName": "CreateMonitor",
        "awsRegion": "us-east-2",
        "sourceIPAddress": "192.0.2.0",
        "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7)",
        "requestParameters": {
            "MonitorName": "TestMonitor",
            "Resources": ["arn:aws:ec2:us-east-2:444455556666:vpc/vpc-febc0b95"],
            "ClientToken": "a1b2c3d4-5678-90ab-cdef-EXAMPLE33333"
        },
        "responseElements": {
            "Arn": "arn:aws:internetmonitor:us-east-2:444455556666:monitor/ct-
onboarding-test",
            "Status": "PENDING"
        },
        "requestID": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "eventID": "a1b2c3d4-5678-90ab-cdef-EXAMPLEbbbbb",
        "readOnly": false,
        "eventType": "AwsApiCall",
        "managementEvent": true,
        "recipientAccountId": "111122223333",
        "eventCategory": "Management"
    }
```


Información de CloudWatch Synthetics en CloudTrail

CloudWatch Synthetics admite el registro de las siguientes acciones como eventos en archivos de registro de CloudTrail:

- CreateCanary
- DeleteCanary
- DescribeCanaries

- DescribeCanariesLastRun
- DescribeRuntimeVersions
- GetCanary
- GetCanaryRuns
- ListTagsForResource
- StartCanary
- StopCanary
- TagResource
- UntagResource
- UpdateCanary

Ejemplo: Entradas de archivos de registro de CloudWatch Synthetics

En el siguiente ejemplo, se muestra una entrada de registro de CloudTrail Synthetics que ilustra la acción `DescribeCanaries`.


```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::123456789012:assumed-role/role_name",
        "accountId":"123456789012",
        "accessKeyId":"AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "EX_PRINCIPAL_ID",
                "arn": "arn:aws:iam::111222333444:role/Administrator",
                "accountId":"123456789012",
                "userName": "SAMPLE_NAME"
            },
            "webIdFederationData": {},
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "2020-04-08T21:43:24Z"
            }
        }
```

```
    },
    "eventTime": "2020-04-08T23:06:47Z",
    "eventSource": "synthetics.amazonaws.com",
    "eventName": "DescribeCanaries",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "127.0.0.1",
    "userAgent": "aws-internal/3 aws-sdk-java/1.11.590
 Linux/4.9.184-0.1.ac.235.83.329.metal1.x86_64 OpenJDK_64-Bit_Server_VM/25.212-b03
 java/1.8.0_212 vendor/Oracle_Corporation",
    "requestParameters": null,
    "responseElements": null,
    "requestID": "201ed5f3-15db-4f87-94a4-123456789",
    "eventID": "73ddbd81-3dd0-4ada-b246-123456789",
    "readOnly": true,
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

En el siguiente ejemplo, se muestra una entrada de registro de CloudTrail Synthetics que ilustra la acción `UpdateCanary`.

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::123456789012:assumed-role/role_name",
        "accountId":"123456789012",
        "accessKeyId":"AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
               "type": "Role",
                "principalId": "EX_PRINCIPAL_ID",
                "arn": "arn:aws:iam::111222333444:role/Administrator",
       "accountId":"123456789012",
                "userName": "SAMPLE_NAME"
            },
            "webIdFederationData": {},
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "2020-04-08T21:43:24Z"
            }
        }
```

```
    },
    "eventTime": "2020-04-08T23:06:47Z",
    "eventSource": "synthetics.amazonaws.com",
    "eventName": "UpdateCanary",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "aws-internal/3 aws-sdk-java/1.11.590
 Linux/4.9.184-0.1.ac.235.83.329.metal1.x86_64 OpenJDK_64-Bit_Server_VM/25.212-b03
 java/1.8.0_212 vendor/Oracle_Corporation",
    "requestParameters": {
        "Schedule": {
            "Expression": "rate(1 minute)"
        },
        "name": "sample_canary_name",
        "Code": {
            "Handler": "myOwnScript.handler",
            "ZipFile": "SAMPLE_ZIP_FILE"
        }
    },
    "responseElements": null,
    "requestID": "fe4759b0-0849-4e0e-be71-1234567890",
    "eventID": "9dc60c83-c3c8-4fa5-bd02-1234567890",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

En el siguiente ejemplo, se muestra una entrada de registro de CloudTrail Synthetics que ilustra la acción `GetCanaryRuns`.

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::123456789012:assumed-role/role_name",
        "accountId":"123456789012",
        "accessKeyId":"AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "EX_PRINCIPAL_ID",
                "arn": "arn:aws:iam::111222333444:role/Administrator",
```

```
       "accountId":"123456789012",
                "userName": "SAMPLE_NAME"
            },
            "webIdFederationData": {},
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "2020-04-08T21:43:24Z"
            }
        }
    },
    "eventTime": "2020-04-08T23:06:30Z",
    "eventSource": "synthetics.amazonaws.com",
    "eventName": "GetCanaryRuns",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "127.0.0.1",
    "userAgent": "aws-internal/3 aws-sdk-java/1.11.590
 Linux/4.9.184-0.1.ac.235.83.329.metal1.x86_64 OpenJDK_64-Bit_Server_VM/25.212-b03
 java/1.8.0_212 vendor/Oracle_Corporation",
    "requestParameters": {
        "Filter": "TIME_RANGE",
        "name": "sample_canary_name",
        "FilterValues": [
            "2020-04-08T23:00:00.000Z",
            "2020-04-08T23:10:00.000Z"
        ]
    },
    "responseElements": null,
    "requestID": "2f56318c-cfbd-4b60-9d93-1234567890",
    "eventID": "52723fd9-4a54-478c-ac55-1234567890",
    "readOnly": true,
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

Etiquetado de los recursos de Amazon CloudWatch

Una etiqueta es un atributo personalizado que usted o AWS asignan a un recurso de AWS. Cada etiqueta tiene dos partes:

- Una clave de etiqueta (por ejemplo, `CostCenter`, `Environment` o `Project`). Las claves de etiqueta distinguen entre mayúsculas y minúsculas.
- Un campo opcional denominado valor de etiqueta (por ejemplo, `111122223333` o `Production`). Omitir el valor de etiqueta es lo mismo que utilizar una cadena vacía. Al igual que las claves de etiqueta, los valores de etiqueta distinguen entre mayúsculas y minúsculas.

Las etiquetas le ayudan a hacer lo siguiente:

- Identificar y organizar sus recursos de AWS. Muchos servicios de AWS admiten el etiquetado, por lo que puede asignar la misma etiqueta a los recursos de diferentes servicios para indicar que los recursos están relacionados. Por ejemplo, puede asignar la misma etiqueta a una regla de CloudWatch que asigne a una instancia EC2.

En las siguientes secciones, se ofrece más información acerca de las etiquetas de CloudWatch.

Recursos admitidos en CloudWatch

Los siguientes recursos en CloudWatch admiten el etiquetado:

- Alarmas: puede etiquetar las alarmas por medio del comando tag-resource de la AWS CLI y la API TagResource. También puede ver y gestionar las etiquetas de alarma mediante la página de detalles de Alarmas de la consola CloudWatch.
- Canaries: puede etiquetar canaries con la consola de CloudWatch. Para obtener más información, consulte Creación de un valor controlado.
- Reglas de Contributor Insights: puede etiquetar las reglas de Contributor Insights cuando las cree por medio del comando put-insight-rule de la AWS CLI y la API PutInsightRule. Puede agregar etiquetas a las reglas existentes mediante el uso del comando tag-resource de la AWS CLI y la API TagResource.
- Secuencias métricas: puede etiquetar las secuencias métricas cuando las cree por medio del comando put-metric-stream de la AWS CLI y la API PutMetricStream. Puede agregar etiquetas a

las secuencias métricas existentes mediante el uso del comando tag-resource de la AWS CLI y la API TagResource.

Para obtener información acerca de cómo añadir y administrar etiquetas, consulte Administración de etiquetas.

Administración de etiquetas

Las etiquetas se componen de las propiedades `Value` y `Key` de un recurso. Puede utilizar la consola de CloudWatch, la AWS CLI o la API de CloudWatch para agregar, editar o eliminar los valores de estas propiedades. Para obtener información sobre cómo trabajar con etiquetas, consulte lo siguiente:

- TagResource,UntagResource, y ListTagsForResource en la Referencia de la API de Amazon CloudWatch
- tag-resource, untag-resource, y list-tags-for-resource en la Referencia de la CLI de Amazon CloudWatch
- Working with Tag Editor (Uso del editor de etiquetas) en la Guía del usuario de Resource Groups

Convenciones de nomenclatura y uso de las etiquetas

Las siguientes convenciones básicas de nomenclatura y uso se aplican al uso de etiquetas con recursos de CloudWatch:

- Cada recurso puede tener un máximo de 50 etiquetas.
- Para cada recurso, cada clave de etiqueta debe ser única y solo puede tener un valor.
- La longitud máxima de la clave de etiqueta es de 128 caracteres Unicode en UTF-8.
- La longitud máxima del valor de etiqueta es de 256 caracteres Unicode en UTF-8.
- Los caracteres permitidos son letras, números y espacios representables en UTF-8, además de los siguientes caracteres: . : + = @ _ / - (guion).
- Las claves y los valores de las etiquetas distinguen entre mayúsculas y minúsculas. Como práctica recomendada, decida una estrategia de uso de mayúsculas y minúsculas en las etiquetas e implemente esa estrategia sistemáticamente en todos los tipos de recursos. Por ejemplo, decida si se va a utilizar `Costcenter`, `costcenter` o `CostCenter` y utilice la misma convención para

todas las etiquetas. Procure no utilizar etiquetas similares con un tratamiento de mayúsculas y minúsculas incoherente.

- El prefijo `aws:` está prohibido para las etiquetas, ya que está reservado para su uso por parte de AWS. Las claves y valores de etiquetas que tienen este prefijo no se pueden editar. Las etiquetas que tengan este prefijo no cuentan para el límite de etiquetas por recurso.

Integración de Grafana

Puede utilizar Grafana 6.5.0 y las versiones posteriores para avanzar contextualmente a través de la consola de CloudWatch y consultar una lista dinámica de métricas mediante el uso de comodines. Esto puede ayudarle a monitorear métricas para los recursos de AWS, como instancias o contenedores de Amazon Elastic Compute Cloud (EC2). Cuando se crean nuevas instancias como parte de un evento de Auto Scaling, aparecen automáticamente en el gráfico. No es necesario realizar un seguimiento de los nuevos ID de instancia. Los paneles prediseñados ayudan a simplificar la experiencia de introducción para monitorear Amazon EC2, Amazon Elastic Block Store y para los recursos de AWS Lambda.

Puede utilizar Grafana 7.0 y las versiones posteriores para realizar consultas acerca de CloudWatch Logs Insights en grupos de registro en CloudWatch Logs. Puede visualizar los resultados de la consulta en gráficos de barras, líneas y apilados y en un formato de tabla. Para más información acerca de CloudWatch Logs Insights, consulte Analyzing Log Data with CloudWatch Logs Insights (Análisis de datos de registro con CloudWatch Logs Insights).

Para obtener más información acerca de los primeros pasos a seguir, consulte Uso de AWS CloudWatch en Grafana en la documentación de Grafana Labs.

Consola de CloudWatch para cuentas y Regiones cruzadas

Para obtener la mejor experiencia de observación y detección entre cuentas para sus métricas, registros y trazas, le recomendamos que utilice la observabilidad entre cuentas de CloudWatch. Para obtener más información, consulte Observabilidad entre cuentas de CloudWatch.

CloudWatch también ofrece un panel de control de CloudWatch entre cuentas y regiones. Esta funcionalidad le proporciona la opción de visualizar los paneles, alarmas, métricas y paneles automáticos entre diversas cuentas. En cambio, no proporciona visibilidad entre cuentas para los registros ni las trazas.

Si también usa la observabilidad entre cuentas de CloudWatch, un caso de uso de este panel de CloudWatch entre cuentas consiste en permitir que una de sus cuentas de origen de observabilidad entre cuentas de CloudWatch vea las métricas de otra cuenta de origen.

En el resto de esta sección se describe el panel entre cuentas y regiones. Puede usar esta opción para crear paneles que resuman los datos de CloudWatch de varias cuentas de AWS y regiones de AWS en un mismo panel. También puede crear una alarma en una cuenta que inspeccione una métrica ubicada en una cuenta diferente.

Muchas organizaciones implementan sus recursos de AWS en varias cuentas, con el fin de proporcionar límites de facturación y seguridad. Si este es su caso, le recomendamos que designe una o varias de sus cuentas como cuentas de monitorización y que cree en estas los paneles para diversas cuentas.

La funcionalidad para diversas cuentas se integra en AWS Organizations a fin de ayudarle a crear sus paneles para diversas cuentas de manera eficiente.

Funcionalidad entre regiones

La funcionalidad de Regiones cruzadas se integra ahora de forma automática. No es necesario realizar ningún paso adicional para poder mostrar métricas de distintas regiones de una sola cuenta en el mismo gráfico o panel. La funcionalidad entre regiones no es compatible con las alarmas, por lo que no se puede crear una alarma en una región que observe una métrica en otra región.

Temas

- Desactivación y limpieza después de utilizar cuentas cruzadas

Habilitación de la funcionalidad de cuentas cruzadas en CloudWatch

A fin de configurar la funcionalidad de cuentas cruzadas en la consola de CloudWatch, utilice la consola de CloudWatch para configurar las cuentas de uso compartido y las cuentas de monitoreo.

Configurar una cuenta de uso compartido

Debe habilitar el uso compartido en cada cuenta que vaya a poner datos a disposición de la cuenta de monitorización.

Esto otorgará los permisos de sólo lectura que elija en el paso 5 a todos los usuarios que visualicen un panel de cuentas cruzadas en la cuenta con la que comparte, si el usuario tiene los permisos correspondientes en la cuenta con la que comparte.

Para permitir que su cuenta comparta datos de CloudWatch con otras cuentas

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, seleccione Configuración.
3. En Share your CloudWatch data (Comparta sus datos de CloudWatch), seleccione Configure (Configurar).
4. En Sharing (Uso compartido), elija Specific accounts (Cuentas específicas) y especifique los ID de las cuentas con las que desea compartir datos.

   Cualquier cuenta que especifique aquí podrá ver los datos de CloudWatch de su cuenta. Especifique únicamente los ID de cuentas que conozca y en las que confíe.
5. En Permissions (Permisos), especifique cómo compartir los datos con una de las siguientes opciones:

   - Provide read-only access to your CloudWatch metrics, dashboards, and alarms (Proporcione acceso de solo lectura a las métricas, paneles y alarmas de CloudWatch). Esta opción permite que las cuentas de monitoreo creen paneles para cuentas cruzadas con widgets que contengan datos de CloudWatch de su cuenta.
   - Include CloudWatch automatic dashboards (Incluir paneles automáticos de CloudWatch). Si selecciona esta opción, los usuarios de la cuenta de monitorización también pueden ver

la información en los paneles automáticos de esta cuenta. Para obtener más información, consulte Introducción a Amazon CloudWatch.

- Incluye el acceso de solo lectura de X-Ray para el Mapa de seguimiento de X-Ray. Si selecciona esta opción, los usuarios de la cuenta de monitoreo también pueden visualizar el mapa de seguimiento de X-Ray y la información de seguimiento de X-Ray en esta cuenta. Para obtener más información, consulte Uso del Mapa de seguimiento de X-Ray.
- Full read-only access to everything in your account (Acceso pleno de solo lectura a todo el contenido de la cuenta). Esta opción habilita las cuentas que se utilizan para compartir, de tal forma que se puedan crear paneles para cuentas cruzadas que incluyan widgets que contengan datos de CloudWatch de su cuenta. También permite que esas cuentas busquen más con más detalle en su cuenta y consulten los datos de su cuenta en las consolas de otros servicios de AWS.

6. Elija Launch CloudFormation template (Iniciar plantilla de CloudFormation).

   En la pantalla de confirmación, escriba **`Confirm`** y elija Launch template (Iniciar plantilla).

7. Seleccione la casilla I acknowledge... (Acepto...) y elija Create stack (Crear pila).

Uso compartido con toda la organización

Al completar el procedimiento anterior, se crea un rol de IAM que permite que la cuenta comparta datos con una cuenta. Cree o edite un rol de IAM que comparta los datos con todas las cuentas de una organización. Solo debe hacerlo si conoce y confía en todas las cuentas de la organización.

Otorgará los permisos de sólo lectura enumerados en las políticas que se muestran en el paso 5 del procedimiento anterior a todos los usuarios que visualicen un panel de cuentas cruzadas en la cuenta con la que comparte, si el usuario tiene los permisos correspondientes en la cuenta con la que comparte.

Para compartir los datos de su cuenta de CloudWatch con todas las cuentas de una organización

1. Si aún no lo ha hecho, lleve a cabo el procedimiento anterior para compartir sus datos con una cuenta de AWS.
2. Inicie sesión en la AWS Management Console y abra la consola de IAM en https://console.aws.amazon.com/iam/.
3. Seleccione Roles en el panel de navegación.
4. En la lista de roles, elija CloudWatch-CrossAccountSharingRole.

5. Seleccione Trust relationships (Relaciones de confianza), Edit trust relationship (Editar relaciones de confianza).

   Aparecerá una política como esta:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:root"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

6. Cambie la política como se indica a continuación, reemplazando *org-id* por el ID de su organización.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "*"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:PrincipalOrgID": "org-id"
        }
      }

    }
  ]
}
```

7. Elija Actualizar política de confianza.

Configure una cuenta de monitoreo

Habilite cada una de las cuentas de monitoreo si desea ver los datos de CloudWatch de cuentas cruzadas.

Cuando se completa el siguiente procedimiento, CloudWatch crea un rol vinculado al servicio que CloudWatch utiliza en la cuenta de monitoreo para obtener acceso a los datos que otras cuentas comparten. Este rol vinculado a servicio se denomina AWSServiceRoleForCloudWatchCrossAccount. Para obtener más información, consulte Uso de roles vinculados a servicios para CloudWatch.

Para permitir que su cuenta visualice datos de CloudWatch de cuentas cruzadas

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Settings (Configuración) y, a continuación, en la sección Cross-account cross-region (Regiones y cuentas cruzadas), elija Configure (Configurar).
3. En la sección View cross-account cross-region (Ver cuentas y regiones cruzadas), elija Enable (Habilitar) y, a continuación, seleccione la casilla de verificación Show selector in the console (Mostrar selector en la consola) para permitir que un selector de cuentas aparezca en la consola de CloudWatch cuando se genere un gráfico de una métrica o se cree una alarma.
4. En View cross-account cross-region (Ver diversas cuentas y regiones), elija una de las siguientes opciones:

   - Account Id Input (Especificar ID de cuenta). Esta opción le pide que introduzca manualmente un ID de cuenta cada vez que desee cambiar de cuenta al consultar datos de varias cuentas.
   - Selector de cuentas de AWS Organization. Esta opción hace que aparezcan las cuentas que especificó al llevar a cabo la integración entre cuentas diferentes con Organizations. La próxima vez que utilice la consola, CloudWatch mostrará una lista desplegable de estas cuentas para que pueda seleccionarlas mientras visualiza los datos para cuentas cruzadas.

     Para ello, primero debe haber utilizado la cuenta administrativa de la organización con el fin de permitir que CloudWatch consulte una lista de cuentas de la organización. Para obtener más información, consulte (Opcional) Integración con AWS Organizations.
   - Custom account selector (Selector de cuentas personalizado). Esta opción pide que se introduzca una lista de ID de cuenta. La próxima vez que utilice la consola, CloudWatch mostrará una lista desplegable de estas cuentas para que pueda seleccionarlas mientras consulta los datos de las cuentas cruzadas.

También puede especificar una etiqueta para cada una de estas cuentas, de forma que le resulte más fácil identificarlas al elegir las cuentas que desea consultar.

La configuración del selector de cuentas que realiza un usuario aquí se retiene solo para ese usuario, no para todos los demás usuarios de la cuenta de monitoreo.

5. Elija Habilitar.

Después de completar esta configuración, puede crear paneles para diversas cuentas. Para obtener más información, consulte Paneles para cuentas y Regiones cruzadas.

(Opcional) Integración con AWS Organizations

Si desea integrar la funcionalidad para diversas cuentas con AWS Organizations, debe poner a disposición de las cuentas de monitorización una lista de todas las cuentas de la organización.

Para habilitar la funcionalidad de CloudWatch para cuentas cruzadas con el fin de obtener acceso a una lista de todas las cuentas de la organización

1. Inicie sesión en la cuenta administrativa de la organización.
2. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
3. En el panel de navegación, elija Settings (Configuración). A continuación, elija Configure (Configurar).
4. En Grant permission to view the list of accounts in the organization (Conceder permiso para ver la lista de cuentas de la organización), elija Specific accounts (Cuentas específicas) para que se le pida que especifique la lista de ID de cuenta. La lista de cuentas de la organización se compartirá únicamente con las cuentas que especifique aquí.
5. Elija Share organization account list (Compartir lista de cuentas de la organización).
6. Elija Launch CloudFormation template (Iniciar plantilla de CloudFormation).

   En la pantalla de confirmación, escriba **`Confirm`** y elija Launch template (Iniciar plantilla).

Solución de problemas de la configuración para cuentas cruzadas de CloudWatch

Esta sección contiene sugerencias para solucionar los problemas de implementación de la consola para cuentas cruzadas en CloudWatch.

Recibo errores de acceso denegado al mostrar datos para diversas cuentas

Compruebe lo siguiente:

- Su cuenta de monitorización debe tener un rol denominado AWSServiceRoleForCloudWatchCrossAccount. Si no lo tiene, debe crearlo. Para obtener más información, consulte Set Up a Monitoring Account.
- Cada cuenta de uso compartido debe tener un rol denominado CloudWatch-CrossAccountSharingRole. Si no lo tiene, debe crearlo. Para obtener más información, consulte Set Up A Sharing Account.
- El rol de uso compartido debe confiar en la cuenta de monitorización.

Para confirmar que los roles están configurados correctamente para la consola de CloudWatch para cuentas cruzadas

1. Inicie sesión en la AWS Management Console y abra la consola de IAM en https://console.aws.amazon.com/iam/.
2. Seleccione Roles en el panel de navegación.
3. En la lista de roles, asegúrese de que existe el rol necesario. En una cuenta de uso compartido, busque CloudWatch-CrossAccountSharingRole. En una cuenta de monitorización, busque AWSServiceRoleForCloudWatchCrossAccount.
4. Si está en una cuenta de uso compartido y CloudWatch-CrossAccountSharingRole ya existe, elija CloudWatch-CrossAccountSharingRole.
5. Seleccione Trust relationships (Relaciones de confianza), Edit trust relationship (Editar relaciones de confianza).
6. Confirme que la política muestra el ID de cuenta de la cuenta de monitorización o el ID de organización de una organización que contiene la cuenta de monitorización.

No se ve ningún menú desplegable de cuentas en la consola

En primer lugar, verifique que ha creado los roles de IAM correctos, como se explica en la sección de solución de problemas anterior. Si están configurados correctamente, asegúrese de haber habilitado esta cuenta de tal forma que pueda ver los datos para diversas cuentas, como se describe en Enable Your Account to View Cross-Account Data.

Desactivación y limpieza después de utilizar cuentas cruzadas

Para desactivar la funcionalidad de cuentas cruzadas para CloudWatch, siga estos pasos.

Paso 1: elimine las pilas o roles entre cuentas

El mejor método es eliminar las pilas de AWS CloudFormation que se utilizaron para habilitar la funcionalidad de cuentas diferentes.

- En cada una de las cuentas de uso compartido, elimine la pila CloudWatch-CrossAccountSharingRole.
- Si utilizó AWS Organizations para habilitar la funcionalidad de cuentas diferentes con todas las cuentas de una organización, elimine la pila CloudWatch-CrossAccountListAccountsRole en la cuenta administrativa de la organización.

Si no usó las pilas de AWS CloudFormation para habilitar la funcionalidad de cuentas diferentes, realice lo siguiente:

- En cada una de las cuentas de uso compartido, elimine el rol de IAM CloudWatch-CrossAccountSharingRole.
- Si utilizó AWS Organizations para habilitar la funcionalidad de cuentas diferentes con todas las cuentas de una organización, elimine el rol de IAM CloudWatch-CrossAccountSharing-ListAccountsRole en la cuenta administrativa de la organización.

Paso 2: cree un rol vinculado al servicio

En la cuenta de monitoreo, elimine el rol de IAM vinculado al servicio AWSServiceRoleForCloudWatchCrossAccount.

CloudWatch Service Quotas

CloudWatch aplica las siguientes cuotas a las métricas, alarmas, solicitudes de la API y a las notificaciones de email.

> **Note**
>
> Para algunos servicios de AWS, incluido CloudWatch, puede utilizar las métricas de uso de CloudWatch para visualizar el uso actual de su servicio en los gráficos y paneles de CloudWatch. Puede utilizar una función de cálculo de métricas de CloudWatch para mostrar las cuotas de servicio de esos recursos en los gráficos. También puede configurar alarmas que le avisen cuando su uso se acerque a una cuota de servicio. Para obtener más información, consulte Visualización de las cuotas de servicio y configuración de alarmas.

Recurso	Cuota predeterminada
Acciones de las alarmas	5/alarma. Esta cuota no se puede cambiar.
Periodo de evaluación de la alarma	El valor máximo, calculado multiplicando el periodo de alarma por la cantidad de periodos de evaluación utilizados, es de un día (86 400 segundos). Esta cuota no se puede cambiar.
Alarmas	10 al mes por cliente de forma gratuita. Las alarmas adicionales incurren en cargos. No hay límite en el número total de alarmas por cuenta. Las alarmas basadas en expresiones matemáticas métricas pueden disponer de hasta 10 métricas. 200 alarmas de Información de métricas por región. Puede solicitar un aumento de cuota.
Modelos de detección de anomalías	500 por región y cuenta.
Solicitudes API	1 000 000 al mes por cliente de forma gratuita.

Recurso	Cuota predeterminada
Canaries	200 por región y por cuenta. Puede solicitar un aumento de cuota.
Solicitudes de API de Contributor Insights	Las siguientes API tienen una cuota de 20 transacciones por segundo (TPS) y por región. • DescribeInsightRules Esta cuota no se puede cambiar. • GetInsightRuleReport Puede solicitar un aumento de cuota. Las siguientes API tienen una cuota de 5 TPS por Región. Esta cuota no se puede cambiar. • DeleteInsightRules • PutInsightRule Las siguientes API tienen una cuota de 1 TPS por Región. Esta cuota no se puede cambiar. • DisableInsightRules • EnableInsightRules
Reglas de Contributor Insights	100 reglas por Región y por cuenta. Puede solicitar un aumento de cuota.
Métricas personalizadas	Sin cuota.

Recurso	Cuota predeterminada
Paneles	Hasta 500 widgets por panel. Hasta 500 métricas por widget de panel. Hasta 2500 métricas por panel, en todos los widgets. Estas cuotas incluyen todas las métricas recuperadas para su uso en funciones matemáticas de métricas, incluso si dichas métricas no se muestran en el gráfico. Estas cuotas no pueden cambiarse.
DescribeAlarms	9 transacciones por segundo (TPS) por cuenta por Región. Número máximo de solicitudes de operación que puede realizar por segundo sin que aplique una limitació n. Puede solicitar un aumento de cuota.
Solicitud DeleteAlarms Solicitud DescribeAlarmHistory Solicitud DisableAlarmActions Solicitud EnableAlarmActions Solicitud SetAlarmState	3 TPS por Región para cada una de estas operaciones. Número máximo de solicitudes de operación que puede realizar por segundo sin que aplique una limitación. Estas cuotas no pueden cambiarse.
Solicitud DescribeAlarmsForMetric	9 TPS por Región. Número máximo de solicitudes de operación que puede realizar por segundo sin que aplique una limitación. Esta cuota no se puede cambiar.

Recurso	Cuota predeterminada
Solicitud DeleteDashboards Solicitud GetDashboard Solicitud ListDashboards Solicitud PutDashboard	10 TPS por Región para cada una de estas operaciones. Número máximo de solicitudes de operación que puede realizar por segundo sin que aplique una limitación. Estas cuotas no pueden cambiarse.
PutAnomalyDetector DescribeAnomalyDetectors	10 TPS por Región. Número máximo de solicitudes de operación que puede realizar por segundo sin que aplique una limitación.
DeleteAnomalyDetector	5 TPS por Región. Número máximo de solicitudes de operación que puede realizar por segundo sin que aplique una limitación.
Dimensiones	30 por métrica. Esta cuota no se puede cambiar.

Recurso	Cuota predeterminada
GetMetricData	10 TPS por región para operaciones que incluyen consultas de Metrics Insights. Para operaciones que no incluyen consultas de Metrics Insights, la cuota es de 50 TPS por región. Este es el número máximo de solicitud es de operación que puede realizar por segundo sin que se aplique una limitación. Puede solicitar un aumento de cuota. Para operaciones de `GetMetricData` que incluyen una consulta de Metrics Insights, la cuota es de 4 300 000 puntos de datos por segundo (DPS) durante las 3 horas más recientes. Esto se calcula con respecto al número total de puntos de datos escaneados por la consulta (que no puede incluir más de 10 000 métricas). 180 000 puntos de datos por segundo (DPS) si el `StartTime` que se usa en la solicitud de API es menor o igual a tres horas a partir de la hora actual. 396 000 DPS si el valor `StartTime` es superior a tres horas a partir de la hora actual. Es el número máximo de puntos de datos que puede solicitar por segundo mediante una o varias llamadas a la API sin que se limiten. Esta cuota no se puede cambiar. El valor DPS se calcula en función de los puntos de datos estimados, no de los reales. La estimación de puntos de datos se calcula utilizando el intervalo de tiempo, el período y el período de retención solicitados. Esto significa que si los puntos de datos reales de las métricas solicitadas son dispersos o están vacíos, se sigue produciendo una limitación si los puntos de datos estimados superan la cuota. La cuota de DPS es por Región.

Recurso	Cuota predeterminada
GetMetricData	Una sola llamada a `GetMetricData` puede incluir lo siguiente: • Hasta 500 estructuras de `MetricDataQuery` . • Hasta 100 funciones de `SERVICE_QUOTA()` . • Hasta 100 funciones de `SEARCH()`. • Hasta 5 funciones de `LAMBDA()`. Estas cuotas no se pueden cambiar.
GetMetricStatistics	400 TPS por Región. Número máximo de solicitudes de operación que puede realizar por segundo sin que aplique una limitación. Puede solicitar un aumento de cuota.
GetMetricWidgetImage	Hasta 500 métricas por imagen. Esta cuota no se puede cambiar. 20 TPS por Región. Número máximo de solicitudes de operación que puede realizar por segundo sin que aplique una limitación. Puede solicitar un aumento de cuota.
ListMetrics	25 TPS por Región. Número máximo de solicitudes de operación que puede realizar por segundo sin que aplique una limitación. Puede solicitar un aumento de cuota.
Valores de los datos de las métricas	El valor de un punto de datos de las métricas debe estar dentro del rango de -2^360 a 2^360. No se admiten valores especiales (por ejemplo, NaN, +Infinity, -Infinity). Esta cuota no se puede cambiar.

Recurso	Cuota predeterminada
Elementos MetricDatum	1000 por solicitud de PutMetricData. Un objeto MetricDatum puede contener un único valor o un objeto StatisticSet que representa muchos valores. Esta cuota no se puede cambiar.
Métricas	10 al mes por cliente de forma gratuita.
Consultas de Metrics Insights	Una única consulta puede procesar un máximo de 10 000 métricas. Esto significa que si las cláusulas SELECT (SELECCIONAR), FROM (DESDE) y WHERE (DONDE) coinciden con más de 10 000 métricas, la consulta solo procesará las primeras 10 000 de las métricas encontradas. Una única consulta puede devolver un máximo de 500 series temporales. Solo puede consultar las tres horas de datos más recientes.
Tasas de solicitudes de la API de Observability Access Manager (OAM).	1 TPS por región para PutSinkPolicy. 10 TPS por Región para cada una de las demás API OAM de CloudWatch. Estas cuotas reflejan el número máximo de solicitudes de operación que puede realizar por segundo sin que se aplique una limitación. Estas cuotas no se pueden cambiar.
Enlaces a cuentas de origen de OAM	Cada cuenta de origen se puede vincular a un máximo de cinco cuentas de supervisión Esta cuota no se puede cambiar.

Recurso	Cuota predeterminada
Receptores de OAM	1 receptor por cuenta y por región Esta cuota no se puede cambiar.
Solicitud PutCompositeAlarm	3 TPS por Región. Número máximo de solicitudes de operación que puede realizar por segundo sin que aplique una limitación. Puede solicitar un aumento de cuota.
Solicitud PutMetricAlarm	3 TPS por Región. Número máximo de solicitudes de operación que puede realizar por segundo sin que aplique una limitación. Puede solicitar un aumento de cuota.
Solicitud PutMetricData	1 MB para solicitudes HTTP POST. PutMetricData puede gestionar 500 transacciones por segundo (TPS), que es el número máximo de solicitudes de operaciones puede realizar por segundo sin estar limitadas. `PutMetricData` puede gestionar 1000 métricas por solicitud. Puede solicitar un aumento de cuota.
Notificaciones de email de Amazon SNS	1000 al mes por cliente de forma gratuita.
Grupos de Synthetics	20 por cuenta. Esta cuota no se puede cambiar.
TagResource	20 TPS por Región. Número máximo de solicitudes de operación que puede realizar por segundo sin que aplique una limitación. Esta cuota no se puede cambiar.

Recurso	Cuota predeterminada
UntagResource	20 TPS por Región. Número máximo de solicitudes de operación que puede realizar por segundo sin que aplique una limitación. Esta cuota no se puede cambiar.

Historial de documentos

En la siguiente tabla, se describen los cambios importantes de cada versión de la Guía del usuario de Amazon CloudWatch a partir de junio de 2018. Para recibir notificaciones sobre los cambios en esta documentación, puede suscribirse a una fuente RSS.

Cambio	Descripción	Fecha
La asignación de servicios de CloudWatch Application Signals admite canario, clientes de RUM y agrupaciones de dependencias de servicios de AWS.	La versión preliminar de Application Signals ha agregado agrupaciones predeterminadas en la asignación de servicios para canarios, clientes de RUM y dependencias de servicios de AWS del mismo tipo. Este cambio reduce el número de iconos en la vista predeterminada de la asignación de servicios para facilitar la visualización y la navegación.	21 de mayo de 2024
Actualización de la política de IAM de CloudWatchReadOnlyAccess	CloudWatch cambió el alcance de un permiso en CloudWatchReadOnlyAccess. El alcance de la política agregó las acciones de `application-signals:BatchGet*` , `application-signals:Get*` y `application-signals:List*` para que los usuarios puedan usar CloudWatch Application Signals para ver, investigar y diagnosticar problemas relacionados con el estado	17 de mayo de 2024

	de sus servicios. CloudWatch también agregó una acción de `iam:GetRole` para que los usuarios puedan comprobar si Application Signals está configurada.	
Actualización de la política de IAM de CloudWatchFullAccessV2	CloudWatch cambió el alcance de un permiso en CloudWatchFullAccessV2. El alcance de la política agregó la entrada `application-signals:*` para que los usuarios puedan usar CloudWatch Application Signals para ver, investigar y diagnosticar problemas relacionados con el estado de sus servicios.	17 de mayo de 2024
Lambda Insights admite AWS GovCloud (Este de EE. UU.) y AWS GovCloud (Oeste de EE. UU.)	Lambda Insights de CloudWatch ha agregado compatibilidad con las regiones AWS GovCloud (Este de EE. UU.) y AWS GovCloud (Oeste de EE. UU.).	29 de abril de 2024
La observabilidad entre cuentas de CloudWatch admite los filtros de recursos	Ahora puede crear filtros para especificar qué espacios de nombres de métricas y grupos de registros se comparten entre la cuenta de origen y la cuenta de supervisión al crear el enlace entre las cuentas.	26 de abril de 2024

Actualizaciones de CloudWatch Application Signals	La versión preliminar de Application Signals ha agregado tres características. Actualmente, Application Signals admite aplicaciones Pyhton. Ofrece un proceso de habilitación más sencillo para las aplicaciones en las arquitecturas de Amazon EKS. Además, incluye nuevas configuraciones que puede usar para administrar la cardinalidad de las métricas que se recopilan.	26 de abril de 2024
Información de contenedores de CloudWatch con observabilidad mejorada para Amazon EKS puede recopilar métricas de AWS Elastic Fabric Adapter (EFA)	Ahora puede utilizar Información de contenedores de CloudWatch con observabilidad mejorada para que Amazon EKS recopile métricas de AWS Elastic Fabric Adapter (EFA) de los clústeres de Amazon EKS.	23 de abril de 2024

Política de IAM actualizada	CloudWatch actualizó la política CloudWatchApplicationSignalsServiceRolePolicy. El alcance de los permisos `logs:StartQuery` y `logs:GetQueryResults` de esta política se modificaron para agregar `arn:aws:logs:*:*:log-group:/aws/appsignals/*:*` y `"arn:aws:logs:*:*:log-group:/aws/application-signals/data:*"` para habilitar Application Signals en más arquitecturas. Esta política está asociada al rol vinculado al servicio AWSServiceRoleForCloudWatchApplicationSignals.	18 de abril de 2024
Internet Monitor proporciona un mapa meteorológico mundial de Internet a los clientes autenticados de AWS	Amazon CloudWatch Internet Monitor ahora muestra un mapa meteorológico mundial de Internet que está disponible en la consola para todos los clientes autenticados de AWS. Para ver el mapa, en la consola de Amazon CloudWatch, vaya a Internet Monitor.	16 de abril de 2024

Información de contenedores de CloudWatch con observabilidad mejorada para Amazon EKS puede recopilar métricas de AWS Neuron	Ahora puede usar Información de contenedores de CloudWatch con observabilidad mejorada para que Amazon EKS recopile métricas de AWS Neuron de los clústeres de Amazon EKS.	16 de abril de 2024
CloudWatch Application Signals agrega una pestaña de descripción general del servicio y más métricas para facilitar el diagnóstico	Una nueva pestaña de descripción general del servicio muestra una descripción general de su servicio, que incluye el número de operaciones, las dependencias, los datos sintéticos y las páginas de clientes. La pestaña muestra las métricas clave de todo el servicio, las principales operaciones y las dependencias. Ahora también puede ver los seguimientos de X-Ray que están correlacionados con problemas como fallas, errores y problemas de latencia.	16 de abril de 2024
Información de contenedores de CloudWatch con observabilidad mejorada para Amazon EKS agrega compatibilidad con Windows	Ahora puede usar Información de contenedores de CloudWatch con observabilidad mejorada para que Amazon EKS recopile métricas de nodos de trabajo de Windows en clústeres de Amazon EKS.	10 de abril de 2024

Se actualizó la política de IAM CloudWatchApplicationSignalsServiceRolePolicy	CloudWatch cambió el alcance de un permiso en CloudwatchApplicationSignalsServiceRolePolicy. El alcance del permiso `cloudwatch:GetMetricData` se modificó a * para que Application Signals pueda recuperar métricas de los orígenes de las cuentas vinculadas.	8 de abril de 2024
Amazon CloudWatch Internet Monitor ahora admite la observabilidad entre cuentas	Ya puede usar la observabilidad entre cuentas de Internet Monitor para supervisar las aplicaciones que abarcan varias cuentas de Cuentas de AWS de una sola Región de AWS.	29 de marzo de 2024

Se actualizaron las políticas CloudWatchAgentServerPolicy y CloudWatchAgentAdminPolicy	CloudWatch agregó permisos a las políticas `CloudWatchAgentServerPolicy` y `CloudWatchAgentAdminPolicy` para permitir al agente de CloudWatch publicar seguimientos de X-Ray y modificar los periodos de retención de los grupos de registros. En ambas políticas, se agregaron los permisos `xray:PutTraceSegments`, `xray:PutTelemetryRecords`, `xray:GetSamplingRules`, `xray:GetSamplingTargets`, `xray:GetSamplingStatisticSummaries` y `logs:PutRetentionPolicy`	12 de febrero de 2024

Nuevo rol vinculado a servicios y la política de IAM para CloudWatch Network Monitor	CloudWatch añadió un nuevo rol vinculado a servicios , denominado AWSServic eRoleForNetworkMonitor. CloudWatch añadió este nuevo rol vinculado a servicios para permitirle crear monitores para obtener métricas de red entre las subredes de origen y las direcciones IP de destino. La nueva política de IAM CloudWatchNetworkM onitorServiceRolePolicy se adjunta a este rol, y la política concede permiso a CloudWatc h para obtener métricas de red en su nombre.	22 de diciembre de 2023
CloudWatch lanza Amazon CloudWatch Network Monitor	CloudWatch lanza una nueva característica, Amazon CloudWatch Network Monitor. Se trata de un nuevo servicio de supervisión de red activo que identifica si existe un problema de red en la red de AWS o en la red de su propia empresa.	22 de diciembre de 2023

Actualización de la política CloudWatchReadOnlyAccess	CloudWatch añadió los permisos existentes de solo lectura para CloudWatch Synthetics, X-Ray y CloudWatch RUM y nuevos permisos de solo lectura para CloudWatch Application Signals a CloudWatchReadOnlyAccess para que los usuarios con esta política puedan clasificar y diagnosticar problemas en el estado del servicio notificados por CloudWatch Application Signals. El permiso de `cloudwatch:GenerateQuery` se añadió para que los usuarios con esta política puedan generar una cadena de consulta de Información de métricas de CloudWatch a partir de una petición en lenguaje natural.	5 de diciembre de 2023

Actualización de la política CloudwatchFullAccessV2	CloudWatch añadió los permisos existentes a CloudWatchFullAccessV2 para CloudWatch Synthetics, X-Ray y CloudWatch RUM, y añadió nuevos permisos para CloudWatch Application Signals para que los usuarios con esta política puedan administrar completamente Application Signals para clasificar y diagnosticar problemas relacionados con el estado del servicio.	5 de diciembre de 2023

Nuevo rol vinculado a servicios y nueva política de IAM	CloudWatch añadió un nuevo rol vinculado a servicios , denominado AWSServiceRoleForCloudWatchApplicationSignals. CloudWatch añadió este nuevo rol vinculado a servicios para permitir a CloudWatch Application Signals recopilar datos de Registros de CloudWatch, datos del seguimiento de X-Ray, datos de métricas de CloudWatch y datos del etiquetado de las aplicaciones que haya habilitado para CloudWatch Application Signals. La nueva política de IAM CloudWatchApplicationSignalsServiceRolePolicy se adjunta a este rol, y la política otorga permiso a CloudWatch Application Signals para recopilar datos de supervisión y etiquetado de otros servicios relevantes de AWS.	30 de noviembre de 2023

CloudWatch lanza la versión preliminar de Application Signals	CloudWatch Application Signals está en versión preliminar. Utilice Application Signals para instrumentar sus aplicaciones en AWS de forma que pueda supervisar el estado actual de las aplicaciones, crear objetivos de nivel de servicio (SLO) y realizar un seguimiento del rendimiento de las aplicaciones a largo plazo en comparación con sus objetivos empresariales. Para obtener más información, consulte Application Signals.	30 de noviembre de 2023
CloudWatch añade compatibilidad para consultar otros orígenes de datos	Puede usar CloudWatch para consultar, visualizar y crear alarmas para métricas de otros orígenes de datos. Para obtener más información, consulte Consultar métricas de otros orígenes de datos.	26 de noviembre de 2023
Compatibilidad de la Información de métricas de CloudWatch con la creación de consultas en lenguaje natural	La Información de métricas de CloudWatch admite la creación y actualización de consultas en lenguaje natural. Para obtener más información, consulte Usar lenguaje natural para generar y actualizar consultas de la Información de métricas de CloudWatch.	26 de noviembre de 2023

CloudWatch lanza Información de contenedores con observabilidad mejorada para Amazon EKS	CloudWatch lanzó una nueva versión de Información de contenedores. Esta versión admite la observabilidad mejorada de los clústeres de Amazon EKS y permite recopilar métricas más detalladas de los clústeres que ejecutan Amazon EKS. Tras la instalación, recopila automáticamente registros detallados de telemetría de infraestructura y de contenedores para sus clústeres de Amazon EKS. A continuación, puede utilizar paneles seleccionados y de uso inmediato para profundizar en la telemetría de aplicaciones e infraestructuras.	6 de noviembre de 2023
Los flujos métricos de CloudWatch agregan la configuración rápida de socios	Los flujos métricos de CloudWatch ahora ofrecen una opción de configuración rápida de socios, que puede utilizar para configurar rápidamente un flujo métrico para algunos proveedores externos.	17 de octubre de 2023

CloudWatch lanza las recomendaciones de alarmas	CloudWatch Synthetics ahora ofrece recomendaciones de alarmas para métricas de otros servicios de AWS. Estas recomendaciones pueden ayudar a identific ar las métricas para las que debe configurar las alarmas y así seguir las prácticas recomendadas de supervisión de estos servicios.	16 de octubre de 2023
CloudWatch Synthetics lanza el tiempo de ejecución syn-nodejs-puppeteer-6.0	CloudWatch Synthetics ha lanzado una versión nueva de tiempo de ejecución `syn-nodejs-puppeteer-6.0` .	26 de septiembre de 2023
Añade compatibilidad con Información de aplicaciones de Amazon CloudWatch	Ahora puede compartir las aplicaciones de Información de aplicaciones de CloudWatc h a través de los límites de las cuentas.	26 de septiembre de 2023

Nuevo rol vinculado a servicios y nueva política de IAM	CloudWatch agregó un nuevo rol vinculado a un servicio, llamado AWSServiceRoleForCloudWatchMetrics_DbPerfInsights. CloudWatch agregó este nuevo rol vinculado a servicios para permitir que CloudWatch obtenga métricas de Performance Insights para alarmas, detección de anomalías e instantáneas. La nueva política de IAM de AWSServiceRoleForCloudWatchMetrics_DBPerfInsightsServiceRolePolicy está asociada a esta función, y la política otorga permiso a CloudWatch para obtener las métricas de Performance Insights en nombre del usuario.	20 de septiembre de 2023
Añade una nueva función matemática métrica	CloudWatch agregó una nueva función matemática de métricas, `DB_PERF_INSIGHTS` , que puede usar para obtener métricas de Performance Insights de los servicios de AWS bases de datos para alarmas, detección de anomalías e instantáneas.	20 de septiembre de 2023

Actualización de la política CloudWatchReadOnlyAccess	CloudWatch agregó el permiso `application-autoscaling:DescribeScalingPolicies` a CloudWatchReadOnlyAccess para que los usuarios con esta política puedan acceder a la información sobre las políticas de escalado automático de aplicaciones.	14 de septiembre de 2023
El agente CloudWatch agregó compatibilidad con AL2023	El agente CloudWatch es compatible con AL2023.	8 de agosto de 2023
Nueva política de IAM gestionada, CloudWatchFullAccessV2	CloudWatch agregó una nueva política CloudWatchFullAccessV2. Esta política otorga acceso total a las acciones y los recursos de CloudWatch y, al mismo tiempo, amplía el alcance de los permisos concedidos a otros servicios, como Amazon SNS y. Amazon EC2 Auto Scaling	1 de agosto de 2023
Rol vinculado a servicios para Amazon CloudWatch Internet Monitor: actualización de una política existente	Agrega nuevos permisos, `elasticloadbalancing:DescribeLoadBalancers yec2:DescribeNetworkInterfaces` , al rol vinculado al servicio de Internet Monitor, permite supervisar el tráfico de recursos específicos de Equilibrador de carga de red.	25 de julio de 2023

Se agregó compatibilidad con los recursos de Equilibrador de carga de red en Amazon CloudWatch Internet Monitor	Añade compatibilidad con la creación de un monitor en Internet Monitor con recursos específicos de Equilibrador de carga de red, a fin de proporcionar niveles de observabilidad más detallados para la aplicación.	25 de julio de 2023
Característica de variables del panel	CloudWatch publicó variables de panel, que puede usar para crear paneles flexibles que pueden mostrar rápidamente diferentes contenidos en función de cómo configure un campo de entrada en el panel. Por ejemplo, puede crear un panel que pueda cambiar rápidamente entre distintas funciones de Lambda o ID de instancia de Amazon EC2, o uno que pueda cambiar a distintas regiones de AWS. Para obtener más información, consulte Crear paneles de control flexibles con variables de panel.	28 de junio de 2023

Internet Monitor ahora permite personalizar el umbral de eventos de salud	Internet Monitor agregó la capacidad de personalizar el umbral para cuando una puntuación de rendimiento global o de disponibilidad desencadena un evento de salud. Para obtener más información, consulte Seguimiento del rendimiento y la disponibilidad en tiempo real en Amazon CloudWatch Internet Monitor.	26 de junio de 2023
Internet Monitor ahora es compatible con todas las regiones comerciales	Internet Monitor ha añadido siete nuevas Regiones de AWS y ahora es compatibl e con todas las regiones comerciales.	19 de junio de 2023
Nuevas versiones de la extensión Lambda Insights	CloudWatch agregó la versión 1.0.229.0 de la extensión Lambda Insights para las plataformas x86-64 y ARM64. Para obtener más información, consulte Versiones disponibl es de la extensión Lambda Insights.	12 de junio de 2023

Actualización de la política CloudWatchReadOnlyAccess	CloudWatch agregó permisos a CloudWatchReadOnly Access. Se agregaron los permisos `logs:StartLiveTail` y `logs:StopLiveTail` a fin de que los usuarios con esta política puedan usar la consola para iniciar y detener las sesiones de seguimiento en tiempo real de Registros de CloudWatch. Para obtener más información, consulte Use live tail to view logs in near real time.	6 de junio de 2023
CloudWatch RUM añade compatibilidad con métricas personalizadas	Puede utilizar los monitores de aplicaciones CloudWatch RUM para crear métricas personalizadas y enviarlas a CloudWatch y CloudWatch Evidently. Esta característica incluye una actualización de la política de IAM administrada de AmazonCloudWatchRUMServiceRolePolicy. En esa política, se cambió una clave de condición para que CloudWatch RUM pudiera enviar métricas personalizadas a espacios de nombres de métricas personalizadas.	9 de febrero de 2023

Políticas gestionadas nuevas y actualizadas para CloudWatch	Para permitir la observabilidad entre cuentas de CloudWatch, se han actualizado las políticas de `CloudWatchFullAccess` y `CloudWatchReadOnlyAccess` , y se han agregado las siguientes políticas gestionadas nuevas: `CloudWatchCrossAccountSharingConfiguration` , `OAMFullAccess` , y `OAMReadOnlyAccess` . Para obtener más información, consulte Actualizaciones de CloudWatch a las políticas administradas de AWS.	7 de febrero de 2023
Actualizaciones de la política de rol vinculado a un servicio de Información de aplicaciones de CloudWatch: actualizar a una política existente.	Información de aplicaciones de CloudWatch actualizó una política de rol vinculada al servicio de AWS existente.	19 de diciembre de 2022
Compatibilidad con Información de aplicaciones de Amazon CloudWatch para aplicaciones y microservicios en contenedores desde la consola de Información de contenedores.	Puede mostrar los problemas detectados de Información de aplicaciones para Amazon ECS y Amazon EKS en el panel de Información de contenedores.	17 de noviembre de 2021
Supervisión de Información de aplicaciones de Amazon CloudWatch para bases de datos SAP HANA.	Puede supervisar las bases de datos SAP HANA con Información de aplicaciones.	15 de noviembre de 2021

Compatibilidad de Información de aplicaciones de Amazon CloudWatch para supervisar todos los recursos de una cuenta.	Puede incorporar y supervisar todos los recursos de una cuenta.	15 de septiembre de 2021
Compatibilidad de Información de aplicaciones de Amazon CloudWatch para Amazon FSx.	Puede supervisar las métricas que se recuperaron de Amazon FSx.	31 de agosto de 2021
Las métricas de SDK ya no son compatibles.	CloudWatch SDK Metrics ya no es compatible.	25 de agosto de 2021
Compatibilidad de Información de aplicaciones de Amazon CloudWatch para configurar la supervisión de contenedores.	Supervise los contenedores mediante las prácticas recomendadas con Información de aplicaciones de Amazon CloudWatch.	18 de mayo de 2021
Las secuencias de métricas están disponibles de forma general	Puede utilizar flujos métricos para transmitir continuamente las métricas de CloudWatch a un destino de su elección. Para obtener más información, consulte Metric streams (Flujos métricos) en la Guía del usuario de Amazon CloudWatch.	31 de marzo de 2021
Supervisión de Información de aplicaciones de Amazon CloudWatch para bases de datos Oracle en Amazon RDS y Amazon EC2.	Supervise las métricas y los registros que se recuperaron de Oracle con Información de aplicaciones de Amazon CloudWatch.	16 de enero de 2021

Lambda Insights está disponible de forma general	CloudWatch Lambda Insights es una solución de supervisión y solución de problemas para aplicaciones sin servidor que se ejecutan en AWS Lambda. Para obtener más información, consulte Using Lambda Insights (Uso de Lambda Insights) en la Guía del usuario de Amazon CloudWatch.	3 de diciembre de 2020
Supervisión de Información de aplicaciones de Amazon CloudWatch para las métricas de Prometheus de JMX exporter.	Supervise las métricas que se recuperaron de Prometheus JMX Exporter con Información de aplicaciones de Amazon CloudWatch.	20 de noviembre de 2020
CloudWatch Synthetics lanza una nueva versión de tiempo de ejecución	CloudWatch Synthetics ha lanzado una versión nueva de tiempo de ejecución. Para obtener más información, consulte valor controlado Runtime Versions (Versiones de tiempo de ejecución de los valores controlados) en la Guía del usuario de Amazon CloudWatch.	11 de septiembre de 2020
Supervisión de Información de aplicaciones de Amazon CloudWatch para Postgre SQL en Amazon RDS y Amazon EC2.	Supervise las aplicaciones que se crearon con PostgreSQL y que se ejecutan en Amazon RDS o Amazon EC2.	11 de septiembre de 2020

CloudWatch admite el uso compartido de paneles	Ahora puede compartir paneles de CloudWatch con personas ajenas a la organización y a la cuenta de AWS. Para obtener más información, consulte Sharing CloudWatch Dashboards (Uso compartido de paneles de CloudWatch) en la Guía del usuario de Amazon CloudWatch.	10 de septiembre de 2020
Configuración de supervisión para aplicaciones .NET con SQL Server en el backend con Información de aplicaciones de CloudWatch	Puede utilizar el tutorial de documentación para ayudarle a configurar supervisiones para aplicaciones .NET con SQL Server en el backend con CloudWatch Application Insights.	19 de agosto de 2020
Compatibilidad de AWS CloudFormation con aplicaciones de Información de aplicaciones de Amazon CloudWatch.	Puede agregar la supervisión de Información de aplicaciones de CloudWatch, incluidas las métricas clave y la telemetría, a su aplicación, base de datos y servidor web, directamente desde las plantillas de AWS CloudFormation.	30 de julio de 2020
Supervisión de Información de aplicaciones de Amazon CloudWatch para Aurora para clústeres de base de datos MySQL.	Supervise Aurora para clústeres de base de datos MySQL (RDS Aurora) con Información de aplicaciones de Amazon CloudWatch.	2 de julio de 2020

Disponibilidad general de Información de colaboradores de CloudWatch	CloudWatch Contributor Insights ya está disponible de forma general. Te permite analizar datos de registro y crear series temporales que muestren datos de colaboradores. Puede ver métricas acerca de los colaboradores Top-N, el número total de colaboradores únicos y su uso. Para obtener más información, consulte Uso de Contributor Insights para analizar datos de alta cardinalidad en la Guía del usuario de Amazon CloudWatch.	2 de abril de 2020
Versión preliminar pública de CloudWatch Synthetics	CloudWatch Synthetics ya está disponible en vista previa pública. Te permite crear canarios para supervisar tus puntos de conexión y API. Para obtener más información, consulte Using Canaries (Uso de canaries) en la Guía del usuario de Amazon CloudWatch.	25 de noviembre de 2019

Versión preliminar pública de Información de colaboradores de CloudWatch	CloudWatch Contributor Insights ya está disponible en vista previa pública. Te permite analizar datos de registro y crear series temporales que muestren datos de colaboradores. Puede ver métricas acerca de los colaboradores Top-N, el número total de colaboradores únicos y su uso. Para obtener más información, consulte Uso de Contributor Insights para analizar datos de alta cardinalidad en la Guía del usuario de Amazon CloudWatch.	25 de noviembre de 2019
CloudWatch lanza la característica ServiceLens	ServiceLens mejora la observabilidad de tus servicios y aplicaciones al permitirte integrar seguimientos, métricas, registros y alarmas en un solo lugar. ServiceLens integra CloudWatch con AWS X-Ray para proporcionar una visión de extremo a extremo de la aplicación.	21 de noviembre de 2019

Uso de CloudWatch para administrar de forma proactiva AWS Service Quotas	Utilice CloudWatch para administrar de forma proactiva las cuotas de servicio de AWS Las métricas de uso de CloudWatch le proporcionan visibilidad acerca del uso de los recursos de su cuenta y de las operaciones de la API. Para obtener más informaci ón, consulte Service Quotas Integration and Usage Metrics (Métricas de integración y de uso de Service Quotas) en la Guía del usuario de Amazon CloudWatch.	19 de noviembre de 2019
CloudWatch envía eventos cuando las alarmas cambian de estado	CloudWatch ahora envía un evento a Amazon EventBrid ge cuando alguna alarma de CloudWatch cambia el estado. Para obtener más informaci ón, consulte Alarm Events and EventBridge (Eventos de alarmas y EventBridge) en la Guía del usuario de Amazon CloudWatch.	8 de octubre de 2019

Información de contenedores	Información de contenedores de CloudWatch ya está disponible de forma general. Le permite recopilar, agregar y resumir métricas y registros de sus aplicaciones en contenedores y microservicios. Para obtener más información, consulte Uso de Información de contenedores en la Guía del usuario de Amazon CloudWatch.	30 de agosto de 2019
Actualizaciones de las métricas de vista previa de Información de contenedores en Amazon EKS y en Kubernetes	Se ha actualizado la vista previa pública de Información de contenedores en Amazon EKS y Kubernetes. InstanceId se incluye ahora como una dimensión para las instancias EC2 del clúster. Esto permite que las alarmas de que se han creado en estas métricas activen las siguientes acciones de EC2: Stop (Detener), Terminate (Terminar), Reboot (Reiniciar) o Recover (Recuperar). Además, el espacio de nombres de Kubernetes notifica ahora las métricas de pod y servicio para simplificar la supervisión y las alarmas en las métricas por espacio de nombres.	19 de agosto de 2019

Actualizaciones para la integración de AWS Systems Manager OpsCenter	Actualizaciones acerca de la integración de Información de aplicaciones de CloudWatch con Centro de operaciones de Systems Manager.	7 de agosto de 2019
Métricas de uso de CloudWatch	Las métricas de uso de CloudWatch le ayudan a realizar un seguimiento del uso de los recursos de CloudWatch y a mantenerse dentro de los límites del servicio. Para obtener más información, consulte https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/CloudWatch-Usage-Metrics.html.	6 de agosto de 2019
Versión preliminar pública de Información de contenedores de CloudWatch	Información de contenedores de CloudWatch ahora se encuentra en vista previa pública. Le permite recopilar, agregar y resumir métricas y registros de sus aplicaciones en contenedores y microservicios. Para obtener más información, consulte Uso de Información de contenedores en la Guía del usuario de Amazon CloudWatch.	9 de julio de 2019

Versión preliminar pública de Detección de anomalías de CloudWatch	La detección de anomalías de CloudWatch se encuentra en vista previa pública. CloudWatch aplica algoritmos de machine learning a los datos anteriores de una métrica para crear un modelo de valores esperados de la métrica. Puede utilizar este modelo para la visualización y para la configuración de alarmas. Para obtener más información, consulte Using CloudWatch Anomaly Detection (Uso de la detección de anomalías de CloudWatch) en la Guía del usuario de Amazon CloudWatch.	9 de julio de 2019
Información de aplicaciones de CloudWatch para .NET y SQL Server	Información de aplicaciones de CloudWatch para .NET y SQL Server permite la observabilidad de aplicaciones basadas en .NET y SQL Server. Puede ayudarlo a configurar los mejores monitores para su aplicación con el fin de analizar de forma continua los datos en busca de señales que indiquen problemas con las aplicaciones.	21 de junio de 2019

Se ha reorganizado la sección de los agentes de CloudWatch	La documentación del agente CloudWatch se ha reescrito para mejorar la claridad, especialmente para los clientes que utilizan la línea de comandos para instalar y configurar el agente. Para obtener más información, consulte Collecting Metrics and Logs from Amazon EC2 Instances and On-Premises Servers with the CloudWatch Agent (Recopilación de métricas y registros de instancias de Amazon EC2 y servidores en las instalaciones con el agente de CloudWatch) en la Guía del usuario de Amazon CloudWatch.	28 de marzo de 2019
Se agregó la función SEARCH a las expresiones matemáticas de las métricas	Ahora puede utilizar una función SEARCH en expresiones matemáticas de las métricas. Esto le permite crear paneles que se actualicen de forma automática a medida que se crean nuevos recursos que coincidan con la consulta de búsqueda. Para obtener más información, consulte Using Search Expressions in Graphs (Uso de expresiones de búsqueda en gráficos) en la Guía del usuario de Amazon CloudWatch.	21 de marzo de 2019

Métricas de SDK de AWS para Enterprise Support	SDK Metrics le ayuda a evaluar el estado de los servicios de AWS y a diagnosticar la latencia derivada de llegar a los límites de uso de la cuenta o provocada por una interrupción del servicio. Para obtener más información, consulte Supervisión de aplicaciones mediante métricas del SDK de AWS en la Guía del usuario de Amazon CloudWatch.	11 de diciembre de 2018
Alarmas en expresiones matemáticas	CloudWatch admite la creación de alarmas en función de expresiones matemáticas métricas. Para obtener más información, consulte Alarms on Math Expressions (Alarmas en expresiones matemáticas) en la Guía del usuario de Amazon CloudWatch.	20 de noviembre de 2018

Nueva página principal de la consola de CloudWatch	Amazon ha creado una nueva página de inicio en la consola de CloudWatch, que muestra automáticamente las métricas y las alarmas clave de todos los servicios de AWS que está utilizando. Para obtener más información, consulte Getting Started with Amazon CloudWatch (Introducción a Amazon CloudWatch) en la Guía del usuario de Amazon CloudWatch.	19 de noviembre de 2018
Plantillas de AWS CloudFormation para CloudWatch Agent	Amazon ha cargado plantillas de AWS CloudFormation que puede utilizar para instalar y actualizar el agente de CloudWatch. Para obtener más información, consulte Instalación del agente de CloudWatch en instancias nuevas mediante AWS CloudFormation en la Guía del usuario de Amazon CloudWatch.	9 de noviembre de 2018

Mejoras en el agente de CloudWatch	El agente de CloudWatch se ha actualizado para que funcione con los protocolos StatsD y collectd. También ha mejorado la compatibilidad entre cuentas. Para obtener más información, consulte Recuperación de las métricas personalizadas con StatsD, Recuperación de las métricas personalizadas con collectd y Envío de métricas y registros a una cuenta diferente de AWS en la Guía del usuario de Amazon CloudWatch.	28 de septiembre de 2018
Compatibilidad con puntos de conexión de Amazon VPC	Ahora puede establecer una conexión privada entre su VPC y CloudWatch. Para obtener más información, consulte Using CloudWatch with Interface VPC Endpoints (Uso de CloudWatch con los puntos de enlace de la VPC) en la Guía del usuario de Amazon CloudWatch.	28 de junio de 2018

En la siguiente tabla se describen cambios importantes que se realizaron en la Guía del usuario de Amazon CloudWatch antes de junio de 2018.

Cambio	Descripción	Fecha de lanzamiento de la nueva versión
Cálculos de métricas	Ahora puede utilizar expresiones matemáticas en las métricas de CloudWatch, lo que genera una nueva serie temporal que puede agregar a los	4 de abril de 2018

Cambio	Descripción	Fecha de lanzamiento de la nueva versión
	gráficos del panel. Para obtener más información, consulte Uso de la calculadora de métricas.	
Alarmas “M de N”	Ahora puede configurar una alarma para disparars e basándose en “M de N” puntos de datos en cualquier intervalo de evaluación de alarma. Para obtener más información, consulte Evaluación de una alarma.	8 de diciembre de 2017
Agente de CloudWatch	Se ha publicado un nuevo agente de CloudWatch unificado. Puede utilizar el agente multiplataforma unificado para recopilar las métricas personali zadas del sistema y los archivos de registro desde las instancias de Amazon EC2 y los servidores en las instalaciones. El nuevo agente es compatible con Windows y Linux, y permite la personalización de las métricas recopiladas, incluidas las métricas de subrecurso como los núcleos por CPU. Para obtener más información, consulte Recopile las métricas, registros y seguimientos con el agente de CloudWatch.	7 de septiembre de 2017
Métricas de gateway NAT	Se han agregado métricas para la gateway NAT de Amazon VPC	7 de septiembre de 2017
Métricas de alta resolución	A partir de ahora, puede configurar opcionalm ente métricas personalizadas como métricas de alta resolución, con una granularidad de solo un segundo. Para obtener más información, consulte Métricas de alta resolución.	26 de julio de 2017
API de panel	A partir de ahora, puede crear, modificar y eliminar paneles mediante las API y la AWS CLI. Para obtener más información, consulte Creación de un panel de CloudWatch.	6 de julio de 2017

Cambio	Descripción	Fecha de lanzamiento de la nueva versión
Métricas de AWS Direct Connect	Se han agregado métricas para AWS Direct Connect.	29 de junio de 2017
Métricas de la VPN de Amazon VPC	Se han agregado métricas para la VPN de Amazon VPC	15 de mayo de 2017
Métricas de AppStream 2.0	Se han agregado métricas para AppStream 2.0.	8 de marzo de 2017
Selector de color de la consola de CloudWatch	A partir de ahora, puede elegir el color de cada métrica en sus widgets del panel. Para obtener más información, consulte Edite un gráfico en un panel de CloudWatch.	27 de febrero de 2017
Alarmas en los paneles	Las alarmas se pueden agregar ahora a los paneles. Para obtener más información, consulte Agregue o elimine un widget de alarma desde un panel de CloudWatch.	15 de febrero de 2017
Se han agregado métricas para Amazon Polly	Se han agregado métricas para Amazon Polly.	1 de diciembre de 2016
Se añadieron métricas para Amazon Managed Service para Apache Flink	Se añadieron métricas para Amazon Managed Service para Apache Flink.	1 de diciembre de 2016

Cambio	Descripción	Fecha de lanzamiento de la nueva versión
Se ha agregado compatibilidad para las estadísticas de percentil	Puede especificar cualquier percentil con hasta dos decimales (por ejemplo, p95.45). Para obtener más información, consulte Percentiles.	17 de noviembre de 2016
Se han agregado métricas para Amazon Simple Email Service	Se han agregado métricas para Amazon Simple Email Service.	2 de noviembre de 2016
Se han actualizado las métricas de retención	Amazon CloudWatch ahora conserva los datos de métricas durante 15 meses en lugar de 14 días.	1 de noviembre de 2016
Se ha actualizado la interfaz de la consola de métricas	La consola de CloudWatch se ha actualizado con mejoras en la funcionalidad existente y en nuevas funcionalidades.	1 de noviembre de 2016
Se han agregado métricas para Amazon Elastic Transcoder	Se han agregado métricas para Amazon Elastic Transcoder.	20 de septiembre de 2016
Se han agregado métricas para Amazon API Gateway	Se han agregado métricas para Amazon API Gateway.	9 de septiembre de 2016
Se han agregado métricas para AWS Key Management Service	Se han agregado métricas para AWS Key Management Service.	9 de septiembre de 2016

Cambio	Descripción	Fecha de lanzamiento de la nueva versión
Se han agregado métricas para los nuevos balanceadores de carga de aplicaciones compatibles con Elastic Load Balancing	Se han agregado métricas para los balanceadores de carga de aplicaciones.	11 de agosto de 2016
Se han añadido nuevas métricas NetworkPacketsIn y NetworkPacketsOut para Amazon EC2	Se han añadido nuevas métricas NetworkPacketsIn y NetworkPacketsOut para Amazon EC2.	23 de marzo de 2016
Se han agregado métricas nuevas para la flota de spot de Amazon EC2	Se han agregado métricas nuevas para la flota de spot de Amazon EC2.	21 de marzo de 2016
Se han agregado métricas nuevas de CloudWatch Logs	Se han agregado métricas nuevas de CloudWatch Logs.	10 de marzo de 2016

Cambio	Descripción	Fecha de lanzamiento de la nueva versión
Se agregaron Amazon OpenSearch Service y métricas y dimensiones de AWS WAF	Se agregaron Amazon OpenSearch Service y métricas y dimensiones de AWS WAF.	14 de octubre de 2015
Se ha agregado compatibilidad con los paneles de CloudWatch	Los paneles son páginas de inicio personalizables en la consola de CloudWatch que puede utilizar para supervisar los recursos en una única vista, incluso aquellos que se esparcen entre regiones diferentes. Para obtener más información, consulte Uso de paneles de Amazon CloudWatch.	8 de octubre de 2015
Se han añadido métricas y dimensiones de AWS Lambda	Se han añadido métricas y dimensiones de AWS Lambda.	4 de septiembre de 2015
Se han agregado métricas y dimensiones de Amazon Elastic Container Service	Se han agregado métricas y dimensiones de Amazon Elastic Container Service.	17 de agosto de 2015
Se han agregado métricas y dimensiones de Amazon Simple Storage Service	Se han agregado métricas y dimensiones de Amazon Simple Storage Service.	26 de julio de 2015

Cambio	Descripción	Fecha de lanzamiento de la nueva versión
Nueva característica: acción de alarma de reinicio	Se ha añadido la acción de alarma de reinicio y nuevo rol de IAM para su uso con acciones de alarma. Para obtener más información, consulte Crear alarmas para detener, terminar, reiniciar o recuperar una instancia EC2.	23 de julio de 2015
Se han agregado métricas y dimensiones de Amazon WorkSpaces	Se han agregado métricas y dimensiones de Amazon WorkSpaces.	30 de abril de 2015
Se han agregado métricas y dimensiones de Amazon Machine Learning	Se han agregado métricas y dimensiones de Amazon Machine Learning.	9 de abril de 2015
Nueva característica: acciones de recuperación para las alarmas de la instancia de Amazon EC2	Acciones de alarma actualizadas para incluir nueva acción de recuperación de instancias EC2. Para obtener más información, consulte Crear alarmas para detener, terminar, reiniciar o recuperar una instancia EC2.	12 de marzo de 2015
Se han añadido métricas y dimensiones de Amazon CloudFront y Amazon CloudSearch	Se han añadido métricas y dimensiones de Amazon CloudSearch y Amazon CloudFront.	6 de marzo de 2015

Cambio	Descripción	Fecha de lanzamiento de la nueva versión
Se han agregado métricas y dimensiones de Amazon Simple Workflow Service	Se han agregado métricas y dimensiones de Amazon Simple Workflow Service.	9 de mayo de 2014
Se ha actualizado la guía para agregar compatibilidad con AWS CloudTrail	Se agregó un tema nuevo en el que se detalla cómo utilizar AWS CloudTrail para registrar actividad en Amazon CloudWatch. Para obtener más información, consulte Registro de llamadas a la API de Amazon CloudWatch con AWS CloudTrail.	30 de abril de 2014
Se ha actualizado la guía para utilizar el nuevo AWS Command Line Interface (AWS CLI)	La CLI de AWS es una CLI entre servicios con una instalación simplificada, configuración unificada y sintaxis de línea de comando coherente. La CLI de AWS se admite en Linux/Unix, Windows y Mac. Los ejemplos de CLI en esta guía se han actualizado para utilizar la nueva CLI de AWS. Si desea obtener más información acerca de cómo instalar y configurar la nueva CLI de AWS, consulte Getting Set Up with the AWS CLI Interface (Introducción a la configuración de la interfaz) en la AWS Command Line Interface User Guide (Guía del usuario).	21 de febrero de 2014
Se agregaron Amazon Redshift y métricas y dimensiones de AWS OpsWorks.	Se agregaron Amazon Redshift y métricas y dimensiones de AWS OpsWorks.	16 de julio de 2013

Cambio	Descripción	Fecha de lanzamiento de la nueva versión
Se han agregado métricas y dimensiones de Amazon Route 53	Se han agregado métricas y dimensiones de Amazon Route 53.	26 de junio de 2013
Nueva característica: acciones de la alarma de Amazon CloudWatch	Se ha agregado una nueva sección para documentar las acciones de alarma de Amazon CloudWatch, que se puede utilizar para detener o terminar una instancia de Amazon Elastic Compute Cloud (EC2). Para obtener más información, consulte Crear alarmas para detener, terminar, reiniciar o recuperar una instancia EC2.	8 de enero de 2013
Métricas de EBS actualizadas	Se han actualizado las métricas de EBS para incluir dos nuevas métricas para volúmenes de IOPS aprovisionados.	20 de noviembre de 2012
Nuevas alertas de facturación	Ahora puede supervisar los cargos de AWS con las métricas de Amazon CloudWatch y crear alarmas para que se le notifique cuando haya superado el umbral especificado. Para obtener más información, consulte Crear una alarma de facturación para supervisar los cargos estimados de AWS.	10 de mayo de 2012
Nuevas métricas	Ahora puede obtener acceso a seis métricas nuevas de Elastic Load Balancing que proporcionan recuentos de distintos códigos de respuesta de HTTP.	19 de octubre de 2011
Nueva característica	Ahora puede acceder a las métricas desde Amazon EMR.	30 de junio de 2011

Cambio	Descripción	Fecha de lanzamiento de la nueva versión
Nueva característica	Ahora puede acceder a las métricas de Amazon Simple Notification Service y Amazon Simple Queue Service.	14 de julio de 2011
Nueva característica	Se ha agregado información sobre el uso del API `PutMetricData` para publicar métricas personalizadas. Para obtener más información, consulte Publicar métricas personalizadas de .	10 de mayo de 2011
Se han actualizado las métricas de retención	Amazon CloudWatch ahora conserva el historial de una alarma durante dos semanas en lugar de seis semanas. Con este cambio, el periodo de retención de alarmas coincide con el periodo de retención de datos de métricas.	7 de abril de 2011
Nueva característica	Se ha agregado la posibilidad de enviar notificaciones de Amazon Simple Notification Service o de Auto Scaling cuando una métrica ha superado un umbral. Para obtener más información, consulte Alarmas.	2 de diciembre de 2010
Nueva característica	Una serie de acciones de CloudWatch ahora incluye los parámetros MaxRecords y NextToken , que le permiten controlar páginas de resultados para mostrar.	2 de diciembre de 2010
Nueva característica	Este servicio ahora se integra con AWS Identity and Access Management (IAM).	2 de diciembre de 2010