Guía del usuario

AWS IoT Analytics

AWS IoT Analytics: Guía del usuario

Table of Contents

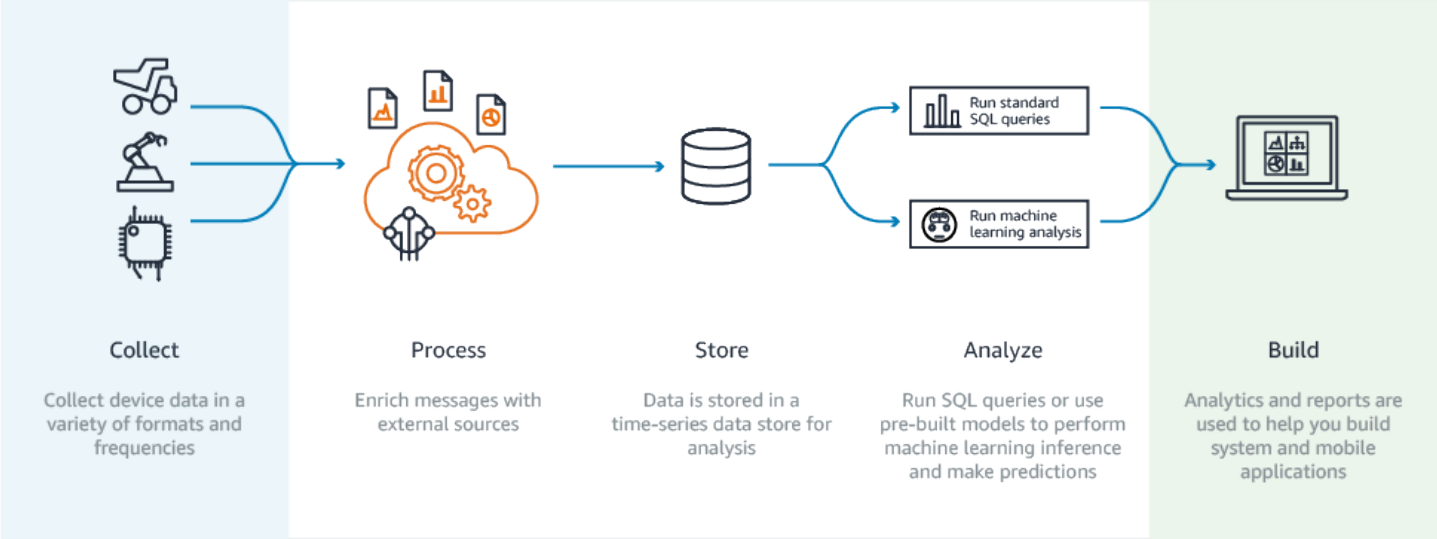
¿Qué es AWS IoT Analytics?

AWS IoT Analytics automatiza los pasos necesarios para analizar los datos de los dispositivos de IoT. AWS IoT Analytics filtra, transforma y enriquece los datos de IoT antes de almacenarlos en un almacén de datos de serie temporal para su análisis. Puede configurar el servicio para recopilar solo los datos que necesite de sus dispositivos, aplicar transformaciones matemáticas a los datos para procesarlos y enriquecerlos con metadatos específicos del dispositivo, como el tipo de dispositivo y la ubicación, antes de almacenarlos. A continuación, puede analizar los datos mediante la ejecución de consultas con el motor de consultas SQL integrado o realizar análisis más complejos e inferencias de machine learning. AWS IoT Analytics permite la exploración avanzada de datos mediante la integración con el cuaderno de Jupyter. AWS IoT Analytics también permite la visualización de datos mediante la integración con Amazon QuickSight. Amazon QuickSight está disponible en las siguientes regiones.

Las herramientas tradicionales de análisis e inteligencia empresarial están diseñadas para procesar datos estructurados. Los datos de IoT sin procesar suelen proceder de dispositivos que registran datos menos estructurados (como la temperatura, el movimiento o el sonido). Como consecuencia, los datos de estos dispositivos pueden tener con frecuencia discontinuidades notables, mensajes dañados y lecturas falsas que se deben limpiar antes de poder llevar a cabo un análisis. Además, los datos de IoT a menudo solo son significativos en el contexto de otros datos de fuentes externas. AWS IoT Analytics le permite abordar estos problemas y recopilar grandes cantidades de datos de dispositivos, procesar mensajes y almacenarlos. A continuación, puede consultar los datos y analizarlos. AWS IoT Analytics incluye modelos predefinidos para los casos de uso habituales del IoT, de modo que pueda responder a preguntas como qué dispositivos están a punto de fallar o qué clientes corren el riesgo de abandonar sus dispositivos ponibles.

Cómo utilizar las AWS IoT Analytics

En el siguiente gráfico se muestra información general de cómo se puede usar AWS IoT Analytics.



Características principales

Recopilación

- Integración con AWS IoT Core: AWS IoT Analytics está plenamente integrado con AWS IoT Core, de modo que puede procesar mensajes de los dispositivos conectados a medida que se reciben.
- Uso de una API por lotes para agregar datos de cualquier origen: AWS IoT Analytics puede recibir datos de cualquier origen a través de HTTP. Esto significa que cualquier dispositivo o servicio que esté conectado a Internet puede enviar datos a AWS IoT Analytics. Para obtener más información, consulte BatchputMessage en la Referencia de la API de AWS IoT Analytics.
- Recopile solo los datos que desee almacenar y analizar; puede usar la consola de AWS IoT Analytics para configurar a AWS IoT Analytics para que reciba mensajes de los dispositivos mediante filtros de temas de MQTT en varios formatos y frecuencias. AWS IoT Analytics valida que los datos estén dentro de los parámetros específicos que usted defina, y crea canales. A continuación, el servicio dirige los canales a las canalizaciones adecuadas para realizar el procesamiento, la transformación y el enriquecimiento de los mensajes.

Proceso

- Limpieza y filtrado: AWS IoT Analytics le permite definir funciones de AWS Lambda que se activan cuando AWS IoT Analytics detecta datos que faltan, de modo que se puede ejecutar código para estimar y cubrir las discontinuidades. También se pueden definir filtros de máximos y mínimos y umbrales de percentiles para eliminar los valores atípicos de los datos.

- Transformación: AWS IoT Analytics puede transformar mensajes utilizando la lógica matemática o condicional que se defina, de modo que puede realizar cálculos comunes tales como conversión de grados Celsius a Fahrenheit.
- Enriquecimiento: AWS IoT Analytics puede enriquecer los datos con orígenes de datos externos como, por ejemplo, información de previsión meteorológica y, a continuación, dirigir los datos al almacén de datos de AWS IoT Analytics.

Almacenar

- Almacén de datos de series temporales: AWS IoT Analytics almacena los datos de los dispositivos en un almacén de datos de series temporales optimizado para realizar un análisis y una recuperación más rápidos. También puede administrar permisos de acceso, implementar políticas de retención de datos y exportar los datos a puntos de acceso externos.
- Almacenar datos procesados y sin procesar: AWS IoT Analytics almacena los datos procesados y además almacena automáticamente los datos sin procesar adquiridos para que pueda procesarlos más adelante.

Análisis

- Ejecución de consultas SQL ad hoc: AWS IoT Analytics proporciona un motor de consultas SQL que permite ejecutar consultas ad hoc y obtener los resultados rápidamente. El servicio le permite utilizar consultas SQL estándar para extraer datos desde el almacén de datos para responder a preguntas como la distancia media que recorre una flota de vehículos conectados o cuántas puertas están bloqueadas después de las 19:00 h en un edificio inteligente. Estas consultas se pueden reutilizar incluso si cambian los dispositivos conectados, el tamaño de la flota y los requisitos de análisis.
- Análisis de series temporales: AWS IoT Analytics admite los análisis de series temporales que permiten analizar el rendimiento de los dispositivos a lo largo del tiempo y entender cómo y dónde se están utilizando, monitorizar continuamente los datos de los dispositivos para predecir problemas de mantenimiento y monitorizar sensores para predecir y reaccionar ante condiciones ambientales.
- Cuadernos alojados para análisis sofisticados y machine learning: AWS IoT Analytics incluye soporte para cuadernos alojados en el cuaderno de Jupyter para análisis estadísticos y machine learning. El servicio incluye un conjunto de plantillas de cuadernos que contienen modelos y visualizaciones de machine learning creados por AWS. Puede utilizar las plantillas para iniciarse en los casos de uso de IoT relacionados con los perfiles de fallo de los dispositivos, la previsión de eventos de bajo uso que podrían indicar que el cliente abandonará el producto, o la segmentación de los dispositivos por niveles de uso del cliente (p. ej., usuarios intensivos, usuarios de fin de semana) o estado del dispositivo. Después de crear un bloc de

notas, puede incluirlo en contenedores y ejecutarlo en un horario que especifique. Para obtener más información, consulte Automatización del flujo de trabajo.

- Predicción: puede realizar una clasificación estadística a través de un método denominado regresión logística. Además, puede utilizar la Memoria a largo-corto plazo (LSTM), que es una potente técnica de redes neuronales para predecir el resultado o estado de un proceso que varía a lo largo del tiempo. Las plantillas de bloc de notas prediseñadas admiten además el algoritmo de clústering de K-means para segmentación de dispositivos, que agrupa los dispositivos en grupos de dispositivos similares. Estas plantillas se utilizan normalmente para realizar perfiles de estado de dispositivos como, por ejemplo, equipos de sistemas de aire acondicionado en una fábrica de chocolate o el desgaste de las palas de una turbina eólica. De nuevo, estas plantillas de bloc de notas pueden incluirse en contenedores y ejecutarse según una programación.

Compilación y visualización

- Integración de QuickSight: AWS IoT Analytics proporciona un conector para Amazon QuickSight que permite visualizar los conjuntos de datos en un panel de QuickSight.
- Integración de la consola: además, puede visualizar los resultados o su análisis ad-hoc en los cuadernos de Jupyter integrados en la consola de AWS IoT Analytics.

Componentes y conceptos de AWS IoT Analytics

Canal

Un canal recopila datos desde un tema MQTT y archiva los mensajes sin procesar antes de publicar los datos en una canalización. También puede enviar mensajes a un canal directamente utilizando la API BatchPutMessage. Los mensajes sin procesar se almacenan en un bucket de Amazon Simple Storage Service (Amazon S3) que usted o AWS IoT Analytics administra.

Canalización

Una canalización consume mensajes de un canal y le permite procesar los mensajes antes de guardarlos en un almacén de datos. Las etapas de procesamiento, denominadas actividades (actividades de canalización), realizan transformaciones en los mensajes, tales como eliminar, cambiar el nombre o añadir atributos al mensaje, filtrar mensajes en función de los valores de los atributos, ejecutar funciones de Lambda con los mensajes para el procesamiento avanzado o realizar transformaciones matemáticas para normalizar los datos de los dispositivos.

Almacén de datos

Las canalizaciones almacenan sus mensajes procesados en un almacén de datos. Un almacén de datos no es una base de datos, sino un repositorio de mensajes que se puede escalar y consultar. Puede tener varios almacenes de datos para mensajes procedentes de distintos dispositivos o ubicaciones, o para mensajes filtrados mediante diferentes atributos, en función de la configuración de la canalización y de los requisitos. Al igual que los mensajes de canal sin procesar, los mensajes procesados de un almacén de datos se almacenan en un bucket de Amazon S3 administrado por usted o AWS IoT Analytics.

Conjunto de datos

Los datos de un almacén de datos se recuperan creando un conjunto de datos. AWS IoT Analytics le permite crear un conjunto de datos SQL o un conjunto de datos de contenedor.

Cuando disponga de un conjunto de datos, puede explorar y obtener información sobre ellos mediante la integración con Amazon QuickSight. También puede realizar funciones de análisis más avanzadas mediante la integración con el cuaderno de Jupyter. El cuaderno de Jupyter proporciona potentes herramientas de ciencia de datos que pueden llevar a cabo machine learning y una amplia gama de análisis estadísticos. Para obtener más información, consulte Plantillas de bloc de notas.

Puede enviar contenido de un conjunto de datos a un bucket de Amazon S3, lo que permite la integración con los lagos de datos existentes o el acceso desde aplicaciones internas y herramientas de visualización. También puede enviar contenido del conjunto de datos como una entrada a AWS IoT Events, un servicio que le permite monitorizar dispositivos o procesos para ver si se producen errores o cambios en la operación, y para activar acciones adicionales cuando se producen estos eventos.

Conjunto de datos SQL

Un conjunto de datos SQL es algo parecido a una vista materializada de una base de datos SQL. Puede crear un conjunto de datos SQL aplicando una acción SQL. Los conjuntos de datos SQL se pueden generar automáticamente de forma periódica especificando un desencadenador.

Conjunto de datos de contenedores

Un conjunto de datos de contenedores le permite ejecutar automáticamente herramientas de análisis y generar resultados. Para obtener más información, consulte Automatización del flujo de trabajo. Aúna un conjunto de datos SQL como entrada, un contenedor de Docker con las herramientas de análisis y los archivos de biblioteca necesarios, variables de entrada y salida

y un desencadenador de programación opcional. Las variables de entrada y salida indican a la imagen ejecutable dónde obtener los datos y almacenar los resultados. El desencadenador puede ejecutar el análisis cuando un conjunto de datos SQL termina de crear su contenido o de acuerdo con una expresión de programación de tiempo. Un conjunto de datos de contenedores se ejecutará automáticamente, y generará y, a continuación, guardará los resultados de las herramientas de análisis.

Desencadenador

Puede crear automáticamente un conjunto de datos especificando un desencadenador. El desencadenador puede ser un intervalo de tiempo (p. ej., crear este conjunto de datos cada dos horas) o el momento en que se crea el contenido de otro conjunto de datos (p. ej., crear este conjunto de datos cuando `myOtherDataset` termine de crear su contenido). O bien, puede generar el contenido del conjunto de datos manualmente mediante la API CreateDatasetContent.

Contenedor de Docker

Puede crear su propio contenedor de Docker para empaquetar las herramientas de análisis o utilizar las opciones que proporciona SageMaker. Para obtener más información, consulte Contenedor de Docker. Puede crear su propio contenedor de Docker para empaquetar las herramientas de análisis o utilizar las opciones proporcionadas por SageMaker. Puede almacenar un contenedor en un registro de Amazon ECR que especifique de forma que esté disponible para su instalación en la plataforma deseada. Los contenedores de Docker son capaces de ejecutar el código de análisis personalizado preparado con Matlab, Octave, Wise.io, SPSS, R, Fortran, Python, Scala, Java, C++, etc. Para obtener más información, consulte Inclusión de un bloc de notas en contenedores.

Ventanas diferenciales

Las ventanas diferenciales son una serie de intervalos de tiempo definidos por el usuario, no solapados y contiguos. Las ventanas diferenciales le permiten crear el contenido del conjunto de datos y realizar análisis sobre los nuevos datos que han llegado al almacén de datos desde el último análisis. Para crear una ventana diferencial, establezca el valor `deltaTime` en la parte `filters` de una `queryAction` de un conjunto de datos. Para obtener más información, consulte la API `CreateDataset`. Normalmente, querrá crear el contenido del conjunto de datos automáticamente estableciendo un desencadenador de intervalo de tiempo (`triggers:schedule:expression`). Esto le permite filtrar los mensajes que han llegado durante un periodo de tiempo específico, de forma que los datos contenidos en mensajes de periodos anteriores no se contabilicen dos veces. Para más información, consulte Ejemplo 6. Creación de un conjunto de datos SQL con una ventana diferencial (CLI).

Acceder a AWS IoT Analytics

Como parte de AWS IoT, AWS IoT Analytics proporciona las siguientes interfaces para permitir a sus dispositivos generar datos y a sus aplicaciones interactuar con los datos que generan:

AWS Command Line Interface (AWS CLI)

Ejecuta comandos para AWS IoT Analytics en Windows, OS X y Linux. Estos comandos le permiten crear y administrar objetos, certificados, reglas y políticas. Para empezar, consulte la AWS Command Line Interface Guía de usuario de . Para obtener más información sobre los comandos de AWS IoT, consulte iot en Referencia de AWS Command Line Interface.

> **Important**
>
> Utilice el comando `aws iotanalytics` para interactuar con AWS IoT Analytics. Utilice el comando `aws iot` para interactuar con otras partes del sistema de IoT.

API de AWS IoT

Cree sus aplicaciones IoT mediante solicitudes HTTP o HTTPS. Estas acciones de la API le permiten crear y administrar objetos, certificados, reglas y políticas. Para obtener más información, consulte Acciones de en la Referencia de la API de AWS IoT.

SDK de AWS

Cree sus aplicaciones AWS IoT Analytics mediante las API específicas de cada idioma. Estos SDK integran las API de HTTP y HTTPS y le permiten programar en cualquiera de los lenguajes admitidos. Para obtener más información, consulte SDK y herramientas de AWS.

SDK de dispositivos de AWS IoT

Cree aplicaciones que se ejecutan en sus dispositivos para enviar mensajes a AWS IoT Analytics. Para obtener más información, consulte SDK de AWS IoT.

Consola de AWS IoT Analytics

Puede crear los componentes para visualizar los resultados en la consola de AWS IoT Analytics.

Casos de uso

Mantenimiento predictivo

AWS IoT Analytics proporciona plantillas predefinidas para crear modelos de mantenimiento predictivo y aplicarlos a los dispositivos. Por ejemplo, es posible utilizar AWS IoT Analytics para predecir cuándo es probable que se produzcan averías en los sistemas de calefacción y ventilación en los vehículos de carga conectados, de modo que el vehículo se pueda redirigir para evitar daños en la carga. O bien, un fabricante de automóviles puede detectar qué clientes tienen gastadas las pastillas de freno y avisarles para que sus vehículos se sometan a revisión.

Reabastecimiento de suministros proactivo

AWS IoT Analytics le permite crear aplicaciones de IoT que pueden monitorear inventarios en tiempo real. Por ejemplo, una compañía de alimentación y bebidas puede analizar los datos de las máquinas expendedoras de forma proactiva y realizar un pedido de mercancía cuando baje el nivel de existencias.

Puntuación de la eficiencia de los procesos

Con AWS IoT Analytics, puede crear aplicaciones que monitoricen constantemente la eficiencia de distintos procesos y tomen medidas para mejorar el proceso. Por ejemplo, una empresa minera puede mejorar la eficiencia de sus camiones de mineral maximizando la carga en cada viaje. Con AWS IoT Analytics, la empresa puede identificar la carga más eficiente para una ubicación o camión a lo largo del tiempo, luego comparar las diferencias con la carga objetivo en tiempo real y planificar mejor las directrices de carga para mejorar la eficiencia.

Agricultura inteligente

AWS IoT Analytics puede enriquecer automáticamente datos de dispositivos de IoT con metadatos contextuales utilizando datos del registro de AWS IoT u orígenes de datos públicos, de modo que los análisis puedan tener en cuenta factores como la hora, la ubicación, la temperatura, la altitud y otras condiciones medioambientales. Con dicho análisis, es posible escribir modelos que generen acciones recomendadas que deben tomar los dispositivos en el campo. Por ejemplo, para determinar cuándo hay que regar, los sistemas de riego podrían enriquecer los datos del sensor de humedad con datos sobre precipitaciones, lo que permite un uso más eficiente del agua.

Primeros pasos con AWS IoT Analytics (consola)

Utilice este tutorial para crear los AWS IoT Analytics recursos (también conocidos como componentes) que necesita para descubrir información útil sobre los datos de sus dispositivos de IoT.

> **Notas**
>
> - Si introduce caracteres en mayúscula en el siguiente tutorial, los cambia AWS IoT Analytics automáticamente a minúsculas.
> - La AWS IoT Analytics consola tiene una función de introducción con un solo clic para crear un canal, una canalización, un almacén de datos y un conjunto de datos. Encontrará esta característica al iniciar sesión en la consola de AWS IoT Analytics .
>   - Este tutorial te guía paso a paso para crear tus AWS IoT Analytics recursos.

Sigue las instrucciones que aparecen a continuación para crear un AWS IoT Analytics canal, una canalización, un banco de datos y un conjunto de datos. El tutorial también muestra cómo usar la AWS IoT Core consola para enviar los mensajes que se van a AWS IoT Analytics ingerir.

Temas

Inicie sesión en la consola AWS IoT Analytics

Para empezar, debes tener una AWS cuenta. Si ya tiene una AWS cuenta, vaya a https://console.aws.amazon.com/iotanalytics/.

Si no tienes una AWS cuenta, sigue estos pasos para crear una.

Para crear una AWS cuenta

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.

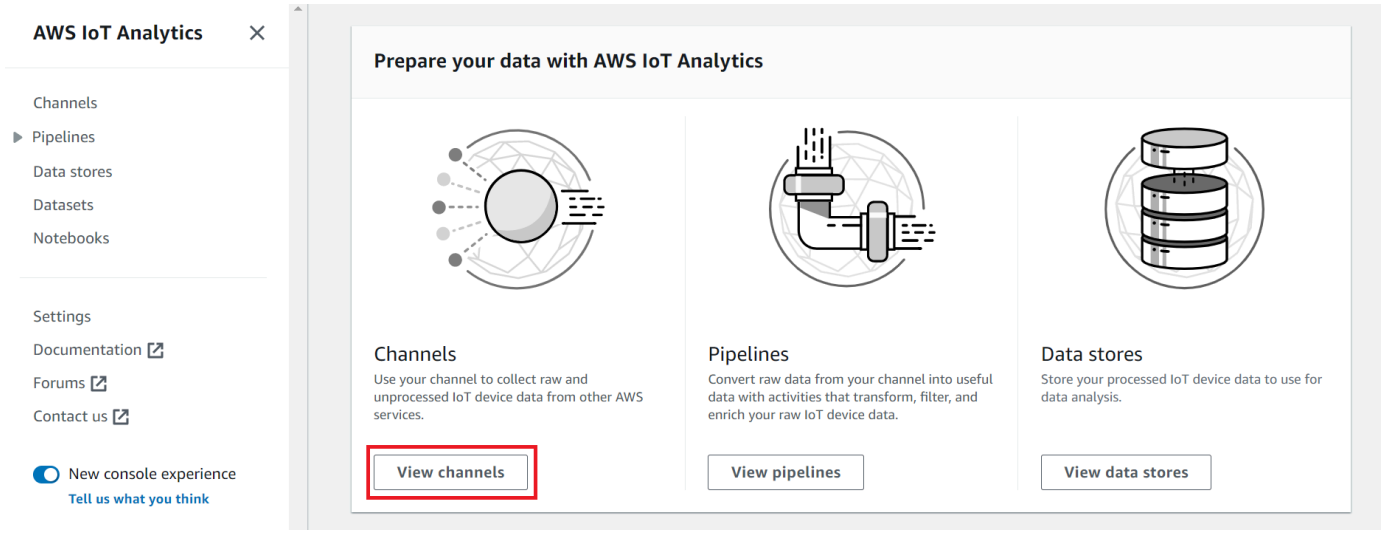
   Cuando te registras en una Cuenta de AWS, Usuario raíz de la cuenta de AWSse crea una. El usuario raíz tendrá acceso a todos los Servicios de AWS y recursos de esa cuenta. Como práctica recomendada de seguridad, asigne acceso administrativo a un usuario y utilice únicamente el usuario raíz para realizar tareas que requieren acceso de usuario raíz.
3. Inicia sesión en AWS Management Console y navega hasta https://console.aws.amazon.com/iotanalytics/.

Crear un canal

Un canal recopila y archiva datos de dispositivos de IoT sin procesar y no estructurados. Siga estos pasos para crear un canal.

Para crear un canal

1. En https://console.aws.amazon.com/iotanalytics/, en la sección Preparación de sus datos con AWS IoT Analytics, seleccione Ver canales.



> **Tip**
>
> También puede seleccionar Canales en el panel de navegación.

2. En la página Channels (Canales), seleccione Create channel (Crear canal).
3. En la página de Especificación de los detalles del canal, introduzca los detalles de su canal.
   a. Introduzca un nombre de canal que sea único y que pueda identificar fácilmente.
   b. (Opcional) En Etiquetas, agregue una o más etiquetas personalizadas (pares clave-valor) al canal. Las etiquetas pueden ayudarlo a identificar los recursos que usted crea en AWS IoT Analytics.
   c. Elija Siguiente.
4. AWS IoT Analytics almacena los datos sin procesar y sin procesar de sus dispositivos de IoT en un bucket de Amazon Simple Storage Service (Amazon S3). Puede elegir su propio bucket de Amazon S3, al que puede acceder y gestionar, o AWS IoT Analytics puede gestionar el bucket de Amazon S3 por usted.
   a. En este tutorial, en Storage type, seleccione Almacenamiento administrado por servicio.
   b. En Elegir durante cuánto tiempo desea almacenar los datos sin procesar, seleccione Indefinidamente.
   c. Elija Siguiente.
5. En la página Configurar la fuente, introduzca la información de la AWS IoT Analytics que desee recopilar los datos de los mensajes AWS IoT Core.

a. Introduzca un filtro de AWS IoT Core tema, por ejemplo,`update/environment/dht1`. Más adelante en este tutorial, utilizará este filtro de tema para enviar datos de mensajes a su canal.

b. En el área rol de IAM, seleccione Crear nuevo. En la ventana Crear un nuevo rol, escriba un nombre para el rol y, a continuación, seleccione Crear rol. Esto crea automáticamente un rol con una política apropiada adjunta.

c. Elija Siguiente.

6. Revise las opciones seleccionadas y, a continuación, elija Crear canal.
7. Verifique que su nuevo canal aparezca en la página Canales.

Crear un almacén de datos

Un almacén de datos recibe y almacena los datos de los mensajes. Un almacén de datos no es una base de datos. En cambio, un almacén de datos es un repositorio escalable y consultable en un bucket de Amazon S3. Puede usar varios almacenes de datos para los mensajes de diferentes dispositivos o ubicaciones. O bien, puede filtrar los datos de los mensajes en función de la configuración y los requisitos de la canalización.

Siga estos pasos para crear un almacén de datos.

Para crear un almacén de datos

1. En https://console.aws.amazon.com/iotanalytics/, en la sección Preparación de sus datos con AWS IoT Analytics, seleccione Ver almacenes de datos.
2. En la página Almacenamiento de datos, seleccione Crear un almacén de datos.
3. En la página Especificar detalles del almacén de datos, introduzca la información básica sobre el almacén de datos.

   a. En ID del almacén de datos, introduzca un ID de almacén de datos único. No se puede cambiar este ID después de crearlo.

   b. (Opcional) En el caso de Etiquetas, seleccione Agregar nueva etiqueta para agregar una o más etiquetas personalizadas (pares clave-valor) al almacén de datos. Las etiquetas pueden ayudarlo a identificar los recursos que usted crea en AWS IoT Analytics.

   c. Elija Siguiente.

4. En la página Configuración del tipo de almacenamiento, especifique de qué forma se almacenarán los datos.

   a. En Tipo de almacenamiento, seleccione Almacenamiento administrado por servicio.

   b. En Configurar durante cuánto tiempo desea conservar los datos procesados, seleccione Indefinidamente.

   c. Elija Siguiente.

5. AWS IoT Analytics los almacenes de datos admiten los formatos de archivo JSON y Parquet. Para el formato de datos de un almacén de datos, seleccione JSON o Parquet. Para más información sobre los tipos de AWS IoT Analytics compatibles, consulte Formatos de archivo.

   Elija Siguiente.

6. (Opcional) AWS IoT Analytics admite particiones personalizadas en su almacén de datos para que pueda consultar datos eliminados y mejorar la latencia. Para más información sobre las particiones personalizadas compatibles, consulte Particiones personalizadas.

   Elija Siguiente.

7. Revise las opciones seleccionadas y, a continuación, elija Crear un almacén de datos.

8. Compruebe que el nuevo almacén de datos aparezca en la página de Almacenes de datos.

Creación de una canalización

Debe crear una canalización para conectar un canal a un almacén de datos. Una canalización básica solo especifica el canal que recopila los datos e identifica el almacén de datos al que se envían los mensajes. Para obtener más información, consulte Canalización de actividades.

Para este tutorial, debe crear una canalización que solo conecte un canal a un almacén de datos. A continuación, puede agregar actividades de canalización para procesar estos datos.

Siga estos pasos para crear una canalización.

Para crear una canalización

1. En https://console.aws.amazon.com/iotanalytics/, en la sección Preparación de sus datos con AWS IoT Analytics, seleccione Ver canalizaciones.

> **Tip**
>
> También puede elegir Canalizaciones en el panel de navegación.

2. En la página Canalizaciones, seleccione Crear canalización.
3. Introduzca los detalles de la canalización.
   a. En Configurar el ID y las fuentes de la canalización, introduzca el nombre de la canalización.
   b. Elige la fuente de tu canalización, que es el AWS IoT Analytics canal desde el que tu canalización leerá los mensajes.
   c. Especifique la salida de su canalización, que es el almacén de datos en el que se almacenan los datos de los mensajes procesados.
   d. (Opcional) En Etiquetas, agregue una o más etiquetas personalizadas (pares clave-valor) a la canalización.
   e. En la página Inferir los atributos de los mensajes, escriba un nombre de atributo y un valor de ejemplo, seleccione un tipo de datos de la lista y, a continuación, seleccione Añadir atributo.
   f. Repita el paso anterior para todos los atributos que necesite y, a continuación, seleccione Siguiente.
   g. En este momento no añadirá ninguna actividad de canalización. En la página Enriquecer, transformar y filtrar mensajes, seleccione Siguiente.
4. Revise las opciones seleccionadas y, a continuación, elija Crear canalización.
5. Compruebe que la nueva canalización aparezca en la página Canalizaciones.

> **Note**
>
> Has creado AWS IoT Analytics recursos para que puedan hacer lo siguiente:
>
> - Recopilar datos de mensajes de dispositivos IoT sin procesar con un canal.
> - Guardar los datos de mensajes de su dispositivo IoT en un almacén de datos.
> - Limpiar, filtrar, transformar y enriquecer sus datos con una canalización.

A continuación, creará un conjunto de datos AWS IoT Analytics SQL para descubrir información útil sobre su dispositivo de IoT.

Creación de un conjunto de datos

> **Note**
>
> Un conjunto de datos suele ser un conjunto de datos que puede o no estar organizado en forma tabular. Por el contrario, AWS IoT Analytics crea su conjunto de datos aplicando una consulta SQL a los datos de su banco de datos.

Ahora dispone de un canal que dirige los datos de mensajes sin procesar a una canalización que los almacena en un almacén de datos donde se pueden consultar. Para consultar los datos, se crea un conjunto de datos. Los conjuntos de datos contienen sentencias y expresiones SQL que se utilizan para consultar el almacén de datos junto con una programación opcional que repite la consulta en el día y la hora que se especifique. Puede utilizar expresiones similares a las expresiones de CloudWatch programación de Amazon para crear las programaciones opcionales.

Para crear un conjunto de datos

1. En https://console.aws.amazon.com/iotanalytics/, en el panel de navegación izquierdo, seleccione Conjuntos de datos.
2. En la página Crear conjunto de datos, seleccione Crear SQL.
3. En la página Especificar los detalles del conjunto de datos, especifique los detalles del conjunto de datos.
   a. Escriba un nombre para el conjunto de datos.
   b. Para Fuente del almacén de datos, seleccione el ID único que identifica al almacén de datos que creó anteriormente.
   c. (Opcional) En Etiquetas, agregue una o más etiquetas personalizadas (pares clave-valor) al conjunto de datos.
4. Utilice expresiones SQL para consultar los datos y responder a las preguntas analíticas. Los resultados de la consulta se almacenan en este conjunto de datos.

a. En el campo Consulta del autor, introduzca una consulta SQL que utilice un comodín para mostrar hasta cinco filas de datos.

```
SELECT * FROM my_data_store LIMIT 5
```

Para obtener más información sobre las funciones de SQL compatibles en AWS IoT Analytics, consulte[Expresiones SQL en AWS IoT Analytics](#).

b. Puede elegir Probar consulta para validar que la entrada es correcta y mostrar los resultados en una tabla después de la consulta.

> **Note**
>
> - En este punto del tutorial, es posible que su almacén de datos esté vacío. Al ejecutar una consulta SQL en un almacén de datos vacío, no se obtendrán resultados, por lo que es posible que solo vea `__dt`.
> - Debe tener cuidado de limitar la consulta SQL a un tamaño razonable para que no se ejecute durante un período prolongado, ya que Athena limita el número máximo de consultas en ejecución. Por ello, debe tener cuidado de limitar la consulta SQL a un tamaño razonable.
>
> Le sugerimos que utilice una cláusula `LIMIT` en la consulta durante las pruebas. Una vez que la prueba se haya realizado correctamente, se puede eliminar esta cláusula.

5. (Opcional) Al crear el contenido de un conjunto de datos con datos de un período de tiempo específico, es posible que algunos datos no lleguen a tiempo para su procesamiento. Para permitir un retraso, puede especificar un desplazamiento o tiempo delta. Para obtener más información, consulte Obtención de notificaciones de datos atrasados a través de Eventos de Amazon CloudWatch.

   En este momento no configurará un filtro de selección de datos. En la página Configurar el filtro de selección de datos, seleccione Siguiente.

6. (Opcional) Puede programar esta consulta para que se ejecute con regularidad a fin de actualizar el conjunto de datos. Las programaciones de los conjuntos de datos se pueden crear y editar en cualquier momento.

No va a programar una ejecución recurrente de la consulta en este momento; por tanto, en la página Definir programación de la consulta, elija Siguiente.

7. AWS IoT Analytics creará versiones del contenido de este conjunto de datos y almacenará los resultados de sus análisis durante el período especificado. La recomendación es 90 días, pero puede optar por establecer su propia política de retención personalizada. También puede limitar el número de versiones almacenadas del contenido de su conjunto de datos.

   Puede usar el período de retención del conjunto de datos predeterminado como Indefinidamente y mantener el Control de versiones desactivado. En la página Configurar los resultados de sus análisis, seleccione Siguiente.

8. (Opcional) Puede configurar las reglas de entrega de los resultados de su conjunto de datos a un destino específico, por ejemplo AWS IoT Events.

   No enviará los resultados a ningún otro lugar de este tutorial, por lo que en la página Configurar las reglas de entrega del contenido del conjunto de datos, seleccione Siguiente.

9. Revise las opciones seleccionadas y, a continuación, elija Crear conjunto de datos.
10. Compruebe que su nuevo conjunto de datos aparezca en la página Conjuntos de datos.

Envíe los datos del mensaje con AWS IoT

Si tiene un canal que redirige los datos a una canalización, que almacena datos en un almacén de datos donde se pueden consultar, entonces está preparado para enviar datos de dispositivos IoT a AWS IoT Analytics. Puede enviar datos a AWS IoT Analytics mediante las siguientes opciones:

- Utilice el intermediario de AWS IoT mensajes.
- Use la operación AWS IoT Analytics BatchPutMessage de la API.

En los siguientes pasos, envía los datos de los mensajes desde el agente de AWS IoT mensajes de la AWS IoT Core consola para que AWS IoT Analytics pueda ingerir estos datos.

> **Note**
>
> Cuando cree nombres de tema para sus mensajes, tenga en cuenta lo siguiente:
>
> - Los nombres de los temas no distinguen entre mayúsculas y minúsculas. Los campos denominados `example` y `EXAMPLE` en la misma carga se considerarán duplicados.

- Los nombres de tema no pueden empezar con el carácter $. Los temas que comienzan por $ son temas reservados y solo pueden ser utilizados por AWS IoT.
- No incluya información de identificación personal en los nombres de tema, ya que esta información puede aparecer en comunicaciones e informes no cifrados.
- AWS IoT Core no puede enviar mensajes entre AWS cuentas o AWS regiones.

Para enviar datos de mensajes con AWS IoT

1. Inicie sesión en la consola de AWS IoT.
2. En el panel de navegación, seleccione Probar y, a continuación, seleccione el Cliente de prueba MQTT.
3. En el Cliente MQTT, seleccione Publicar en un tema.
4. En Nombre de tema, escriba un nombre que coincida con el filtro de tema que introdujo al crear un canal. En este ejemplo se utiliza `update/environment/dht1`.
5. En Carga de mensajes, introduzca el siguiente contenido JSON.

```
{
  "thingid": "dht1",
  "temperature": 26,
  "humidity": 29,
  "datetime": "2018-01-26T07:06:01"
}
```

6. (Opcional) Seleccione Añadir configuración para ver más opciones de protocolo de mensajes.
7. Seleccione Publish (Publicar).

   De este modo, se publica un mensaje que es capturado por el canal. A continuación, la canalización dirige el mensaje al almacén de datos.

Comprueba el progreso de los AWS IoT mensajes

Puede comprobar que los mensajes se están insertando en su canal siguiendo estos pasos.

Para comprobar el progreso de los AWS IoT mensajes

1. Inicie sesión en https://console.aws.amazon.com/iotanalytics/.

2. En el panel de navegación, seleccione Canales y, a continuación, seleccione el nombre del canal que creó anteriormente.
3. En la página Detalles del canal, desplácese hacia abajo hasta la sección Monitorización y, a continuación, ajuste el periodo de tiempo que se muestra (1h 3h 12h 1d 3d 1w). Seleccione un valor, como, por ejemplo, 1w para ver los datos de la última semana.

Puede usar una característica similar para monitorear la actividad, el tiempo de ejecución y los errores de la canalización en la página de Detalles de la canalización. En este tutorial, no ha especificado actividades como parte de la canalización, por lo que no debería ver ningún error de tiempo de ejecución.

Para monitorear la actividad de la canalización

1. En el panel de navegación, seleccione Canalizaciones y, a continuación, seleccione el nombre de la canalización que creó anteriormente.
2. En la página Detalles de la canalización, desplácese hacia abajo hasta la sección Monitorización y, a continuación, ajuste el periodo de tiempo que se muestra seleccionando uno de los indicadores del periodo de tiempo (1h 3h 12h 1d 3d 1w).

Acceder a los resultados de la consulta

El contenido del conjunto de datos es un archivo que contiene el resultado de su consulta, en formato CSV.

1. En https://console.aws.amazon.com/iotanalytics/, en el panel de navegación izquierdo, seleccione Conjuntos de datos.
2. En la página Conjuntos de datos, seleccione el nombre del conjunto de datos que creó anteriormente.
3. En la página de información del conjunto de datos, en la esquina superior derecha, seleccione Ejecutar ahora.
4. Para comprobar si el conjunto de datos está listo, busque debajo del conjunto de datos un mensaje similar a este: Ha iniciado correctamente la consulta para su conjunto de datos. En la pestaña Contenido del conjunto de datos figuran los resultados de la consulta y se muestra el mensaje Se ha realizado con éxito.

5. Para obtener una vista previa de los resultados de la consulta realizada con éxito, en la pestaña Contenido del conjunto de datos, seleccione el nombre de la consulta. Para ver o guardar el archivo CSV que contiene los resultados de la consulta, seleccione Descargar.

> **Note**
>
> AWS IoT Analytics puede incrustar la parte HTML de un cuaderno de Jupyter en la página de contenido del conjunto de datos. Para obtener más información, consulte Visualización de datos de AWS IoT Analytics con la consola.

Explore sus datos

Dispone de varias opciones para almacenar, analizar y visualizar sus datos.

Amazon Simple Storage Service

Puede enviar contenido de un conjunto de datos a un bucket de Amazon S3, lo que permite la integración con los lagos de datos existentes o el acceso desde aplicaciones internas y herramientas de visualización. Consulte el campo de `contentDeliveryRules::destination::s3DestinationConfiguration` la CreateDatasetoperación.

AWS IoT Events

Puede enviar el contenido del conjunto de datos como entrada a AWS IoT Events un servicio que le permita monitorear los dispositivos o procesos para detectar fallas o cambios en el funcionamiento e iniciar acciones adicionales cuando se produzcan dichos eventos.

Para ello, cree un conjunto de datos mediante la CreateDatasetoperación y especifique una AWS IoT Events entrada en el campo`contentDeliveryRules :: destination :: iotEventsDestinationConfiguration :: inputName`. También debes especificar el `roleArn` rol, que otorga AWS IoT Analytics permisos de ejecución`iotevents:BatchPutMessage`. Siempre que se cree el contenido del conjunto de datos, AWS IoT Analytics enviará cada entrada del contenido del conjunto de datos como un mensaje a la AWS IoT Events entrada especificada. Por ejemplo, si su conjunto de datos incluye el siguiente contenido:

```
"what","who","dt"
"overflow","sensor01","2019-09-16 09:04:00.000"
```

```
"overflow","sensor02","2019-09-16 09:07:00.000"
"underflow","sensor01","2019-09-16 11:09:00.000"
...
```

A continuación, AWS IoT Analytics envía mensajes que contienen campos como los siguientes.

```
{ "what": "overflow", "who": "sensor01", "dt": "2019-09-16 09:04:00.000" }
```

```
{ "what": "overflow", "who": "sensor02", "dt": "2019-09-16 09:07:00.000" }
```

Querrá crear una AWS IoT Events entrada que reconozca los campos que le interesan (uno o más de ellos `what`,`who`,`dt`) y crear un modelo de AWS IoT Events detector que utilice estos campos de entrada en los eventos para activar acciones o establecer variables internas.

Cuaderno de Jupyter

El cuaderno de Jupyter es una solución de código abierto que permite utilizar lenguajes de programación para realizar exploraciones de datos ad hoc y análisis avanzados. Puede profundizar y aplicar análisis más complejos y utilizar métodos de machine learning, como la agrupación en clústeres k-means y los modelos de regresión para la predicción, en los datos de sus dispositivos de IoT.

AWS IoT Analytics utiliza instancias de Amazon SageMaker Notebooks para alojar sus Jupyter Notebooks. Antes de crear una instancia de bloc de notas, debes crear una relación entre Amazon AWS IoT Analytics y Amazon SageMaker:

1. Ve a la SageMaker consola y crea una instancia de bloc de notas:
   a. Rellene los detalles y, a continuación, elija Create a new role (Crear un nuevo rol). Anote el ARN del rol.
   b. Cree una instancia del bloc de notas.
2. Ve a la consola de IAM y modifica el SageMaker rol:
   a. Abra el rol. Debe tener una política administrada.
   b. Seleccione Añadir política insertada y, a continuación, en Servicio, seleccione iotAnalytics. Seleccione Seleccionar acciones y, a continuación, escriba **`GetDatasetContent`** en el cuadro de búsqueda y elíjala. Seleccione Review Policy (Revisar la política).

c. Revise la exactitud de la política, escriba un nombre y, a continuación, elija Crear política.

Esto le da permiso al rol recién creado para leer un conjunto de AWS IoT Analytics datos.

1. Vuelva a https://console.aws.amazon.com/iotanalytics/ y, en el panel de navegación de la izquierda, seleccione Cuadernos. En la página Blocs de notas, seleccione Crear bloc de notas.
2. En la página Seleccionar una plantilla, seleccione Plantilla en blanco de IoTA.
3. En la página Configurar bloc de notas, escriba un nombre para el bloc de notas. En Seleccionar conjunto de datos de origen, seleccione Seleccionar y, a continuación, seleccione el conjunto de datos que creó anteriormente. En Seleccione una instancia de bloc de notas, elija la instancia de bloc de notas en la que creó SageMaker.
4. Después de revisar sus opciones, seleccione Crear bloc de notas.
5. En la página de blocs de notas, la instancia de tu bloc de notas se abrirá en la SageMaker consola de Amazon.

Plantillas de bloc de notas

Las plantillas de AWS IoT Analytics bloc de notas contienen visualizaciones y modelos de aprendizaje automático AWS creados para ayudarte a empezar con AWS IoT Analytics los casos de uso. Puede utilizar estas plantillas de cuadernos para obtener más información, o bien, reutilizarlas para adaptarlas a los datos de sus dispositivos de IoT y ofrecer un valor inmediato.

Puede encontrar las siguientes plantillas de bloc de notas en la AWS IoT Analytics consola:

- Detección de anomalías contextuales: aplicación de la detección de anomalías contextuales en la velocidad del viento medida con un modelo de media móvil exponencialmente ponderada (PEWMA) de Poisson.
- Previsión de producción de panel solar: aplicación de modelos de series temporales lineales, estacionales y por partes para predecir la producción de paneles solares.
- Mantenimiento predictivo en motores a reacción: aplicación de redes neuronales de memoria multivariante de corto y largo plazo (LSTM) y regresión logística para predecir fallas en los motores a reacción.

- Segmentación de clientes de hogares inteligentes: aplicación de k-means y análisis de componentes principales (PCA) para detectar diferentes segmentos de clientes en datos de uso de hogares inteligentes.
- Previsión de atascos en ciudades inteligentes: aplicación de LSTM para predecir las tasas de utilización de carreteras urbanas.
- Previsión de la calidad del aire en ciudades inteligentes: aplicación de LSTM para predecir la contaminación por partículas en el centro de las ciudades.

Introducción a AWS IoT Analytics

Esta sección le muestra los comandos básicos que se utilizan para recopilar, almacenar, procesar y consultar sus datos de dispositivo mediante AWS IoT Analytics. Los ejemplos que se muestran aquí utilizan la AWS Command Line Interface (AWS CLI). Para obtener más información sobre AWS CLI, consulte la Guía del usuario de AWS Command Line Interface. Para obtener más información sobre los comandos de la CLI disponibles para AWS IoT, consulte iot en Referencia de AWS Command Line Interface.

> **Important**
>
> Utilice el comando `aws iotanalytics` para interactuar con AWS IoT Analytics utilizando la AWS CLI. Utilice el comando `aws iot` para interactuar con otras partes del sistema de IoT utilizando la AWS CLI.

> **Note**
>
> A medida que introduzca los nombres de las entidades de AWS IoT Analytics (canal, conjunto de datos, almacén de datos y canalización) en los ejemplos siguientes, tenga en cuenta que el sistema cambiará a minúsculas cualquier letra en mayúsculas que utilice. Los nombres de las entidades deben comenzar por una letra minúscula y solo pueden contener letras minúsculas, guiones bajos y dígitos.

Creación de un canal

Un canal recopila y archiva datos de mensajes sin procesar antes de publicar los datos en una canalización. Los mensajes entrantes se envían a un canal, por lo que el primer paso consiste en crear un canal para los datos.

```
aws iotanalytics create-channel --channel-name mychannel
```

Si desea que los mensajes de AWS IoT se inserten en AWS IoT Analytics, puede crear una regla de motor de reglas de AWS IoT para enviar los mensajes a este canal. Esto se muestra más adelante en Incorporación de datos en AWS IoT Analytics. Otra forma de obtener los datos en un canal es utilizar el comando `BatchPutMessage` de AWS IoT Analytics.

Para obtener una lista de los canales que ya ha creado:

```
aws iotanalytics list-channels
```

Para obtener más información sobre un canal.

```
aws iotanalytics describe-channel --channel-name mychannel
```

Los mensajes de canal sin procesar se almacenan en un bucket de Amazon S3 administrado por AWS IoT Analytics o en uno administrado por usted. Use el parámetro `channelStorage` para especificar cuál. El valor predeterminado es un bucket de Amazon S3 administrado por los servicios. Si desea que los mensajes de canal se almacenen en un bucket de Amazon S3 que administre, debe conceder permiso a AWS IoT Analytics para realizar estas acciones en su bucket de Amazon S3 en su nombre: `s3:GetBucketLocation` (verificar la ubicación del bucket), `s3:PutObject` (almacenar), `s3:GetObject` (leer) y `s3:ListBucket` (volver a procesar).

Example

```
{
    "Version": "2012-10-17",
    "Id": "MyPolicyID",
    "Statement": [
        {
            "Sid": "MyStatementSid",
            "Effect": "Allow",
            "Principal": {
                "Service": "iotanalytics.amazonaws.com"
            },
            "Action": [
                "s3:GetObject",
                "s3:GetBucketLocation",
                "s3:ListBucket",
                "s3:PutObject"
            ],
            "Resource": [
                "arn:aws:s3:::my-iot-analytics-bucket",
                "arn:aws:s3:::my-iot-analytics-bucket/*"
            ]
        }
    ]
}
```

Si realiza cambios en las opciones o permisos del almacenamiento de canal administrado por el cliente, es posible que tenga que volver a procesar los datos de canal para garantizar que los datos introducidos previamente están incluidos en el contenido del conjunto de datos. Consulte Reprocesamiento de los datos de canal.

Creación de un almacén de datos

Un almacén de datos recibe y almacena los mensajes. No se trata de una base de datos, sino de un repositorio escalable y consultable de sus mensajes. Puede crear varios almacenes de datos para almacenar los mensajes que provengan de diferentes dispositivos o ubicaciones, o bien, puede usar un solo almacén de datos para recibir todos sus mensajes de AWS IoT.

```
aws iotanalytics create-datastore --datastore-name mydatastore
```

Para obtener una lista de los almacenes de datos que ha creado.

```
aws iotanalytics list-datastores
```

Para obtener más información sobre un almacén de datos.

```
aws iotanalytics describe-datastore --datastore-name mydatastore
```

Políticas de Amazon S3 para recursos de AWS IoT Analytics

Puede almacenar los mensajes procesados del almacén de datos en un bucket de Amazon S3 administrado por AWS IoT Analytics o en uno que usted administre. Al crear un almacén de datos, seleccione el bucket de Amazon S3 que desee mediante el parámetro `datastoreStorage` de la API. El valor predeterminado es un bucket de Amazon S3 administrado por los servicios.

Si desea que los mensajes de almacenamiento de canal se almacenen en un bucket de Amazon S3 que administre, debe conceder permiso a AWS IoT Analytics para realizar estas acciones en su bucket de Amazon S3 en su nombre:

- `s3:GetBucketLocation`
- `s3:PutObject`
- `s3:DeleteObject`

Si utiliza el almacén de datos como origen para un conjunto de datos de consulta SQL, configure una política de bucket de Amazon S3 que conceda permiso a AWS IoT Analytics para invocar consultas de Amazon Athena sobre el contenido de su bucket.

> **Note**
>
> Le recomendamos que especifique `aws:SourceArn` en su política de bucket para ayudar a prevenir el problema de seguridad del suplente confuso. Esto restringe el acceso al permitir solo las solicitudes que provienen de una cuenta específica. Para obtener más información sobre el problema del suplente confuso, consulte the section called “Prevención del suplente confuso entre servicios”.

A continuación, se muestra una política de bucket de ejemplo que concede estos permisos necesarios.

```
{
    "Version": "2012-10-17",
    "Id": "MyPolicyID",
    "Statement": [
        {
            "Sid": "MyStatementSid",
            "Effect": "Allow",
            "Principal": {
                "Service": "iotanalytics.amazonaws.com"
            },
            "Action": [
                "s3:GetBucketLocation",
                "s3:GetObject",
                "s3:ListBucket",
                "s3:ListBucketMultipartUploads",
                "s3:ListMultipartUploadParts",
                "s3:AbortMultipartUpload",
                "s3:PutObject",
                "s3:DeleteObject"
            ],
            "Resource": [
                "arn:aws:s3:::DOC-EXAMPLE-BUCKET",
                "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"
            ],
            "Condition": {
```

```
                "ArnLike": {
                    "aws:SourceArn": [
                        "arn:aws:iotanalytics:us-east-1:123456789012:dataset/DOC-
EXAMPLE-DATASET",
                        "arn:aws:iotanalytics:us-east-1:123456789012:datastore/DOC-
EXAMPLE-DATASTORE"
                    ]
                }
            }
        }
    ]
}
```

Para obtener más información, consulte Acceso entre cuentas en la Guía del usuario de Amazon Athena.

> **Note**
>
> Si realiza cambios en la configuración o en los permisos del almacén de datos gestionado por el cliente, es posible que tenga que volver a procesar los datos del canal para asegurarse de que los datos introducidos previamente se incluyen en el contenido del conjunto de datos. Para obtener más información, consulte Reprocesamiento de los datos del canal.

Formatos de archivo

Los almacenes de datos de AWS IoT Analytics ahora admiten los formatos de archivo JSON y Parquet. El formato de archivo predeterminado es JSON.

- JSON (JavaScript Object Notation): es un formato de texto que admite pares nombre-valor y listas ordenadas de valores.
- Apache Parquet: es un formato de almacenamiento en columnas que se utiliza para almacenar y consultar grandes volúmenes de datos de manera eficiente.

Para configurar el formato de archivo del almacén de datos de AWS IoT Analytics, puede usar el objeto `FileFormatConfiguration` al crear el almacén de datos.

`fileFormatConfiguration`

Contiene la información de configuración de los formatos de archivo. AWS IoT Analytics los almacenes de datos admiten JSON y Parquet.

El formato de archivo predeterminado es JSON. Puede especificar solo un formato. No se puede cambiar el formato de archivo después de crear el almacén de datos.

`jsonConfiguration`

Contiene la información de configuración del formato JSON.

`parquetConfiguration`

Contiene la información de configuración del formato Parquet.

`schemaDefinition`

Información necesaria para definir un esquema.

`columns`

Especifica una o varias columnas que almacenan los datos.

Cada esquema puede tener hasta 100 columnas. Cada columna puede tener hasta 100 tipos anidados.

`name`

El nombre de la columna.

Restricciones de longitud: de 1 a 255 caracteres.

`type`

El tipo de datos. Para obtener más información sobre el tipo de datos compatible, consulte Tipos de datos comunes en la Guía para desarrolladores de AWS Glue.

Restricciones de longitud: de 1 a 131072 caracteres.

AWS IoT Analytics admite todos los tipos de datos que aparecen en la página Tipos de datos en Amazon Athena, excepto `DECIMAL(`*`precision, scale`*`) - `*`precision`*.

Crear un almacén de datos (consola)

En el siguiente procedimiento se explica cómo crear un almacén de datos que guarde los datos en formato Parquet.

Para crear un almacén de datos

1. Inicie sesión en https://console.aws.amazon.com/iotanalytics/.
2. En el panel de navegación, seleccione Almacenamiento de datos.
3. En la página Almacenamiento de datos, seleccione Crear un almacén de datos.
4. En la página Especificar detalles del almacén de datos, introduzca la información básica sobre el almacén de datos.

   a. En ID del almacén de datos, introduzca un ID de almacén de datos único. No se puede cambiar este ID después de crearlo.

   b. (Opcional) En el caso de Etiquetas, seleccione Agregar nueva etiqueta para agregar una o más etiquetas personalizadas (pares clave-valor) al almacén de datos. Las etiquetas pueden ayudarlo a identificar los recursos que usted crea en AWS IoT Analytics.

   c. Seleccione Next (Siguiente).
5. En la página Configuración del tipo de almacenamiento, especifique de qué forma se almacenarán los datos.

   a. En Tipo de almacenamiento, seleccione Almacenamiento administrado por servicio.

   b. En Configurar durante cuánto tiempo desea conservar los datos procesados, seleccione Indefinidamente.

   c. Seleccione Next (Siguiente).
6. En la página Configurar formato de datos, defina la estructura y el formato de los registros de datos.

   a. En Clasificación, seleccione Parquet. No se puede cambiar este formato después de crear el almacén de datos.

   b. En Fuente de inferencia, seleccione cadena JSON para el almacén de datos.

   c. En Cadena, introduzca el esquema en formato JSON, como se muestra en el siguiente ejemplo.

```
{
    "device_id": "0001",
    "temperature": 26,
    "humidity": 29,
    "datetime": "2018-01-26T07:06:01"
}
```

d. Seleccione Inferir esquema.

e. En Configurar esquema de Parquet, confirme que el formato coincide con su ejemplo de JSON. Si el formato no coincide, actualice el esquema de Parquet manualmente.

- Si desea que su esquema muestre más columnas, seleccione Añadir nueva columna, introduzca un nombre de columna y, a continuación, seleccione el tipo de datos.

> **Note**
>
> De forma predeterminada, puede tener 100 columnas en su esquema. Para obtener más información, consulte AWS IoT Analytics Cupos.

- Puede cambiar el tipo de datos de una columna existente. Para obtener más información sobre los tipos de datos compatibles, consulte Tipos de datos comunes en la Guía para desarrolladores de AWS Glue.

> **Note**
>
> Una vez creado el almacén de datos, no podrá cambiar el tipo de datos de una columna existente.

- Para eliminar una columna existente, seleccione Eliminar columna.

f. Seleccione Next (Siguiente).

7. (Opcional) AWS IoT Analytics admite particiones personalizadas en un almacén de datos para que pueda consultar datos depurados y mejorar la latencia. Para más información sobre las particiones personalizadas compatibles, consulte Particiones personalizadas.

   Seleccione Next (Siguiente).

8. En la página Revisar y crear, revise sus opciones y, a continuación, seleccione Crear un almacén de datos.

> **Important**
>
> No se puede cambiar el ID del almacén de datos, el formato del archivo ni el tipo de datos de una columna después de crear el almacén de datos.

9. Compruebe que el nuevo almacén de datos aparezca en la página de Almacenes de datos.

Particiones personalizadas

AWS IoT Analytics admite la partición de datos para que pueda organizar los datos en su almacén de datos. Cuando utiliza la partición de datos para organizar los datos, puede realizar consultas en los datos depurados. Esto reduce la cantidad de datos que se analizan por consulta y mejora la latencia.

Puede particionar los datos en función de los atributos de los datos de los mensajes o de los atributos añadidos a mediante las actividades de canalización.

Para comenzar, habilite la partición de datos en un almacén de datos. Especifique una o más dimensiones de partición de datos y conecte el almacén de datos particionado a una canalización de AWS IoT Analytics. A continuación, escriba consultas utilizando la cláusula `WHERE` para optimizar el rendimiento.

Crear un almacén de datos (consola)

En el siguiente procedimiento se explica cómo crear un almacén de datos con una partición personalizada.

Para crear un almacén de datos

1. Inicie sesión en la consola de AWS IoT Analytics.
2. En el panel de navegación, seleccione Almacenamiento de datos.
3. En la página Almacenamiento de datos, seleccione Crear un almacén de datos.
4. En la página Especificar detalles del almacén de datos, introduzca la información básica sobre el almacén de datos.

   a. En ID del almacén de datos, introduzca un ID de almacén de datos único. No se puede cambiar este ID después de crearlo.

   b. (Opcional) En el caso de Etiquetas, seleccione Agregar nueva etiqueta para agregar una o más etiquetas personalizadas (pares clave-valor) al almacén de datos. Las etiquetas pueden ayudarlo a identificar los recursos que usted crea en AWS IoT Analytics.

   c. Seleccione Next (Siguiente).
5. En la página Configuración del tipo de almacenamiento, especifique de qué forma se almacenarán los datos.

   a. En Tipo de almacenamiento, seleccione Almacenamiento administrado por servicio.

b. En Configurar durante cuánto tiempo desea conservar los datos procesados, seleccione Indefinidamente.

c. Seleccione Next (Siguiente).

6. En la página Configurar formato de datos, defina la estructura y el formato de los registros de datos.

a. Para el formato de datos de su almacén de datos Clasificación, seleccione JSON o Parquet. Para más información sobre los AWS IoT Analytics compatibles, consulte Formatos de archivo.

> **Note**
>
> No se puede cambiar este formato después de crear el almacén de datos.

b. Seleccione Next (Siguiente).

7. Creación de particiones personalizadas para este almacén de datos.

a. En Agregar particiones de datos, elija Habilitar.

b. En Fuente de las particiones de datos, especifique la información básica sobre el origen de su partición.

Seleccione Fuente de muestra y elija el canal de AWS IoT Analytics que recopila los mensajes para este almacén de datos.

c. En Atributos de mensajes de muestra, seleccione los atributos de mensaje que desee usar para particionar el almacén de datos. A continuación, añada sus selecciones como dimensiones de la partición de atributos o dimensiones de la partición de marca de tiempo en Acciones.

> **Note**
>
> Solo se puede agregar una partición de marca de tiempo al almacén de datos.

d. En el caso de Dimensiones de la partición personalizada del almacén de datos, defina la información básica sobre las dimensiones de la partición. Cada atributo de mensajes de muestra que haya seleccionado en el paso anterior se convertirá en las dimensiones de la partición. Personalice cada dimensión con estas opciones:

- Tipo de partición: especifique si esta dimensión de la partición es de tipo Atributo o Marca de tiempo.
- Nombre del atributo y Nombre de la dimensión: de forma predeterminada, AWS IoT Analytics utilizará el nombre del atributo de mensajes de muestra que haya seleccionado como identificador para la dimensión de la partición de su atributo. Edite el nombre del atributo para personalizar el nombre de la dimensión de la partición. Puede usar el nombre de la dimensión en la cláusula `WHERE` para optimizar el rendimiento de las consultas.
  - El nombre de cualquier dimensión del atributo de la partición lleva el prefijo `__partition_`.
  - En el caso de los tipos de particiones de marca de tiempo, AWS IoT Analytics crea estas cuatro dimensiones con los nombres `__year`, `__month`, `__day`, `__hour`.
- Ordenar: reorganice las dimensiones de la partición para mejorar la latencia de las consultas.

Para el Formato de marca de tiempo, especifique el formato de la partición de marca de tiempo haciendo coincidir la marca de tiempo ingresada con los datos de sus mensajes. Puede elegir una de las opciones de formato que figura en AWS IoT Analytics o especificar una que coincida con el formato de los datos. Obtenga más información sobre cómo especificar formateadores de fecha y hora.

Para añadir una nueva dimensión que no sea un atributo de mensaje, seleccione Añadir nuevas particiones.

e. Seleccione Next (Siguiente).

8. En la página Revisar y crear, revise sus opciones y, a continuación, seleccione Crear un almacén de datos.

> ⚠ Important
>
> - No puede cambiar el ID del almacén de datos después de crearlo.
> - Para editar las particiones existentes, debe crear otro almacén de datos y volver a procesar los datos mediante una canalización.

9. Compruebe que el nuevo almacén de datos aparezca en la página de Almacenes de datos.

Creación de una canalización

Una canalización consume mensajes de un canal y le permite procesar y filtrar los mensajes antes de guardarlos en un almacén de datos. Para conectar un canal a un almacén de datos, cree una canalización. La canalización más sencilla posible no contiene ninguna otra actividad aparte de la especificación del canal que recopila los datos y la identificación del almacén de datos donde se envían los mensajes. Para obtener información sobre otras canalizaciones más complejas, consulte Actividades de canalización.

Para comenzar, recomendamos que cree una canalización que lo único que haga sea conectar un canal a un almacén de datos. A continuación, después de verificar los flujos de datos sin procesar en el almacén de datos, podrá introducir actividades de canalización adicionales para procesarlos.

Ejecute el siguiente comando para crear una canalización.

```
aws iotanalytics create-pipeline --cli-input-json file://mypipeline.json
```

El archivo `mypipeline.json` contiene el siguiente contenido.

```
{
    "pipelineName": "mypipeline",
    "pipelineActivities": [
        {
            "channel": {
                "name": "mychannelactivity",
                "channelName": "mychannel",
                "next": "mystoreactivity"
            }
        },
        {
            "datastore": {
                "name": "mystoreactivity",
                "datastoreName": "mydatastore"
            }
        }
    ]
}
```

Ejecute el siguiente comando para enumerar las canalizaciones existentes.

```
aws iotanalytics list-pipelines
```

Ejecute el siguiente comando para ver la configuración de una canalización individual.

```
aws iotanalytics describe-pipeline  --pipeline-name mypipeline
```

Incorporación de datos en AWS IoT Analytics

Si tiene un canal que redirige los datos a una canalización que almacena datos en un almacén de datos donde se pueden consultar, podrá enviar datos de mensaje en AWS IoT Analytics. Aquí mostramos dos métodos para la obtención de datos en AWS IoT Analytics. Puede enviar un mensaje mediante el agente de mensajes de AWS IoT o utilice la API `BatchPutMessage` de AWS IoT Analytics.

Temas

Uso del agente de mensajes de AWS IoT

Para utilizar el agente de mensajes de AWS IoT, cree un motor de reglas de AWS IoT. La regla enruta mensajes con un tema específico en AWS IoT Analytics. Sin embargo, en primer lugar, esta regla requiere que cree un rol que conceda los permisos necesarios.

Creación de un rol de IAM

Para disponer de mensajes de AWS IoT dirigidos en un canal de AWS IoT Analytics, debe configurar una regla. Sin embargo, debe crear primero un rol de IAM que conceda a esa regla permiso para enviar datos de mensajes a un canal de AWS IoT Analytics.

Ejecute el siguiente comando para crear el rol.

```
aws iam create-role --role-name myAnalyticsRole --assume-role-policy-document file://
arpd.json
```

El contenido del archivo `arpd.json` debe ser similar al siguiente.

```
{
    "Version": "2012-10-17",
```

```
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "iot.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

A continuación, asocie un documento de política al rol.

```
aws iam put-role-policy --role-name myAnalyticsRole --policy-name myAnalyticsPolicy --
policy-document file://pd.json
```

El contenido del archivo `pd.json` debe ser similar al siguiente.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "iotanalytics:BatchPutMessage",
            "Resource": [
               "arn:aws:iotanalytics:us-west-2:your-account-number:channel/mychannel"
            ]
        }
    ]
}
```

Creación de una regla de AWS IoT

Cree una regla de AWS IoT que envíe mensajes a su canal.

```
aws iot create-topic-rule --rule-name analyticsTestRule --topic-rule-payload file://
rule.json
```

El contenido del archivo `rule.json` debe ser similar al siguiente.

```
{
```

```
    "sql": "SELECT * FROM 'iot/test'",
    "ruleDisabled": false,
    "awsIotSqlVersion": "2016-03-23",
    "actions": [ {
        "iotAnalytics": {
            "channelName": "mychannel",
            "roleArn": "arn:aws:iam::your-account-number:role/myAnalyticsRole"
        }
    } ]
}
```

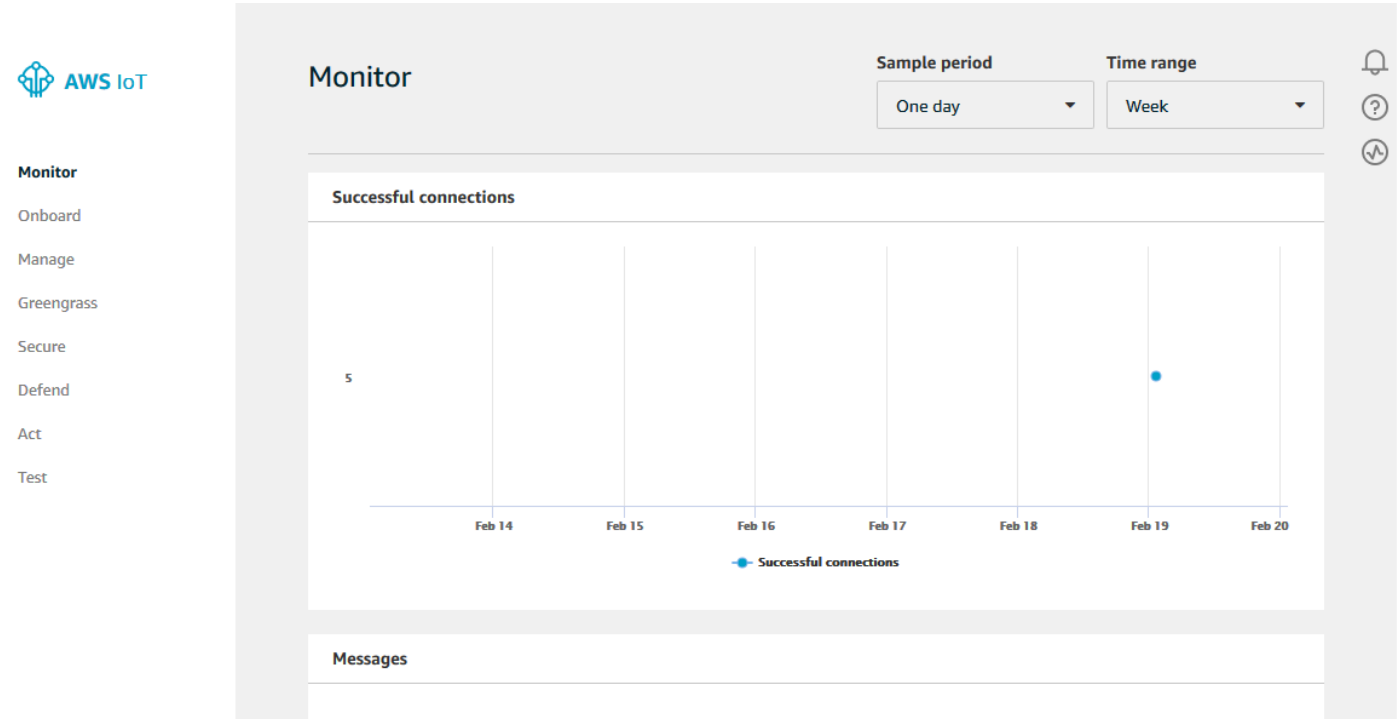
Sustituya `iot/test` por el tema de MQTT de los mensajes que deben enrutarse. Sustituya el nombre del canal y el rol por los creados en las secciones anteriores.

Envío de mensajes MQTT a AWS IoT Analytics

Una vez que anexe una regla a un canal, un canal a una canalización, y una canalización a un almacén de datos, los datos que coinciden con la regla fluirán ahora a través de AWS IoT Analytics para el almacén de datos preparado para la consulta. Para probar esto, puede utilizar la consola de AWS IoT para enviar un mensaje.

> **Note**
>
> Los nombres de los campos de las cargas de mensajes (datos) que se envían a AWS IoT Analytics.
>
> - Solo deben contener caracteres alfanuméricos y guiones bajos (_); no se permiten otros caracteres especiales.
> - Deben comenzar por un carácter alfabético o un carácter de subrayado (_).
> - No pueden contener guiones (-).
> - En términos de expresión regular: "`^[A-Za-z_]([A-Za-z0-9]*|[A-Za-z0-9][A-Za-z0-9_]*)$`".
> - No pueden tener más de 255 caracteres.
> - No distinguen entre mayúsculas y minúsculas. Los campos denominados `foo` y `FOO` en la misma carga se considerarán duplicados.
>
> Por ejemplo, `{"temp_01": 29}` o `{"_temp_01": 29}` son válidos, pero `{"temp-01": 29}`, `{"01_temp": 29}` o `{"__temp_01": 29}` no son válidos en cargas de mensajes.

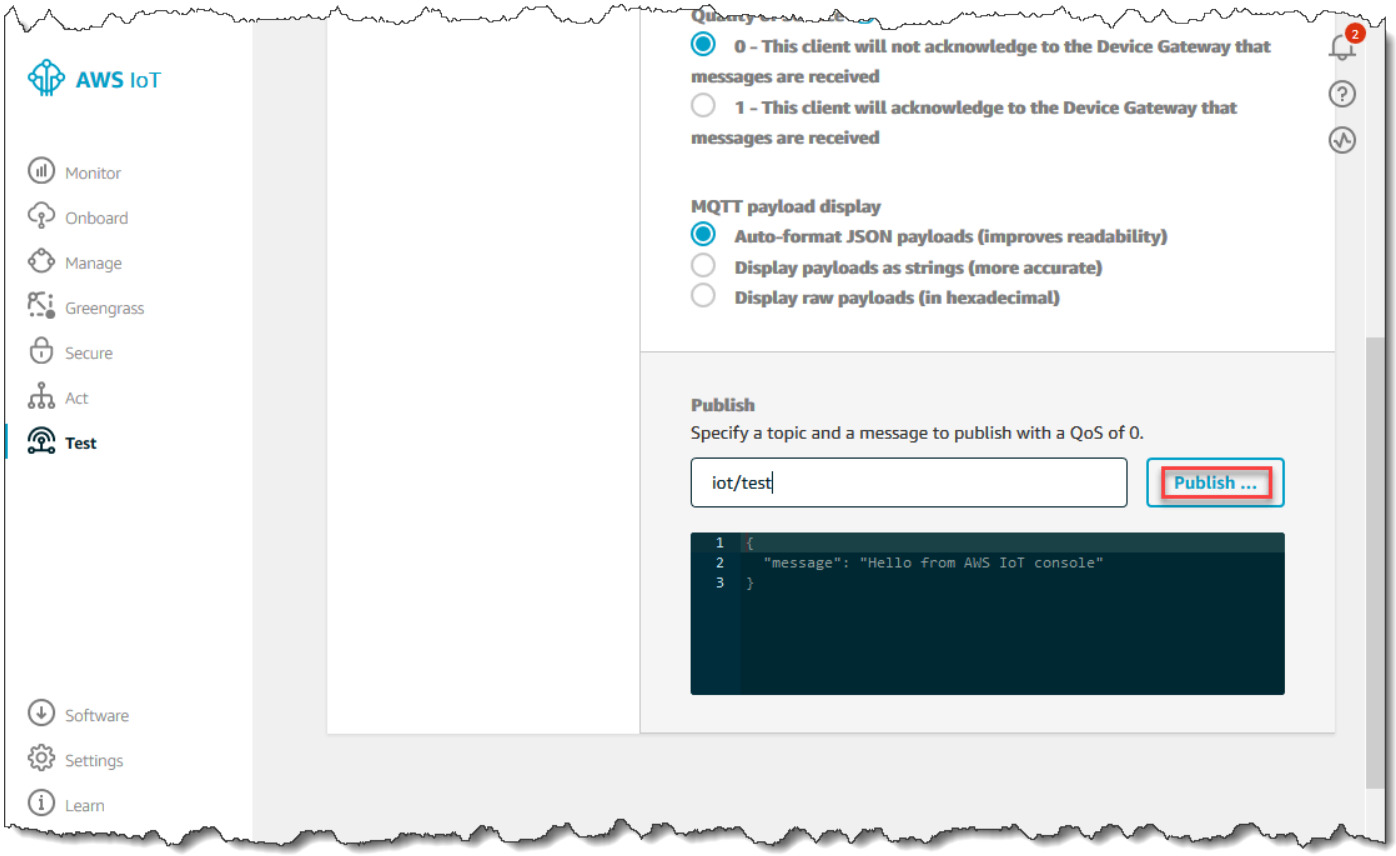
1. En la consola de AWS IoT, en el panel de navegación izquierdo, seleccione Test (Probar).



2. En la página Cliente de MQTT, en la sección Publish (Publicar), en Specify a topic (Especificar un tema), escriba **`iot/test`**. En la sección de carga del mensaje, verifique que los siguientes contenidos JSON están presentes, o escríbalos si no es así.

```
{
    "message": "Hello from the IoT console"
}
```

3. Seleccione Publish to topic (Publicar en el tema).



Esta publica un mensaje que se dirige al almacén de datos que se creó anteriormente.

Uso de la API BatchputMessage

Otra forma de obtener datos de mensajes en AWS IoT Analytics es usar el comando `BatchPutMessage` de la API. Este método no requiere la configuración de una regla de AWS IoT para dirigir al canal los mensajes con un tema específico. Sin embargo, sí requiere que el dispositivo que envía sus datos/mensajes al canal sea capaz de ejecutar software creado con el SDK de AWS o de utilizar AWS CLI para llamar a `BatchPutMessage`.

1. Cree un archivo `messages.json` que contenga los mensajes que se van a enviar (en este ejemplo, solo se envía un mensaje):

```
[
    { "messageId": "message01", "payload": "{ \"message\": \"Hello from the CLI
\" }" }
]
```

2. Ejecute el comando `batch-put-message`.

```
aws iotanalytics batch-put-message  --channel-name mychannel  --messages file://
messages.json --cli-binary-format raw-in-base64-out
```

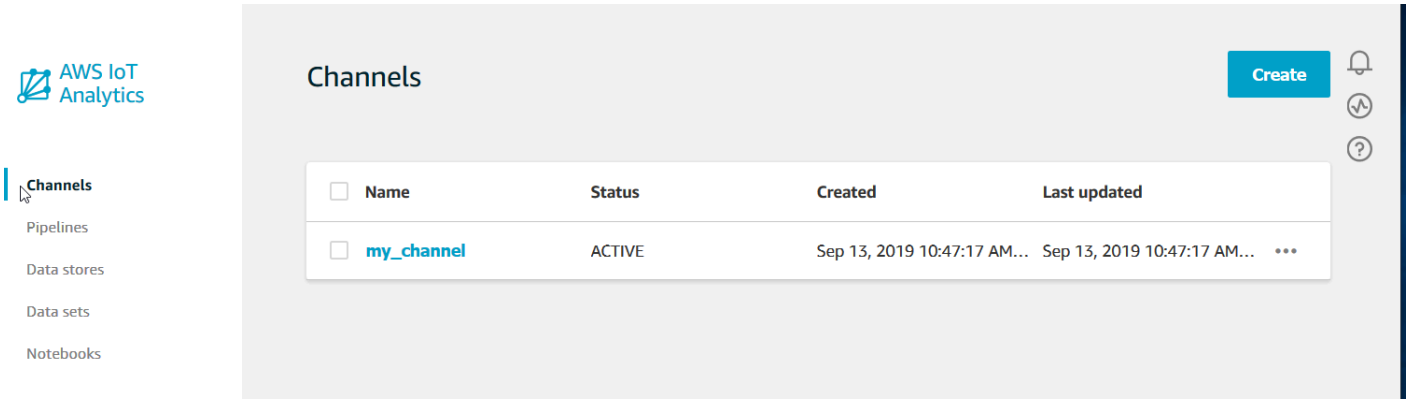
Si no hay errores, verá el siguiente resultado.

```
{
    "batchPutMessageErrorEntries": []
}
```

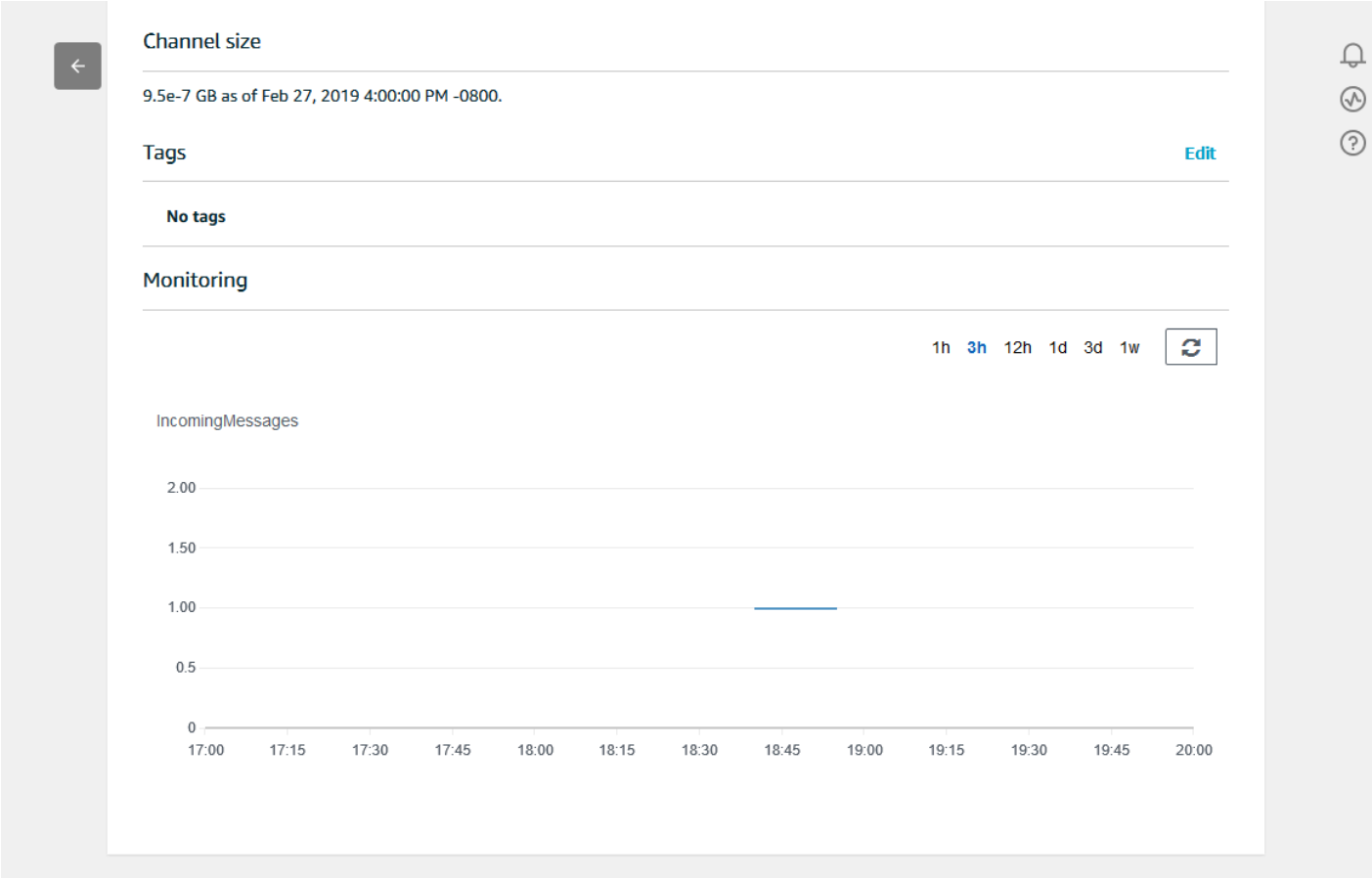
Monitorización de la ingesta de datos

Puede comprobar que los mensajes que ha enviado estén insertados en su canal mediante la consola de AWS IoT Analytics.

1. En la consola de AWS IoT Analytics, en el panel de navegación izquierdo, seleccione Preparar y (si es necesario) elija Canales. A continuación, seleccione el nombre del canal que creó anteriormente.

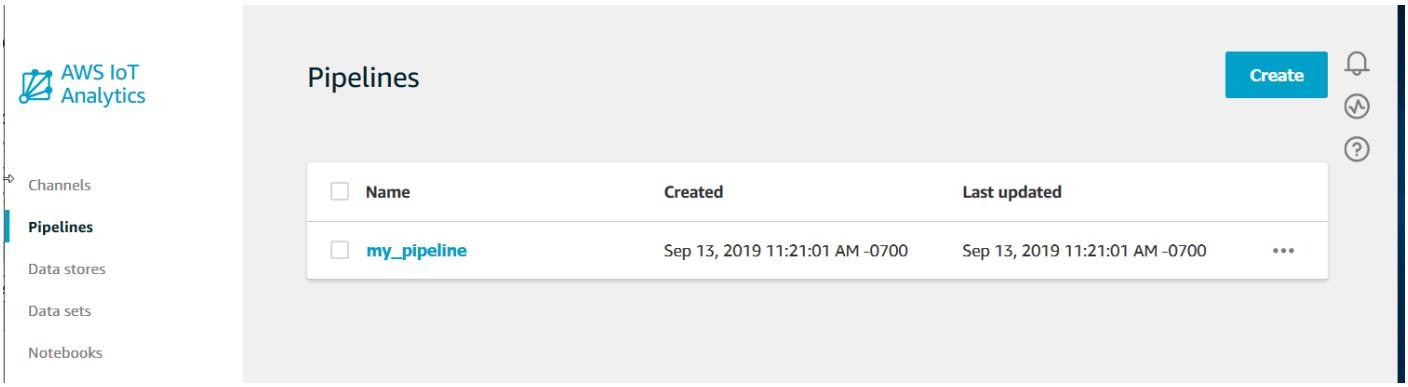


2. En la página de detalles del canal, vaya a la sección Monitoring (Monitorización). Ajuste el período mostrado según sea necesario seleccionando uno de los indicadores de tiempo (1 h 3 h 12 h 1 d 3 d 1 w). Debe ver una línea de gráficos que indica el número de mensajes insertados en este canal durante el período especificado.

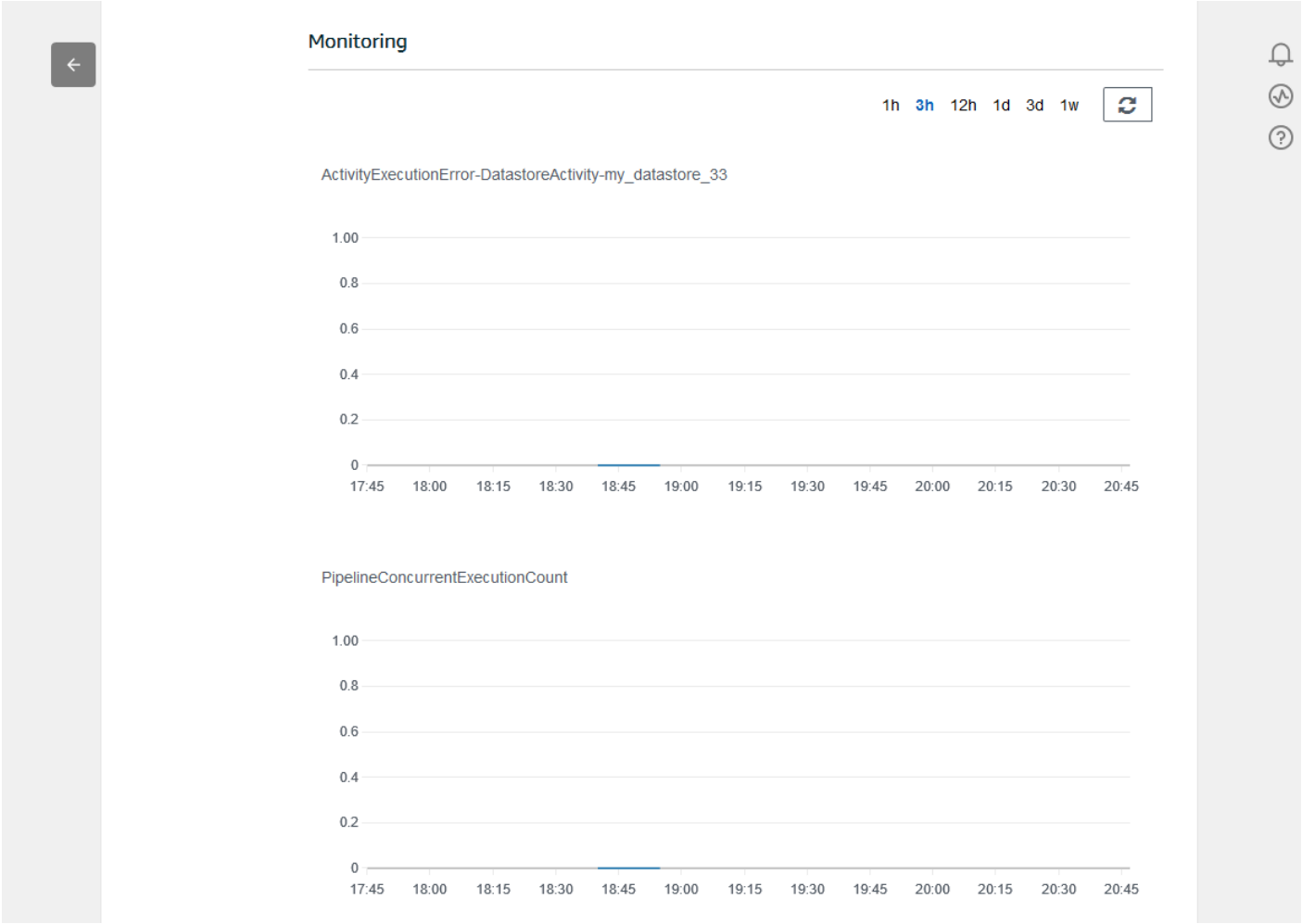


Existe una capacidad de monitorización similar para comprobar las ejecuciones de actividad de canalización. Puede monitorizar los errores de ejecución de la actividad en la página de detalles de la canalización. Si no ha especificado actividades como parte de la canalización, no se deben mostrar errores de ejecución.

1. En la consola de AWS IoT Analytics, en el panel de navegación izquierdo, seleccione Preparar y, a continuación, Canales y elija el nombre de un pipeline que haya creado anteriormente.



2. En la página de detalles de la canalización, vaya a la sección Monitoring (Monitorización). Ajuste el período mostrado según sea necesario seleccionando uno de los indicadores de tiempo (1 h 3 h 12 h 1 d 3 d 1 w). Debería ver una línea de gráficos que indica la cantidad de errores de ejecución de la actividad de canalización durante el período especificado.



Creación de un conjunto de datos

Para recuperar datos de un almacén de datos, cree un conjunto de datos SQL o un conjunto de datos de contenedor. AWS IoT Analytics puede consultar los datos para responder a preguntas analíticas. Aunque un almacén de datos no es una base de datos, se utilizan expresiones SQL para consultar los datos y producir resultados que se almacenan en un conjunto de datos.

Temas

- Consulta de datos
- Acceso a los datos consultados

Consulta de datos

Para consultar los datos, se crea un conjunto de datos. Un conjunto de datos contiene el SQL que se utiliza para consultar el almacén de datos, junto con una programación opcional que repite la consulta en el día y la hora que se elijan. Las programaciones opcionales se crean con expresiones similares a las expresiones de programación de Amazon CloudWatch.

Ejecute el siguiente comando para crear un conjunto de datos.

```
aws iotanalytics create-dataset --cli-input-json file://mydataset.json
```

Donde el archivo `mydataset.json` incluye el siguiente contenido.

```
{
    "datasetName": "mydataset",
    "actions": [
        {
            "actionName":"myaction",
            "queryAction": {
                "sqlQuery": "select * from mydatastore"
            }
        }
    ]
}
```

Ejecute el siguiente comando para crear el contenido del conjunto de datos ejecutando la consulta.

```
aws iotanalytics create-dataset-content --dataset-name mydataset
```

Espere unos minutos a que se cree el contenido del conjunto de datos antes de continuar.

Acceso a los datos consultados

El resultado de la consulta es su contenido de conjuntos de datos almacenados como un archivo en formato CSV. El archivo se pone a su disposición a través de Amazon S3. En el siguiente ejemplo, se muestra cómo puede comprobar que los resultados están listos y descargar el archivo.

Ejecute el siguiente comando `get-dataset-content`.

```
aws iotanalytics get-dataset-content --dataset-name mydataset
```

Si su conjunto de datos contiene algún dato, entonces la salida de `get-dataset-content`, tiene `"state": "SUCCEEDED"` en el campo `status`, como en el siguiente ejemplo.

```
{
    "timestamp": 1508189965.746,
    "entries": [
        {
          "entryName": "someEntry",
          "dataURI": "https://aws-iot-analytics-datasets-f7253800-859a-472c-aa33-
e23998b31261.s3.amazonaws.com/results/f881f855-c873-49ce-abd9-b50e9611b71f.csv?X-Amz-"

        }
    ],
    "status": {
      "state": "SUCCEEDED",
      "reason": "A useful comment."
    }
}
```

`dataURI` es una URL firmada a los resultados de salida. Es válida durante un breve periodo de tiempo (unas pocas horas). En función de su flujo de trabajo, es posible que desee llamar siempre a `get-dataset-content` antes de tener acceso al contenido, ya que al llamar a este comando se genera una nueva URL firmada.

Exploración de datos de AWS IoT Analytics

Dispone de varias opciones para almacenar, analizar y visualizar sus datos de AWS IoT Analytics.

Temas en esta página:

Amazon S3

Puede enviar contenido de un conjunto de datos a un bucket de Amazon Simple Storage Service (Amazon S3), lo que permite la integración con los lagos de datos existentes o el

acceso desde aplicaciones internas y herramientas de visualización. Consulte el campo `contentDeliveryRules::destination::s3DestinationConfiguration` en CreateDataset.

AWS IoT Events

Puede enviar contenido del conjunto de datos como una entrada a AWS IoT Events, un servicio que le permite monitorizar dispositivos o procesos para ver si se producen errores o cambios en la operación, y para activar acciones adicionales cuando se producen estos eventos.

Para ello, cree un conjunto de datos mediante CreateDataset y especifique una entrada de AWS IoT Events en el campo `contentDeliveryRules :: destination :: iotEventsDestinationConfiguration :: inputName`. También debe especificar el `roleArn` del rol que concede a AWS IoT Analytics el permiso para ejecutar "IoTEvents:BatchPutMessage". Siempre que se cree el contenido del conjunto de datos, AWS IoT Analytics enviará cada entrada de contenido del conjunto de datos como un mensaje a la entrada especificada de AWS IoT Events. Por ejemplo, si su conjunto de datos contiene:

```
"what","who","dt"
"overflow","sensor01","2019-09-16 09:04:00.000"
"overflow","sensor02","2019-09-16 09:07:00.000"
"underflow","sensor01","2019-09-16 11:09:00.000"
...
```

entonces AWS IoT Analytics enviará mensajes que contengan campos como estos:

```
{ "what": "overflow", "who": "sensor01", "dt": "2019-09-16 09:04:00.000" }
```

```
{ "what": "overflow", "who": "sensor02", "dt": "2019-09-16 09:07:00.000" }
```

y querrá crear una entrada de AWS IoT Events que reconozca los campos que le interesan (uno o más de `what`,`who`,`dt`) y crear un modelo de detector de AWS IoT Events que utilice estos campos de entrada en eventos para activar acciones o establecer variables internas.

Amazon QuickSight

AWS IoT Analytics ofrece integración directa con Amazon QuickSight. Amazon QuickSight es un rápido servicio de análisis empresariales que puede utilizar para crear visualizaciones, realizar análisis ad hoc y obtener rápidamente información empresarial útil a partir de sus datos. Amazon QuickSight permite a las organizaciones escalar de cientos a miles de usuarios y ofrece

un rendimiento fiable gracias a su sólido motor en memoria (SPICE). Amazon QuickSight está disponible en estas regiones.

Cuaderno de Jupyter

Los conjuntos de datos AWS IoT Analytics también pueden ser consumidos directamente por el cuaderno de Jupyter para realizar análisis avanzados y exploración de datos. El cuaderno de Jupyter es una solución de código abierto. Puede descargarla e instalarla desde http://jupyter.org/install.html. También se dispone de integración adicional con SageMaker, una solución de bloc de notas alojada en Amazon.

Conservación de varias versiones de conjuntos de datos

Puede elegir la cantidad de versiones del contenido del conjunto de datos que desea retener, y durante cuánto tiempo, especificando valores para los campos `retentionPeriod and versioningConfiguration` del conjunto de datos al invocar las API CreateDataset y UpdateDataset:

```
...
"retentionPeriod": {
  "unlimited": "boolean",
  "numberOfDays": "integer"
},
"versioningConfiguration": {
  "unlimited": "boolean",
  "maxVersions": "integer"
},
...
```

La configuración de estos dos parámetros se combina para determinar la cantidad de versiones de contenido del conjunto de datos que se conserva, y durante cuánto tiempo, de las siguientes maneras.

	retentionPeriod	retentionPeriod:	retentionPeriod:
	[no especificado]	ilimitado = TRUE, numberOfDays = no establecido	ilimitado = FALSE, numberOfDays = X

versioningConfiguration: [no especificado]	Solo la versión más reciente, además de la versión correcta más reciente (si son diferentes) se conservan durante 90 días.	Solo la versión más reciente, además de la versión correcta más reciente (si son diferentes) se conservan por tiempo ilimitado.	Solo la versión más reciente, además de la versión correcta más reciente (si son diferentes) se conservan durante X días.
versioningConfiguration: ilimitado = TRUE, maxVersions no establecido	Se conservarán todas las versiones de los últimos 90 días, independientemente de la cantidad.	No existe ningún límite para el número de versiones conservadas.	Se conservarán todas las versiones de los últimos X días, independientemente de la cantidad.
versioningConfiguration: ilimitado = FALSE, maxVersions = Y	No se conservarán más de Y versiones de los últimos 90 días.	Se conservarán hasta Y versiones independientemente de su antigüedad.	No se conservarán más de Y versiones de los últimos X días.

Sintaxis de carga de mensajes

Los nombres de los campos de las cargas de mensajes (datos) que se envían a AWS IoT Analytics:

- Solo deben contener caracteres alfanuméricos y guiones bajos (_); no se permiten otros caracteres especiales.
- Deben comenzar por un carácter alfabético o un carácter de subrayado (_).
- No pueden contener guiones (-).
- En términos de expresión regular: "`^[A-Za-z_]([A-Za-z0-9]*|[A-Za-z0-9][A-Za-z0-9_]*)$`".
- No pueden tener más de 255 caracteres.
- No distinguen entre mayúsculas y minúsculas. Los campos denominados "foo" y "FOO" en la misma carga se considerarán duplicados.

Por ejemplo, `{"temp_01": 29}` o `{"_temp_01": 29}` son válidos, pero `{"temp-01": 29}`, `{"01_temp": 29}` o `{"__temp_01": 29}` no son válidos en cargas de mensajes.

Trabajar con datos de AWS IoT SiteWise

AWS IoT SiteWise es un servicio administrado que se puede utilizar para recopilar, modelar, analizar y visualizar datos de equipos industriales a escala. El servicio proporciona un marco de modelado de recursos para crear representaciones de sus dispositivos industriales, procesos e instalaciones.

Con los modelos de recursos de AWS IoT SiteWise, puede definir qué datos de equipos industriales consumir y cómo procesar sus datos en métricas complejas. Puede configurar modelos de activos para recopilar y procesar datos en la nube de AWS. Para obtener más información, consulte la Guía de usuario de la AWS IoT SiteWise.

AWS IoT Analytics se integra con AWS IoT SiteWise para que pueda ejecutar y programar consultas SQL sobre los datos de AWS IoT SiteWise. Para empezar a consultar los datos de AWS IoT SiteWise, cree un almacén de datos siguiendo los procedimientos descritos en Configuración de los ajustes de almacenamiento de la Guía del usuario de AWS IoT SiteWise. A continuación, siga los pasos indicados en Creación de un conjunto de datos con datos de AWS IoT SiteWise (consola) o en Creación de un conjunto de datos con datos de AWS IoT SiteWise (AWS CLI) para crear un conjunto de datos de AWS IoT Analytics y ejecutar una consulta SQL en sus datos industriales.

Temas

- Creación de un conjunto de datos de AWS IoT Analytics con datos de AWS IoT SiteWise
- Acceso al contenido de conjunto de datos
- Tutorial: Consulta AWS IoT SiteWise de datos en AWS IoT Analytics

Creación de un conjunto de datos de AWS IoT Analytics con datos de AWS IoT SiteWise

Un conjunto de datos AWS IoT Analytics contiene sentencias y expresiones SQL que se utilizan para consultar los datos del almacén de datos junto con una programación opcional que repite la consulta en el día y la hora especificados. Puede utilizar expresiones similares a expresiones de programación de Amazon CloudWatch para crear las programaciones opcionales.

> **Note**
>
> Un conjunto de datos suele ser un conjunto de datos que puede o no estar organizado en forma tabular. Por el contrario, AWS IoT Analytics crea el conjunto de datos mediante la aplicación de una consulta SQL a los datos del almacén de datos.

Siga estos pasos para empezar a crear un conjunto de datos para sus datos de AWS IoT SiteWise.

Temas

Creación de un conjunto de datos con datos de AWS IoT SiteWise (consola)

Siga estos pasos para crear un conjunto de datos en la consola de AWS IoT Analytics para sus datos de AWS IoT SiteWise.

Para crear un conjunto de datos

1. En https://console.aws.amazon.com/iotanalytics/, en el panel de navegación izquierdo, seleccione Conjuntos de datos.
2. En la página Crear conjunto de datos, seleccione Crear SQL.
3. En la página Especificar los detalles del conjunto de datos, especifique los detalles del conjunto de datos.

   a. Escriba un nombre para el conjunto de datos.

   b. En Fuente del almacén de datos, seleccione el ID único que identifica a su almacén de datos de AWS IoT SiteWise.

   c. (Opcional) En Etiquetas, agregue una o más etiquetas personalizadas (pares clave-valor) al conjunto de datos.
4. Utilice expresiones SQL para consultar los datos y responder a las preguntas analíticas.

   a. En el campo Consulta de autor, introduzca una consulta SQL que utilice un comodín para mostrar hasta cinco filas de datos.

```
SELECT * FROM my_iotsitewise_datastore.asset_metadata LIMIT 5
```

Para obtener más información sobre las funciones de SQL compatibles en AWS IoT Analytics, consulte Expresiones SQL en AWS IoT Analytics. O bien, consulte Tutorial: Consulta AWS IoT SiteWise de datos en AWS IoT Analytics para ver ejemplos de consultas estadísticas que pueden brindarle información sobre sus datos.

b. Puede elegir Probar consulta para validar que la entrada es correcta y para mostrar los resultados en una tabla después de la consulta.

> **Note**
>
> Como Amazon Athena limita el número máximo de consultas en ejecución, debe limitar la consulta SQL a un tamaño razonable para que no se ejecute durante un período prolongado.

5. (Opcional) Al crear el contenido de un conjunto de datos con datos de un período de tiempo específico, es posible que algunos datos no lleguen a tiempo para su procesamiento. Para permitir un retraso, puede especificar un desplazamiento o tiempo delta. Para obtener más información, consulte Obtención de notificaciones de datos atrasados a través de Eventos de Amazon CloudWatch.

   Tras configurar un filtro de selección de datos en la página Configurar filtro de selección de datos, seleccione Siguiente.

6. (Opcional) En la página Definir programación de la consulta, puede programar esta consulta para que se ejecute de forma regular a fin de actualizar el conjunto de datos. Las programaciones de los conjuntos de datos se pueden crear y editar en cualquier momento.

   > **Note**
   >
   > Los datos de AWS IoT SiteWise se ingieren en AWS IoT Analytics cada seis horas. Recomendamos seleccionar una frecuencia de seis horas o más.

   Elija una opción para Frecuencia y después Siguiente.

7. AWS IoT Analytics creará versiones del contenido de este conjunto de datos y almacenará los resultados de sus análisis durante el período especificado. La recomendación es 90 días, pero puede optar por establecer su propia política de retención personalizada. También puede limitar el número de versiones almacenadas del contenido de su conjunto de datos.

Tras seleccionar las opciones en la página Configurar los resultados del conjunto de datos, seleccione Siguiente.

8. (Opcional) Puede configurar las reglas de entrega de los resultados de su conjunto de datos a un destino específico, por ejemplo AWS IoT Events.

   Tras seleccionar las opciones en la página Configurar las reglas de entrega de contenido del conjunto de datos, seleccione Siguiente.

9. Revise las opciones seleccionadas y, a continuación, elija Crear conjunto de datos.
10. Compruebe que su nuevo conjunto de datos aparezca en la página Conjuntos de datos.

Creación de un conjunto de datos con datos de AWS IoT SiteWise (AWS CLI)

Ejecute los siguientes comandos de la AWS CLI para empezar a consultar sus datos de AWS IoT SiteWise.

Los ejemplos que se muestran aquí utilizan la AWS Command Line Interface (AWS CLI). Para obtener más información sobre AWS CLI, consulte la Guía del usuario de AWS Command Line Interface. Para obtener más información sobre los comandos de la CLI disponibles en AWS IoT Analytics, consulte iotanalytics en la Referencia de la AWS Command Line Interface.

Para crear un conjunto de datos

1. Ejecute el siguiente comando de `create-dataset` para crear un conjunto de datos.

   ```
   aws iotanalytics create-dataset --cli-input-json file://my_dataset.json
   ```

   Donde el archivo `my_dataset.json` incluye el siguiente contenido.

   ```
   {
       "datasetName": "my_dataset",
       "actions": [
           {
               "actionName":"my_action",
               "queryAction": {
                   "sqlQuery": "SELECT * FROM my_iotsitewise_datastore.asset_metadata
    LIMIT 5"
               }
           }
   ```

```
    ]
}
```

Para obtener más información sobre las funciones de SQL compatibles en AWS IoT Analytics, consulte Expresiones SQL en AWS IoT Analytics. O bien, consulte Tutorial: Consulta AWS IoT SiteWise de datos en AWS IoT Analytics para ver ejemplos de consultas estadísticas que pueden brindarle información sobre sus datos.

2. Ejecute el siguiente comando `create-dataset-content` para crear el contenido del conjunto de datos ejecutando su consulta.

```
aws iotanalytics create-dataset-content --dataset-name my_dataset
```

Acceso al contenido de conjunto de datos

El resultado de la consulta SQL es el contenido de los conjuntos de datos, que se almacenan como un archivo en formato CSV. El archivo se pone a su disposición a través de Amazon S3. Los siguientes pasos muestran cómo puede comprobar que los resultados están listos y descargar el archivo.

Temas

- Acceso al contenido de conjunto de datos en AWS IoT Analytics (consola)
- Acceso al contenido del conjunto de datos en AWS IoT Analytics (AWS CLI)

Acceso al contenido de conjunto de datos en AWS IoT Analytics (consola)

Si su conjunto de datos contiene datos, puede obtener una vista previa y descargar los resultados de su consulta SQL en la consola de AWS IoT Analytics.

Para acceder a los resultados de su conjunto de datos de AWS IoT Analytics

1. En la consola, en la página Conjuntos de datos, seleccione el nombre del conjunto de datos al que quiere acceder.
2. En la página de resumen de conjunto de datos, seleccione la pestaña Contenido.
3. En la tabla Contenido del conjunto de datos, seleccione el nombre de la consulta de la que desea obtener una vista previa de los resultados o descargar un archivo csv con los resultados.

Acceso al contenido del conjunto de datos en AWS IoT Analytics (AWS CLI)

Si su conjunto de datos contiene datos, puede obtener una vista previa y descargar los resultados de su consulta SQL.

Los ejemplos que se muestran aquí utilizan la AWS Command Line Interface (AWS CLI). Para obtener más información sobre AWS CLI, consulte la Guía del usuario de AWS Command Line Interface. Para obtener más información sobre los comandos de la CLI disponibles en AWS IoT Analytics, consulte iotanalytics en la Referencia de la AWS Command Line Interface.

Para acceder a los resultados de su conjunto de datos de AWS IoT Analytics (AWS CLI)

1. Ejecute el siguiente comando `get-dataset-content` para ver el resultado de la consulta.

```
aws iotanalytics get-dataset-content --dataset-name my_iotsitewise_dataset
```

2. Si el conjunto de datos contiene datos, el resultado de `get-dataset-content`, mostrará `"state": "SUCCEEDED"` en el campo `status`, como en el siguiente ejemplo.

```
{
    "timestamp": 1508189965.746,
    "entries": [
        {
          "entryName": "my_entry_name",
          "dataURI": "https://aws-iot-analytics-datasets-f7253800-859a-472c-aa33-
e23998b31261.s3.amazonaws.com/results/f881f855-c873-49ce-abd9-b50e9611b71f.csv?X-
Amz-"

        }
    ],
    "status": {
      "state": "SUCCEEDED",
      "reason": "A useful comment."
    }
}
```

3. El resultado de `get-dataset-content` incluye una `dataURI`, que es una URL firmada a los resultados de salida. Es válida durante un breve periodo de tiempo (unas pocas horas). Visite la URL `dataURI` para acceder a los resultados de su consulta SQL.

> **Note**
>
> En función de su flujo de trabajo, es posible que desee llamar siempre a `get-dataset-content` antes de tener acceso al contenido, ya que al llamar a este comando se genera una nueva URL firmada.

Tutorial: Consulta AWS IoT SiteWise de datos en AWS IoT Analytics

En este tutorial se muestra cómo consultar AWS IoT SiteWise datos en AWS IoT Analytics. El tutorial utiliza datos de una demostración en la AWS IoT SiteWise que se proporciona un conjunto de datos de muestra para un parque eólico.

> **Important**
>
> Se le cobrará por los recursos que esta demostración cree y consuma.

Temas

Requisitos previos

Para este tutorial, necesita los siguientes recursos:

- Debe tener una AWS cuenta para empezar AWS IoT SiteWise y AWS IoT Analytics. Si no tiene una, siga los procedimientos descritos en Para crear una cuenta de AWS.
- Un equipo de desarrollo con Windows, macOS, Linux o Unix para acceder a la AWS Management Console. Para obtener más información, consulte Introducción a AWS Management Console.
- AWS IoT SiteWise datos que definen AWS IoT SiteWise modelos y activos y transmiten datos que representan datos de equipos de parques eólicos. Para crear sus datos, siga los pasos que

se indican en la Guía del AWS IoT SiteWise usuario sobre cómo crear la AWS IoT SiteWise demostración.

- Los datos de su equipo de parque eólico de AWS IoT SiteWise demostración están en un almacén de datos existente que usted administra. Para obtener más información sobre cómo crear un banco de datos para sus AWS IoT SiteWise datos, consulte Configurar los ajustes de almacenamiento en la Guía del AWS IoT SiteWise usuario.

> **Note**
>
> AWS IoT SiteWise Los metadatos aparecen en el almacén de AWS IoT SiteWise datos poco después de su creación; sin embargo, los datos sin procesar pueden tardar hasta seis horas en aparecer. Mientras tanto, puede crear un AWS IoT Analytics conjunto de datos y realizar consultas en sus metadatos.

Siguiente paso

Carga y verificación de datos

Carga y verificación de datos

Los datos que consulta en este tutorial son un conjunto de AWS IoT SiteWise datos de muestra que modela las turbinas de los motores eólicos de un parque eólico.

> **Note**
>
> A lo largo de este tutorial, consultará tres tablas del almacén de datos:
>
> - `raw`: contiene datos sin procesar de cada activo.
> - `asset_metadata`: contiene información general sobre cada activo.
> - `asset_hierarchy_metadata`: contiene información sobre las relaciones entre los activos.

Para realizar consultas SQL en este tutorial

1. Sigue los pasos que se indican en Creación de un conjunto de datos con datos de AWS IoT SiteWise (consola) o Creación de un conjunto de datos con datos de AWS IoT SiteWise (AWS CLI) para crear un AWS IoT Analytics conjunto de datos para tus AWS IoT SiteWise datos.

2. Para actualizar la consulta del conjunto de datos en este tutorial, haga lo siguiente.

   a. En la AWS IoT Analytics consola, en la página Conjuntos de datos, elige el nombre del conjunto de datos que creaste en la página anterior.

   b. En la página de resumen del conjunto de datos, seleccione Editar para editar la consulta SQL.

   c. Para mostrar los resultados en una tabla después de la consulta, seleccione Probar consulta.

   Como alternativa, puede ejecutar el siguiente comando `update-dataset` para modificar la consulta SQL con la AWS CLI.

   ```
   aws iotanalytics update-dataset --cli-input-json file://update-query.json
   ```

   Contenidos de `update-query.json`:

   ```
   {
       "datasetName": "my_dataset",
       "actions": [
           {
               "actionName": "myDatasetUpdateAction",
               "queryAction": {
                   "sqlQuery": "SELECT * FROM my_iotsitewise_datastore.asset_metadata
    LIMIT 3"
               }
           }
       ]
   }
   ```

3. En la AWS IoT Analytics consola o con el AWS CLI, ejecuta la siguiente consulta en tus datos para comprobar que la `asset_metadata` tabla se ha cargado correctamente.

   ```
   SELECT COUNT(*) FROM my_iotsitewise_datastore.asset_metadata
   ```

   Del mismo modo, puede comprobar que sus tablas `asset_hierarchy_metadata` y `raw` no estén vacías.

Paso siguiente

Exploración de datos

Exploración de datos

Una vez creados AWS IoT SiteWise los datos y cargados en un banco de datos, puede crear un AWS IoT Analytics conjunto de datos y ejecutar consultas SQL AWS IoT Analytics para obtener información sobre sus activos. Las siguientes consultas muestran cómo puede explorar sus datos antes de ejecutar consultas estadísticas.

Para explorar datos con consultas SQL

1. Vea una muestra de columnas y valores en cada tabla, por ejemplo, en la tabla de datos sin procesar.

   ```
   SELECT * FROM my_iotsitewise_datastore.raw LIMIT 5
   ```

2. Úselo `SELECT DISTINCT` para consultar `asset_metadata` la tabla y enumerar los nombres (únicos) de sus AWS IoT SiteWise activos.

   ```
   SELECT DISTINCT assetname FROM my_iotsitewise_datastore.asset_metadata ORDER BY
    assetname
   ```

3. Para obtener información sobre las propiedades de un AWS IoT SiteWise activo concreto, utilice la `WHERE` cláusula.

   ```
   SELECT assetpropertyname,
        assetpropertyunit,
        assetpropertydatatype
   FROM my_iotsitewise_datastore.asset_metadata
   WHERE assetname = 'Demo Turbine Asset 2'
   ```

4. Con AWS IoT Analytics, puede unir datos de dos o más tablas de su banco de datos, como en el siguiente ejemplo.

   ```
   SELECT * FROM my_iotsitewise_datastore.raw AS raw
   JOIN my_iotsitewise_datastore.asset_metadata AS asset_metadata
   ON raw.seriesId = asset_metadata.timeseriesId
   ```

Para ver todas las relaciones entre sus activos, utilice la funcionalidad `JOIN` en la siguiente consulta.

```
SELECT DISTINCT parent.assetName as "Parent name",
    child.assetName AS "Child name"
FROM (
    SELECT sourceAssetId AS parent,
            targetAssetId AS child
    FROM my_iotsitewise_datastore.asset_hierarchy_metadata
    WHERE associationType = 'CHILD'
)
AS relations
JOIN my_iotsitewise_datastore.asset_metadata AS child
    ON relations.child = child.assetId
JOIN my_iotsitewise_datastore.asset_metadata AS parent
    ON relations.parent = parent.assetId
```

Siguiente paso

Ejecución de consultas estadísticas

Ejecución de consultas estadísticas

Ahora que ha explorado sus AWS IoT SiteWise datos, puede realizar consultas estadísticas que proporcionen información valiosa sobre sus equipos industriales. Las siguientes consultas muestran parte de la información que se puede recuperar.

Para realizar consultas estadísticas sobre datos de AWS IoT SiteWise demostración de parques eólicos

1. Ejecute el siguiente comando SQL para buscar los valores más recientes de todas las propiedades con valores numéricos para un activo concreto (Demo Turbine Asset 4).

```
SELECT assetName,
    assetPropertyName,
    assetPropertyUnit,
    max_by(value, timeInSeconds) AS Latest
FROM (
    SELECT *,
        CASE assetPropertyDataType
```

```
        WHEN 'DOUBLE' THEN
        cast(doubleValue AS varchar)
        WHEN 'INTEGER' THEN
        cast(integerValue AS varchar)
        WHEN 'STRING' THEN
        stringValue
        WHEN 'BOOLEAN' THEN
        cast(booleanValue AS varchar)
        ELSE NULL
        END AS value
    FROM my_iotsitewise_datastore.asset_metadata AS asset_metadata
    JOIN my_iotsitewise_datastore.raw AS raw
        ON raw.seriesId = asset_metadata.timeSeriesId
    WHERE startYear=2021
        AND startMonth=7
        AND startDay=8
        AND assetName='Demo Turbine Asset 4'
)
GROUP BY assetName, assetPropertyName, assetPropertyUnit
```

2. Una las dos tablas de metadatos y la tabla con datos sin procesar para identificar las propiedades de velocidad máxima del viento para todos los activos, además de sus activos principales.

```
SELECT child_assets_data_set.parentAssetId,
        child_assets_data_set.childAssetId,
        asset_metadata.assetPropertyId,
        asset_metadata.assetPropertyName,
        asset_metadata.timeSeriesId,
        raw_data_set.max_speed
FROM (
   SELECT sourceAssetId AS parentAssetId,
        targetAssetId AS childAssetId
    FROM my_iotsitewise_datastore.asset_hierarchy_metadata
    WHERE associationType = 'CHILD'
)
AS child_assets_data_set
JOIN mls_demo.asset_metadata AS asset_metadata
    ON asset_metadata.assetId = child_assets_data_set.childAssetId
JOIN (
    SELECT seriesId, MAX(doubleValue) AS max_speed
    FROM my_iotsitewise_datastore.raw
    GROUP BY seriesId
```

```
)
AS raw_data_set
ON raw_data_set.seriesId = asset_metadata.timeseriesid
WHERE assetPropertyName = 'Wind Speed'
ORDER BY max_speed DESC
```

3. Para encontrar el valor medio de una propiedad concreta (Wind Speed) para un activo (Demo Turbine Asset 2), ejecute el siguiente comando SQL. Debe reemplazar `my_bucket_id` por el ID de su bucket.

```
SELECT AVG(doubleValue) as "Average wind speed"
FROM my_iotsitewise_datastore.raw
WHERE seriesId =
    (SELECT timeseriesId
    FROM my_iotsitewise_datastore.asset_metadata as asset_metadata
    WHERE asset_metadata.assetname = 'Demo Turbine Asset 2'
            AND asset_metadata.assetpropertyname = 'Wind Speed')
```

Siguiente paso

Limpieza de los recursos del tutorial

Limpieza de los recursos del tutorial

Después de completar el tutorial, limpie los recursos para evitar incurrir en cargos.

Para eliminar tu demo AWS IoT SiteWise

La AWS IoT SiteWise demostración se borra automáticamente después de una semana. Si ha terminado de usar los recursos de la demostración, puede eliminarla antes. Para borrar la demostración manualmente, siga los siguientes pasos.

1. Vaya a la consola de AWS CloudFormation.
2. Seleccione IoTSiteWiseDemoAssets de la lista de Stacks (Pilas).
3. Seleccione Eliminar (Delete). Al eliminar la pila, se eliminan todos los recursos creados para la demostración.
4. En el cuadro de diálogo de confirmación, introduzca Eliminar.

   La pila tarda unos 15 minutos en borrarse. Si la demostración no se elimina, vuelva a seleccionar Delete (Eliminar) en la esquina superior derecha. Si la demostración no se borra de

nuevo, sigue los pasos de la AWS CloudFormation consola para omitir los recursos que no se pudieron eliminar e inténtalo de nuevo.

Para eliminar el almacén de datos

- Para eliminar el almacén de datos administrado, ejecute el comando `delete-datastore` de la CLI, como en el siguiente ejemplo.

```
aws iotanalytics delete-datastore --datastore-name my_IotSiteWise_datastore
```

Para eliminar el AWS IoT Analytics conjunto de datos

- Para eliminar el conjunto de datos, ejecute el comando `delete-dataset`de la CLI, como en el siguiente ejemplo. No es necesario eliminar el contenido del conjunto de datos antes de realizar esta operación.

```
aws iotanalytics delete-dataset --dataset-name my_dataset
```

> **Note**
>
> Este comando no genera ninguna salida.

Actividades de canalización

La canalización funcional más sencilla conecta un canal a un almacén de datos, por lo que sería una canalización con dos actividades: una actividad `channel` y una actividad `datastore`. Puede lograr el procesamiento de mensajes más potente añadiendo las actividades adicionales a la canalización.

Puede utilizar la operación RunPipelineActivity para simular los resultados de la ejecución de una actividad de canalización sobre la carga del mensaje que proporcione. Puede que le resulte útil para el desarrollo y la depuración de actividades de canalización. La sección Ejemplo de RunPipelineActivity demuestra su uso.

Actividad de canal

La primera actividad de una canalización debe ser la actividad `channel` que determina el origen de los mensajes a procesar.

```
{
    "channel": {
        "name": "MyChannelActivity",
        "channelName": "mychannel",
        "next": "MyLambdaActivity"
    }
}
```

Actividad de almacén de datos

La actividad `datastore`, que especifica dónde se almacenarán los datos procesados, es la última actividad.

```
{
    "datastore": {
        "name": "MyDatastoreActivity",
        "datastoreName": "mydatastore"
    }
}
```

Actividad AWS Lambda

Puede utilizar una actividad **`lambda`** para realizar un procesamiento complejo de los mensajes. Por ejemplo, puede enriquecer los mensajes con datos de la salida de operaciones de API externas o filtrar los mensajes según la lógica de Amazon DynamoDB. Sin embargo, no se puede utilizar esta actividad de canalización para añadir mensajes adicionales o eliminar los existentes antes de entrar en un almacén de datos.

La función de AWS Lambda utilizada en esta actividad **`lambda`** debe recibir y devolver una matriz de objetos JSON. Para ver un ejemplo, consulte the section called “Ejemplo de función de Lambda 1”.

Para conceder a AWS IoT Analytics permiso para invocar su función de Lambda, debe añadir una política. Por ejemplo, ejecute el siguiente comando de la CLI y sustituya *`exampleFunctionName`* con el nombre de la función de Lambda, sustituya *`123456789012`* con su ID de la cuenta de AWS y utilice el nombre de recurso de Amazon (ARN) de la canalización que invoca la función de Lambda en cuestión.

```
aws lambda add-permission --function-name exampleFunctionName --
action lambda:InvokeFunction --statement-id iotanalytics --principal
 iotanalytics.amazonaws.com --source-account 123456789012 --source-arn
 arn:aws:iotanalytics:us-east-1:123456789012:pipeline/examplePipeline
```

El comando devuelve lo siguiente:

```
{
    "Statement": "{\"Sid\":\"iotanalyticsa\",\"Effect\":\"Allow\",
\"Principal\":{\"Service\":\"iotanalytics.amazonaws.com\"},\"Action\":
\"lambda:InvokeFunction\",\"Resource\":\"arn:aws:lambda:aws-region:aws-
account:function:exampleFunctionName\",\"Condition\":{\"StringEquals\":
{\"AWS:SourceAccount\":\"123456789012\"},\"ArnLike\":{\"AWS:SourceArn\":
\"arn:aws:iotanalytics:us-east-1:123456789012:pipeline/examplePipeline\"}}}"
}
```

Para obtener más información, consulte Uso de políticas basadas en recursos para AWS Lambda en la Guía del desarrollador de AWS Lambda.

Ejemplo de función de Lambda 1

En este ejemplo, la función de Lambda añade información adicional en función de los datos en el mensaje original. Un dispositivo publica un mensaje con una carga similar a la del siguiente ejemplo.

```
{
  "thingid": "00001234abcd",
  "temperature": 26,
  "humidity": 29,
  "location": {
    "lat": 52.4332935,
    "lon": 13.231694
  },
  "ip": "192.168.178.54",
  "datetime": "2018-02-15T07:06:01"
}
```

Y el dispositivo tiene la siguiente definición de canalización.

```
{
    "pipeline": {
        "activities": [
            {
                "channel": {
                    "channelName": "foobar_channel",
                    "name": "foobar_channel_activity",
                    "next": "lambda_foobar_activity"
                }
            },
            {
                "lambda": {
                    "lambdaName": "MyAnalyticsLambdaFunction",
                    "batchSize": 5,
                    "name": "lambda_foobar_activity",
                    "next": "foobar_store_activity"
                }
            },
            {
                "datastore": {
                    "datastoreName": "foobar_datastore",
                    "name": "foobar_store_activity"
                }
            }
        ],
        "name": "foobar_pipeline",
        "arn": "arn:aws:iotanalytics:eu-west-1:123456789012:pipeline/foobar_pipeline"
    }
```

```
}
```

La siguiente función de Lambda en Python (`MyAnalyticsLambdaFunction`) añade la URL de GMaps y la temperatura, en Fahrenheit, al mensaje.

```
import logging
import sys

# Configure logging
logger = logging.getLogger()
logger.setLevel(logging.INFO)
streamHandler = logging.StreamHandler(stream=sys.stdout)
formatter = logging.Formatter('%(asctime)s - %(name)s - %(levelname)s - %(message)s')
streamHandler.setFormatter(formatter)
logger.addHandler(streamHandler)

def c_to_f(c):
    return 9.0/5.0 * c + 32

def lambda_handler(event, context):
    logger.info("event before processing: {}".format(event))
    maps_url = 'N/A'

    for e in event:
        #e['foo'] = 'addedByLambda'
        if 'location' in e:
            lat = e['location']['lat']
            lon = e['location']['lon']
            maps_url = "http://maps.google.com/maps?q={},{}".format(lat,lon)

        if 'temperature' in e:
            e['temperature_f'] = c_to_f(e['temperature'])

        logger.info("maps_url: {}".format(maps_url))
        e['maps_url'] = maps_url

    logger.info("event after processing: {}".format(event))

    return event
```

Ejemplo de función de Lambda 2

Una técnica útil es comprimir y serializar cargas de mensajes para reducirlos costos de transporte y almacenamiento. En este segundo ejemplo, la función de Lambda asume la carga del mensaje que representa un JSON original, que se ha comprimido y luego codificado (serializado) mediante base64 como una cadena. Devuelve el JSON original:

```
import base64
import gzip
import json
import logging
import sys

# Configure logging
logger = logging.getLogger()
logger.setLevel(logging.INFO)
streamHandler = logging.StreamHandler(stream=sys.stdout)
formatter = logging.Formatter('%(asctime)s - %(name)s - %(levelname)s - %(message)s')
streamHandler.setFormatter(formatter)
logger.addHandler(streamHandler)

def decode_to_bytes(e):
    return base64.b64decode(e)

def decompress_to_string(binary_data):
    return gzip.decompress(binary_data).decode('utf-8')

def lambda_handler(event, context):
    logger.info("event before processing: {}".format(event))

    decompressed_data = []

    for e in event:
        binary_data = decode_to_bytes(e)
        decompressed_string = decompress_to_string(binary_data)

        decompressed_data.append(json.loads(decompressed_string))

    logger.info("event after processing: {}".format(decompressed_data))

    return decompressed_data
```

Actividad AddAttributes

Una actividad `addAttributes` añade atributos que se basan en los atributos existentes en el mensaje. Esto le permite modificar la forma del mensaje antes de que se almacene. Por ejemplo, puede utilizar `addAttributes` para normalizar datos procedentes de diferentes generaciones de firmware del dispositivo.

Considere el siguiente mensaje de entrada.

```
{
    "device": {
        "id": "device-123",
        "coord": [ 47.6152543, -122.3354883 ]
    }
}
```

La actividad `addAttributes` tiene el siguiente aspecto.

```
{
    "addAttributes": {
        "name": "MyAddAttributesActivity",
        "attributes": {
            "device.id": "id",
            "device.coord[0]": "lat",
            "device.coord[1]": "lon"
        },
        "next": "MyRemoveAttributesActivity"
    }
}
```

Esta actividad mueve el ID del dispositivo al nivel raíz y extrae el valor de la matriz `coord`, promoviéndolos a atributos de nivel superior denominados `lat` y `lon`. Como resultado de esta actividad, el mensaje de entrada se transforma en el siguiente ejemplo.

```
{
    "device": {
        "id": "device-123",
        "coord": [ 47.6, -122.3 ]
    },
    "id": "device-123",
```

```
    "lat": 47.6,
    "lon": -122.3
}
```

El atributo original sigue estando presente. Si desea eliminarlo, puede utilizar la actividad `removeAttributes`.

Actividad RemoveAttributes

Una actividad `removeAttributes` elimina atributos de un mensaje. Por ejemplo, dado el mensaje resultante de la actividad `addAttributes`.

```
{
    "device": {
        "id": "device-123",
        "coord": [ 47.6, -122.3 ]
    },
    "id": "device-123",
    "lat": 47.6,
    "lon": -122.3
}
```

Para normalizar dicho mensaje de forma que incluya únicamente los datos necesarios en el nivel raíz, utilice la siguiente actividad `removeAttributes`.

```
{
    "removeAttributes": {
        "name": "MyRemoveAttributesActivity",
        "attributes": [
            "device"
        ],
        "next": "MyDatastoreActivity"
    }
}
```

Esto da lugar a que fluya el siguiente mensaje por la canalización.

```
{
    "id": "device-123",
    "lat": 47.6,
```

```
    "lon": -122.3
}
```

Actividad SelectAttributes

La actividad `selectAttributes` crea un mensaje nuevo utilizando únicamente los atributos especificados del mensaje original. El resto de atributos se descartan. `selectAttributes` crea atributos nuevos únicamente en la raíz del mensaje. Por lo tanto, dado este mensaje:

```
{
    "device": {
        "id": "device-123",
        "coord": [ 47.6152543, -122.3354883 ],
        "temp": 50,
        "hum": 40
    },
    "light": 90
}
```

y esta actividad:

```
{
    "selectAttributes": {
        "name": "MySelectAttributesActivity",
        "attributes": [
            "device.temp",
            "device.hum",
            "light"
        ],
        "next": "MyDatastoreActivity"
    }
}
```

El resultado es el siguiente mensaje avanzando por la canalización.

```
{
    "temp": 50,
    "hum": 40,
    "light": 90
}
```

Una vez más, `selectAttributes` solo puede crear objetos en el nivel raíz.

Actividad Filter

Una actividad `filter` filtra un mensaje en función de sus atributos. La expresión utilizada en esta actividad parece una cláusula `WHERE` de SQL que debe devolver un valor booleano.

```
{
    "filter": {
        "name": "MyFilterActivity",
        "filter": "temp > 40 AND hum < 20",
        "next": "MyDatastoreActivity"
    }
}
```

Actividad DeviceRegistryEnrich

La actividad `deviceRegistryEnrich` le permite añadir datos desde el registro de dispositivos de AWS IoT a la carga del mensaje. Por ejemplo, si se recibe el mensaje siguiente:

```
{
    "temp": 50,
    "hum": 40,
    "device" {
        "thingName": "my-thing"
    }
}
```

y una actividad `deviceRegistryEnrich` que tiene un aspecto similar a este:

```
{
    "deviceRegistryEnrich": {
        "name": "MyDeviceRegistryEnrichActivity",
        "attribute": "metadata",
        "thingName": "device.thingName",
        "roleArn": "arn:aws:iam::<your-account-number>:role:MyEnrichRole",
        "next": "MyDatastoreActivity"
    }
}
```

El mensaje de salida se parece ahora a este ejemplo.

```
{
    "temp" : 50,
    "hum" : 40,
    "device" {
        "thingName" : "my-thing"
    },
    "metadata" : {
        "defaultClientId": "my-thing",
        "thingTypeName": "my-thing",
        "thingArn": "arn:aws:iot:us-east-1:<your-account-number>:thing/my-thing",
        "version": 1,
        "thingName": "my-thing",
        "attributes": {},
        "thingId": "aaabbbccc-dddeeef-gghh-jjkk-llmmnnoopp"
    }
}
```

Debe especificar un rol en el campo `roleArn` de la definición de actividad que tenga asociados los permisos pertinentes. El rol debe tener una política de permisos parecida a la del siguiente ejemplo.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:DescribeThing"
            ],
            "Resource": [
                "arn:aws:iot:<region>:<account-id>:thing/<thing-name>"
            ]
        }
    ]
}
```

y una política de confianza parecida a esta:

```
{
    "Version": "2012-10-17",
    "Statement": [
```

```
        {
            "Sid": "",
            "Effect": "Allow",
            "Principal": {
                "Service": "iotanalytics.amazonaws.com"
            },
            "Action": [
                "sts:AssumeRole"
            ]
        }
    ]
}
```

Actividad DeviceShadowEnrich

Una actividad `deviceShadowEnrich` añade información del servicio sombras de dispositivo de AWS IoT a un mensaje. Por ejemplo, dado el mensaje:

```
{
    "temp": 50,
    "hum": 40,
    "device": { "thingName": "my-thing" }
}
```

y la actividad `deviceShadowEnrich` siguiente:

```
{
    "deviceShadowEnrich": {
        "name": "MyDeviceShadowEnrichActivity",
        "attribute": "shadow",
        "thingName": "device.thingName",
        "roleArn": "arn:aws:iam::<your-account-number>:role:MyEnrichRole",
        "next": "MyDatastoreActivity"
    }
}
```

El resultado es un mensaje con el aspecto del siguiente ejemplo.

```
{
    "temp": 50,
    "hum": 40,
```

```
    "device": {
        "thingName": "my-thing"
    },
    "shadow": {
        "state": {
            "desired": {
                "attributeX": valueX, ...
            },
            "reported": {
                "attributeX": valueX, ...
            },
            "delta": {
                "attributeX": valueX, ...
            }
        },
        "metadata": {
            "desired": {
                "attribute1": {
                    "timestamp": timestamp
                }, ...
            },
            "reported": ": {
                "attribute1": {
                    "timestamp": timestamp
                }, ...
            }
        },
        "timestamp": timestamp,
        "clientToken": "token",
        "version": version
    }
}
```

Debe especificar un rol en el campo `roleArn` de la definición de actividad que tenga asociados los permisos pertinentes. El rol debe tener una política de permisos parecida a la siguiente.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:GetThingShadow"
```

```
            ],
            "Resource": [
                "arn:aws:iot:<region>:<account-id>:thing/<thing-name>"
            ]
        }
    ]
}
```

y una política de confianza parecida a esta:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "",
            "Effect": "Allow",
            "Principal": {
                "Service": "iotanalytics.amazonaws.com"
            },
            "Action": [
                "sts:AssumeRole"
            ]
        }
    ]
}
```

Actividad math

Una actividad `math` calcula una expresión aritmética utilizando los atributos del mensaje. La expresión debe devolver un número. Por ejemplo, si se recibe el siguiente mensaje de entrada:

```
{
    "tempF": 50,
}
```

después de procesarlo mediante la actividad `math` siguiente:

```
{
    "math": {
        "name": "MyMathActivity",
        "math": "(tempF - 32) / 2",
```

```
        "attribute": "tempC",
        "next": "MyDatastoreActivity"
    }
}
```

el mensaje resultante tiene este aspecto:

```
{
    "tempF" : 50,
    "tempC": 9
}
```

Operadores y funciones de la actividad math

Puede utilizar los siguientes operadores en una actividad `math`:

+	suma
-	resta
*	multiplicación
/	división
%	módulo

Puede utilizar las siguientes funciones en una actividad `math`:

- abs(Decimal)
- acos(Decimal)
- asin(Decimal)
- atan(Decimal)
- atan2(Decimal, Decimal)
- ceil(Decimal)
- cos(Decimal)
- cosh(Decimal)

- exp(Decimal)
- ln(Decimal)
- log(Decimal)
- mod(Decimal, Decimal)
- power(Decimal, Decimal)
- round(Decimal)
- sign(Decimal)
- sin(Decimal)
- sinh(Decimal)
- sqrt(Decimal)
- tan(Decimal)
- tanh(Decimal)
- trunc(Decimal, Entero)

abs(Decimal)

Devuelve el valor absoluto de un número.

Ejemplos: `abs(-5)` devuelve 5.

Tipo de argumento	Resultado
`Int`	`Int`, el valor absoluto del argumento.
`Decimal`	`Decimal`, el valor absoluto del argumento.
`Boolean`	`Undefined` .
`String`	`Decimal`. El resultado es el valor absoluto del argumento. Si la cadena no se puede convertir, el resultado es `Undefined` .
Array (Matriz)	`Undefined` .
Objeto	`Undefined` .

Tipo de argumento	Resultado
Null	`Undefined` .
Sin definir	`Undefined` .

acos(Decimal)

Devuelve el coseno inverso de un número en radianes. Los argumentos `Decimal` se redondean con doble precisión antes de la aplicación de la función.

Ejemplos: `acos(0)` = 1.5707963267948966

Tipo de argumento	Resultado
`Int`	`Decimal` (con doble precisión), el coseno inverso del argumento. Se devuelven resultados imaginarios como `Undefined` .
`Decimal`	`Decimal` (con doble precisión), el coseno inverso del argumento. Se devuelven resultados imaginarios como `Undefined` .
`Boolean`	`Undefined` .
`String`	`Decimal` (con doble precisión), el coseno inverso del argumento. Si la cadena no se puede convertir, el resultado es `Undefined` . Se devuelven resultados imaginarios como `Undefined` .
Array (Matriz)	`Undefined` .
Objeto	`Undefined` .
Null	`Undefined` .
Sin definir	`Undefined` .

asin(Decimal)

Devuelve el seno inverso de un número en radianes. Los argumentos `Decimal` se redondean con doble precisión antes de la aplicación de la función.

Ejemplos: `asin(0)` = 0.0

Tipo de argumento	Resultado
`Int`	`Decimal` (con doble precisión), el seno inverso del argumento. Se devuelven resultados imaginarios como `Undefined` .
`Decimal`	`Decimal` (con doble precisión), el seno inverso del argumento. Se devuelven resultados imaginarios como `Undefined` .
`Boolean`	`Undefined` .
`String`	`Decimal` (con doble precisión), el seno inverso del argumento. Si la cadena no se puede convertir, el resultado es `Undefined` . Se devuelven resultados imaginarios como `Undefined` .
Array (Matriz)	`Undefined` .
Objeto	`Undefined` .
Null	`Undefined` .
Sin definir	`Undefined` .

atan(Decimal)

Devuelve la tangente inversa de un número en radianes. Los argumentos `Decimal` se redondean con doble precisión antes de la aplicación de la función.

Ejemplos: `atan(0)` = 0.0

Tipo de argumento	Resultado
`Int`	`Decimal` (con doble precisión), la tangente inversa del argumento. Se devuelven resultados imaginarios como `Undefined` .
`Decimal`	`Decimal` (con doble precisión), la tangente inversa del argumento. Se devuelven resultados imaginarios como `Undefined` .
`Boolean`	`Undefined` .
`String`	`Decimal` (con doble precisión), la tangente inversa del argumento. Si la cadena no se puede convertir, el resultado es `Undefined` . Se devuelven resultados imaginarios como `Undefined` .
Array (Matriz)	`Undefined` .
Objeto	`Undefined` .
Null	`Undefined` .
Sin definir	`Undefined` .

atan2(Decimal, Decimal)

Devuelve el ángulo, en radianes, entre el eje x positivo y el punto (x, y) definido en los dos argumentos. El ángulo es positivo para ángulos en sentido antihorario (semiplano superior y > 0), y negativo para ángulos en sentido horario. Los argumentos `Decimal` se redondean a doble precisión antes de la aplicación de la función.

Ejemplos: `atan(1, 0)` = 1.5707963267948966

Tipo de argumento	Tipo de argumento	Resultado
`Int` / `Decimal`	`Int` / `Decimal`	`Decimal` (con doble precisión), el ángulo entre el eje x y el punto (x, y) especificado.
`Int` / `Decimal` / `String`	`Int` / `Decimal` / `String`	`Decimal`, la tangente inversa del punto descrito. Si una cadena no se puede convertir, el resultado es `Undefined` .
Otro valor	Otro valor	`Undefined` .

ceil(Decimal)

Redondea el valor `Decimal` indicado al valor `Int` superior más cercano.

Ejemplos:

`ceil(1.2)` = 2

`ceil(11.2)` = -1

Tipo de argumento	Resultado
`Int`	`Int`, el valor del argumento.
`Decimal`	`Int`, la cadena se convierte en `Decimal` y se redondea al `Int` superior más cercano. Si la cadena no se puede convertir en un valor `Decimal`, el resultado es `Undefined` .
Otro valor	`Undefined` .

cos(Decimal)

Devuelve el coseno de un número en radianes. Los argumentos `Decimal` se redondean con doble precisión antes de la aplicación de la función.

Ejemplos: `cos(0)` = 1

Tipo de argumento	Resultado
`Int`	`Decimal` (con doble precisión), el coseno del argumento. Se devuelven resultados imaginarios como `Undefined` .
`Decimal`	`Decimal` (con doble precisión), el coseno del argumento. Se devuelven resultados imaginarios como `Undefined` .
`Boolean`	`Undefined` .
`String`	`Decimal` (con doble precisión), el coseno del argumento. Si la cadena no se puede convertir en un valor `Decimal`, el resultado es `Undefined` . Se devuelven resultados imaginarios como `Undefined` .
Array (Matriz)	`Undefined` .
Objeto	`Undefined` .
Null	`Undefined` .
Sin definir	`Undefined` .

cosh(Decimal)

Devuelve el coseno hiperbólico de un número en radianes. Los argumentos `Decimal` se redondean con doble precisión antes de la aplicación de la función.

Ejemplos: `cosh(2.3)` = 5.037220649268761

Tipo de argumento	Resultado
`Int`	`Decimal` (con doble precisión), el coseno hiperbólico del argumento. Se devuelven resultados imaginarios como `Undefined` .
`Decimal`	`Decimal` (con doble precisión), el coseno hiperbólico del argumento. Se devuelven resultados imaginarios como `Undefined` .
`Boolean`	`Undefined` .
`String`	`Decimal` (con doble precisión), el coseno hiperbólico del argumento. Si la cadena no se puede convertir en un valor `Decimal`, el resultado es `Undefined` . Se devuelven resultados imaginarios como `Undefined` .
Array (Matriz)	`Undefined` .
Objeto	`Undefined` .
Null	`Undefined` .
Sin definir	`Undefined` .

exp(Decimal)

Devuelve e elevado al argumento decimal. Los argumentos `Decimal` se redondean a doble precisión antes de la aplicación de la función.

Ejemplos: `exp(1)` = 1

Tipo de argumento	Resultado
`Int`	`Decimal` (con doble precisión), argumento potencia e.

Tipo de argumento	Resultado
`Decimal`	`Decimal` (con doble precisión), argumento potencia e
`String`	`Decimal` (con doble precisión), argumento potencia e. Si el `String` no se puede convertir en un valor `Decimal`, el resultado es `Undefined` .
Otro valor	`Undefined` .

ln(Decimal)

Devuelve el logaritmo natural del argumento. Los argumentos `Decimal` se redondean con doble precisión antes de la aplicación de la función.

Ejemplos: `ln(e)` = 1

Tipo de argumento	Resultado
`Int`	`Decimal` (con doble precisión), el log natural del argumento.
`Decimal`	`Decimal` (con doble precisión), el log natural del argumento.
`Boolean`	`Undefined` .
`String`	`Decimal` (con doble precisión), el log natural del argumento. Si la cadena no se puede convertir en un valor `Decimal`, el resultado es `Undefined` .
Array (Matriz)	`Undefined` .
Objeto	`Undefined` .
Null	`Undefined` .

Tipo de argumento	Resultado
Sin definir	`Undefined` .

log(Decimal)

Devuelve el logaritmo decimal del argumento. Los argumentos `Decimal` se redondean con doble precisión antes de la aplicación de la función.

Ejemplos: `log(100)` = 2.0

Tipo de argumento	Resultado
`Int`	`Decimal` (con doble precisión), el log de base 10 del argumento.
`Decimal`	`Decimal` (con doble precisión), el log de base 10 del argumento.
`Boolean`	`Undefined` .
`String`	`Decimal` (con doble precisión), el log de base 10 del argumento. Si el valor `String` no se puede convertir en un valor `Decimal`, el resultado es `Undefined` .
Array (Matriz)	`Undefined` .
Objeto	`Undefined` .
Null	`Undefined` .
Sin definir	`Undefined` .

mod(Decimal, Decimal)

Devuelve el resto de la división del primer argumento del segundo argumento. También puede utilizar `%` como un operador infijo para la misma funcionalidad de módulo.

Ejemplos: `mod(8, 3)` = 3

Operando izquierdo	Operando derecho	Output
`Int`	`Int`	`Int`, el módulo del primer argumento del segundo argumento.
`Int` / `Decimal`	`Int` / `Decimal`	`Decimal`, el módulo del primer argumento del segundo argumento.
`String` / `Int` / `Decimal`	`String` / `Int` / `Decimal`	Si todas las cadenas se convierten en `Decimals`, el resultado es el residuo de dividir el primer argumento entre el segundo argumento. De lo contrario, `Undefined` .
Otro valor	Otro valor	`Undefined` .

power(Decimal, Decimal)

Devuelve el primer argumento elevado al segundo argumento. Los argumentos `Decimal` se redondean con doble precisión antes de la aplicación de la función.

Examples: `power(2, 5)` = 32.0

Tipo de argumento 1	Tipo de argumento 2	Output
`Int` / `Decimal`	`Int` / `Decimal`	Un `Decimal` (con doble precisión), el primer argumento elevado a la potencia del segundo argumento.
`Int` / `Decimal` / `String`	`Int` / `Decimal` / `String`	Un `Decimal` (con doble precisión), el primer

Tipo de argumento 1	Tipo de argumento 2	Output
		argumento elevado a la potencia del segundo argumento. Cualquier cadena se convierte `Decimals`. Si cualquier `String` no se convierte en un valor `Decimal`, el resultado es `Undefined` .
Otro valor	Otro valor	`Undefined` .

round(Decimal)

Redondea el valor `Decimal` indicado al valor `Int` más cercano. Si el valor `Decimal` está a la misma distancia de dos valores `Int`, (p. ej., 0.5), el valor `Decimal` se redondea al valor superior.

Ejemplos:

`Round(1.2)` = 1

`Round(1.5)` = 2

`Round(1.7)` = 2

`Round(-1.1)` = -1

`Round(-1.5)` = -2

Tipo de argumento	Resultado
`Int`	El argumento
`Decimal`	El valor `Decimal` se redondea al valor `Int` inferior más cercano.
`String`	El valor `Decimal` se redondea al valor `Int` inferior más cercano. Si la cadena no se puede

Tipo de argumento	Resultado
	convertir en un valor `Decimal`, el resultado es `Undefined` .
Otro valor	`Undefined` .

sign(Decimal)

Devuelve el signo del número especificado. Cuando el signo del argumento es positivo, se devuelve 1. Cuando el signo del argumento es negativo, se devuelve -1. Si el argumento es 0, se devuelve 0.

Ejemplos:

`sign(-7)` = -1

`sign(0)` = 0

`sign(13)` = 1

Tipo de argumento	Resultado
`Int`	`Int`, el signo del valor `Int`.
`Decimal`	`Int`, el signo del valor `Decimal`.
`String`	`Int`, el signo del valor `Decimal`. La cadena se convierte en un valor `Decimal` y se devuelve el signo del valor `Decimal`. Si el valor `String` no se puede convertir en un valor `Decimal`, el resultado es `Undefined` .
Otro valor	`Undefined` .

sin(Decimal)

Devuelve el seno de un número en radianes. Los argumentos `Decimal` se redondean con doble precisión antes de la aplicación de la función.

Ejemplos: `sin(0)` = 0.0

Tipo de argumento	Resultado
`Int`	`Decimal` (con doble precisión), el seno del argumento.
`Decimal`	`Decimal` (con doble precisión), el seno del argumento.
`Boolean`	`Undefined` .
`String`	`Decimal`, el seno del argumento. Si la cadena no se puede convertir en un valor `Decimal`, el resultado es `Undefined` .
`Array`	`Undefined` .
`Object`	`Undefined` .
`Null`	`Undefined` .
`Undefined`	`Undefined` .

sinh(Decimal)

Devuelve el seno hiperbólico de un número. Los valores `Decimal` se redondean con doble precisión antes de la aplicación de la función. El resultado es un valor `Decimal` de doble precisión.

Ejemplos: `sinh(2.3)` = 4.936961805545957

Tipo de argumento	Resultado
`Int`	`Decimal` (con doble precisión); el seno hiperbólico del argumento.
`Decimal`	`Decimal` (con doble precisión); el seno hiperbólico del argumento.
`Boolean`	`Undefined` .

Tipo de argumento	Resultado
`String`	`Decimal`, el seno hiperbólico del argumento. Si la cadena no se puede convertir en un valor `Decimal`, el resultado es `Undefined` .
`Array`	`Undefined` .
`Object`	`Undefined` .
`Null`	`Undefined` .
`Undefined`	`Undefined` .

sqrt(Decimal)

Devuelve la raíz cuadrada de un número. Los argumentos `Decimal` se redondean con doble precisión antes de la aplicación de la función.

Ejemplos: `sqrt(9)` = 3.0

Tipo de argumento	Resultado
`Int`	La raíz cuadrada del argumento.
`Decimal`	La raíz cuadrada del argumento.
`Boolean`	`Undefined` .
`String`	La raíz cuadrada del argumento. Si la cadena no se puede convertir en un valor `Decimal`, el resultado es `Undefined` .
`Array`	`Undefined` .
`Object`	`Undefined` .
`Null`	`Undefined` .
`Undefined`	`Undefined` .

tan(Decimal)

Devuelve la tangente de un número en radianes. Los valores `Decimal` se redondean con doble precisión antes de la aplicación de la función.

Ejemplos: `tan(3)` = -0.1425465430742778

Tipo de argumento	Resultado
`Int`	`Decimal` (con doble precisión), la tangente del argumento.
`Decimal`	`Decimal` (con doble precisión), la tangente del argumento.
`Boolean`	`Undefined` .
`String`	`Decimal` (con doble precisión), la tangente del argumento. Si la cadena no se puede convertir en un valor `Decimal`, el resultado es `Undefined` .
`Array`	`Undefined` .
`Object`	`Undefined` .
`Null`	`Undefined` .
`Undefined`	`Undefined` .

tanh(Decimal)

Devuelve la tangente hiperbólica de un número en radianes. Los valores `Decimal` se redondean con doble precisión antes de la aplicación de la función.

Ejemplos: `tanh(2.3)` = 0.9800963962661914

Tipo de argumento	Resultado
`Int`	`Decimal` (con doble precisión), la tangente hiperbólica del argumento.
`Decimal`	`Decimal` (con doble precisión), la tangente hiperbólica del argumento.
`Boolean`	`Undefined` .
`String`	`Decimal` (con doble precisión), la tangente hiperbólica del argumento. Si la cadena no se puede convertir en un valor `Decimal`, el resultado es `Undefined` .
`Array`	`Undefined` .
`Object`	`Undefined` .
`Null`	`Undefined` .
`Undefined`	`Undefined` .

trunc(Decimal, Entero)

Trunca el primer argumento según el número del valor `Decimal` especificado por el segundo argumento. Si el segundo argumento es inferior a cero, se establecerá en cero. Si el segundo argumento es superior a 34, se establecerá en 34. Los ceros finales se eliminan del resultado.

Ejemplos:

`trunc(2.3, 0)` = 2

`trunc(2.3123, 2)` = 2.31

`trunc(2.888, 2)` = 2.88

`trunc(2.00, 5)` = 2

Tipo de argumento 1	Tipo de argumento 2	Resultado
`Int`	`Int`	El valor de origen.
`Int` / `Decimal` / `String`	`Int` / `Decimal`	El primer argumento se trunca en la longitud indicada por el segundo argumento. El segundo argumento, si no es un valor `Int`, se redondea al valor `Int` inferior más cercano. Las cadenas se convierten en valores de tipo `Decimal`. Si se produce un error en la conversión de cadena, el resultado obtenido es `Undefined` .
Otro valor		Sin definir.

RunPipelineActivity

A continuación, se muestra un ejemplo de cómo se utilizará el comando `RunPipelineActivity` para probar una actividad de canalización. Para este ejemplo, probamos una actividad math.

1. Cree un archivo `maths.json` que contenga la definición de la actividad de canalización que desea probar.

```
{
    "math": {
        "name": "MyMathActivity",
        "math": "((temp - 32) * 5.0) / 9.0",
        "attribute": "tempC"
    }
}
```

2. Cree un archivo `payloads.json`, que contenga las cargas de ejemplo que se utilizan para probar la actividad de canalización.

```
[
    "{\"humidity\": 52, \"temp\": 68 }",
    "{\"humidity\": 52, \"temp\": 32 }"
]
```

3. Llame a la operación `RunPipelineActivities` desde la línea de comandos.

```
aws iotanalytics run-pipeline-activity  --pipeline-activity file://maths.json  --
payloads file://payloads.json --cli-binary-format raw-in-base64-out
```

Esto produce los siguientes resultados:

```
{
    "logResult": "",
    "payloads": [
        "eyJodW1pZGl0eSI6NTIsInRlbXAiOjY4LCJ0ZW1wQyI6MjB9",
        "eyJodW1pZGl0eSI6NTIsInRlbXAiOjMyLCJ0ZW1wQyI6MH0="
    ]
}
```

Las cargas que aparecen en los resultados son cadenas codificadas en Base64. Cuando estas cadenas se decodifican, se obtienen los siguientes resultados.

```
{"humidity":52,"temp":68,"tempC":20}
{"humidity":52,"temp":32,"tempC":0}
```

Reprocesamiento de los mensajes de canal

AWS IoT Analytics le permite volver a procesar los datos de canal. Puede ser útil en los siguientes casos:

- Desea reproducir datos adquiridos existentes en lugar de volver a comenzar.
- Realiza una actualización en una canalización y desea poner al día los datos con respecto a los cambios.
- Desea incluir los datos que se ingirieron antes de realizar cambios en las opciones de almacenamiento administrado por el cliente, los permisos de los canales o el almacén de datos.

Parámetros

Cuando vuelva a procesar los mensajes de canal a través de la canalización con AWS IoT Analytics, debe especificar la siguiente información:

`StartPipelineReprocessing`

Inicia el reprocesamiento de los mensajes del canal a través de la canalización.

`ChannelMessages`

Especifique uno o más conjuntos de mensajes de canal que desea volver a procesar

Si usa el objeto `channelMessages`, no debe especificar un valor para `startTime` y`endTime`.

`s3Paths`

Especifique una o más claves que identifican los objetos de Amazon Simple Storage Service (Amazon S3) que guardan sus mensajes de canal. Debe utilizar la ruta completa de la clave.

Ruta de ejemplo: `00:00:00/1582940490000_1582940520000_123456789012_mychannel_0_2118.0.jso`

Tipo: matriz de cadenas

Restricciones de los miembros de la matriz: de 1 a 100 elementos.

Restricciones de longitud: de 1 a 1024 caracteres.

`endTime`

La hora de finalización (exclusiva) de los datos del canal que se vuelve a procesar.

Si especifica un valor para el parámetro `endTime`, no debe utilizar el objeto `channelMessages`.

Tipo: Timestamp

`startTime`

La hora de inicio (inclusive) de los datos de mensajes sin procesar que se vuelven a procesar.

Si especifica un valor para el parámetro `startTime`, no debe utilizar el objeto `channelMessages`.

Tipo: Timestamp

`pipelineName`

El nombre de la canalización en la que se desea iniciar el reprocesamiento.

Tipo: String

Restricciones de longitud: de 1 a 128 caracteres.

Reprocesamiento de los mensajes de canal (consola)

En este tutorial, se explica cómo volver a procesar los datos del canal que están almacenados en el objeto de Amazon S3 especificado en la consola de AWS IoT Analytics.

Antes de comenzar, asegúrese de que los mensajes de canal que desea volver a procesar se guarden en un bucket de Amazon S3 administrado por el cliente.

1. Inicie sesión en la consola de AWS IoT Analytics.
2. En el panel de navegación, seleccione Canalizaciones.
3. Seleccione su canalización de destino.
4. Seleccione Volver a procesar mensajes desde Acciones.
5. En la página Reprocesamiento de la canalización, seleccione Objetos S3 en Volver a procesar mensajes.

La consola AWS IoT Analytics también proporciona las siguientes opciones:

- Todo el rango disponible: vuelva a procesar todos los datos válidos del canal.
- Últimos 120 días: vuelva a procesar los datos que llegaron en los últimos 120 días.
- Últimos 90 días: vuelva a procesar los datos que llegaron en los últimos 90 días.
- Últimos 30 días: vuelva a procesar los datos que llegaron en los últimos 30 días.
- Intervalo personalizado: vuelva a procesar los datos que llegaron en el intervalo de tiempo especificado. Puede elegir cualquier intervalo de tiempo.

6. Introduzca la clave del objeto Amazon S3 que almacena los mensajes de su canal.

   Para buscar la clave, haga lo siguiente:

   a. Uso de la consola de Amazon S3.
   b. Seleccione el objeto de Amazon S3 de destino.
   c. En Propiedades, en la sección Descripción general del objeto, copie la clave.

7. Seleccione Comenzar reprocesamiento.

Reprocesamiento de los mensajes de canal (API)

Cuando utilice la API `StartPipelineReprocessing`, tenga en cuenta lo siguiente:

- Los parámetros `startTime` y `endTime` especifican cuándo se adquirieron los datos sin procesar, pero se trata de estimaciones aproximadas. Puede redondearlos a la hora más cercana. El parámetro `startTime` es inclusivo, pero el parámetro `endTime` es exclusivo.
- El comando lanza el reprocesamiento de forma asíncrona y vuelve inmediatamente.
- No se garantiza que los mensajes reprocesados se procesen en el orden en que se recibieron originalmente. Es más o menos el mismo, pero no exacto.
- Puede realizar hasta 1000 solicitudes de la API `StartPipelineReprocessing` cada 24 horas para volver a procesar los mensajes del mismo canal a través de una canalización.
- El reprocesamiento de los datos sin procesar genera costos adicionales.

Para obtener más información, consulte la API StartPipelineReprocessing en la Referencia de la API de AWS IoT Analytics.

Cancelación de las actividades de reprocesamiento de canales

Para cancelar una actividad de reprocesamiento de canalizaciones, use la API CancelPipelineReprocessing o seleccione Cancelar el reprocesamiento en la página Actividades de la consola de AWS IoT Analytics. Si cancela el reprocesamiento, los datos restantes no se volverán a procesar. Debe iniciar otra solicitud de reprocesamiento.

Utilice la API DescribePipeline para comprobar el estado del reprocesamiento. Consulte el campo `reprocessingSummaries` de la respuesta.

Automatización del flujo de trabajo

AWS IoT Analytics proporciona un análisis de datos avanzado para AWS IoT. Puede recopilar automáticamente datos de IoT, procesarlos, almacenarlos y analizarlos con las herramientas de aprendizaje automático y de análisis de datos. Puede ejecutar contenedores que alojen su propio código de análisis personalizado o cuaderno de Jupyter, o utilizar contenedores de código personalizado de terceros para no tener que volver a crear las herramientas de análisis existentes. Puede usar las siguientes capacidades para tomar datos de entrada de un almacén de datos e incluirlos en un flujo de trabajo automatizado:

Crear contenido del conjunto de datos de forma periódica.

- Programar la creación automática del contenido del conjunto de datos especificando un desencadenador al llamar a `CreateDataset` (`triggers:schedule:expression`). Los datos que tiene en un almacén de datos se utilizan para crear el contenido del conjunto de datos. Seleccione los campos que desee mediante una consulta SQL (`actions:queryAction:sqlQuery`).

- Defina un intervalo de tiempo continuo y no solapado que garantice que el contenido nuevo del conjunto de datos solo incluya los datos que hayan llegado desde la última vez. Utilice los campos `actions:queryAction:filters:deltaTime` y `:offsetSeconds` para especificar el intervalo de tiempo delta. A continuación, especifique un desencadenador para crear el contenedor del conjunto de datos cuando transcurra el intervalo de tiempo. Consulte the section called “Ejemplo 6. Creación de un conjunto de datos SQL con una ventana diferencial (CLI)”.

Crear el contenido del conjunto de datos tras la finalización de otro conjunto de datos

- Active la creación de nuevo contenido del conjunto de datos una vez completada la creación de contenido del otro conjunto datos `triggers:dataset:name`.

Ejecutar automáticamente las aplicaciones de análisis.

- Cree contenedores para sus propias aplicaciones de análisis de datos personalizadas y actívelos para que se ejecuten cuando se cree contenido de otro conjunto de datos. De esta forma, puede incluir en la aplicación datos desde el contenido de un conjunto de datos que se crea de forma periódica. Puede tomar medidas automáticamente en los resultados de su análisis desde la aplicación. (`actions:containerAction`)

Crear el contenido del conjunto de datos tras la finalización de otro conjunto de datos

Active la creación de nuevo contenido del conjunto de datos una vez completada la creación de contenido del otro conjunto datos `triggers:dataset:name`.

Ejecutar automáticamente las aplicaciones de análisis.

Cree contenedores para sus propias aplicaciones de análisis de datos personalizadas y actívelos para que se ejecuten cuando se cree contenido de otro conjunto de datos. De esta forma, puede incluir en la aplicación datos desde el contenido de un conjunto de datos que se crea de forma periódica. Puede tomar medidas automáticamente en los resultados de su análisis desde la aplicación. (`actions:containerAction`)

Casos de uso

Automatizar la medición de la calidad de los productos para reducir los gastos operativos

Dispone de un sistema con una válvula inteligente que mide la temperatura, la humedad y la presión. El sistema recopila eventos periódicamente y también cuando se producen determinados eventos, como, por ejemplo, cuando se abre y cierra una válvula. Con AWS IoT Analytics, puede automatizar un análisis que agrega los datos no solapados de esas ventanas periódicas y crea informes KPI sobre la calidad del producto final. Después de procesar cada lote, mida la calidad general del producto y reduzca los gastos de explotación maximizando el volumen de producción.

Automatizar el análisis de una flota de dispositivos

Realiza análisis (algoritmos, ciencia de datos o ML para los KPI) cada 15 minutos con datos generados por cientos de dispositivos. Con cada ciclo de análisis, se genera y almacena el estado para la siguiente ejecución de análisis. Para cada uno de los análisis, únicamente se desea utilizar los datos recibidos en un periodo de tiempo determinado. Con AWS IoT Analytics puede organizar el análisis y crear el KPI y el informe para cada ejecución y, a continuación, almacenar los datos para futuros análisis.

Automatizar la detección de anomalías

AWS IoT Analytics le permite automatizar el flujo de trabajo de detección de anomalías que se debe ejecutar manualmente cada 15 minutos en los datos nuevos que han llegado a un almacén de datos. También le permite automatizar un panel que muestre el uso de los dispositivos y los usuarios más activos en un periodo de tiempo determinado.

Predecir los resultados de procesos industriales

Imagine que tiene líneas de producción industrial. Mediante los datos enviados a AWS IoT Analytics, incluidas las mediciones disponibles de los procesos, puede instrumentar los flujos de trabajo analíticos para predecir los resultados de los procesos. Los datos del modelo se pueden organizar en una matriz M x N en la que cada fila contiene datos de varios momentos en los que se toman muestras de laboratorio. AWS IoT Analytics le ayuda a poner en funcionamiento su flujo de trabajo analítico mediante la creación de ventanas diferenciales y el uso de sus herramientas de ciencia de datos para crear KPI y guardar el estado de los dispositivos de medición.

Uso de un contenedor de Docker.

Esta sección incluye información sobre cómo construir su propio contenedor de Docker. Existe un riesgo para la seguridad si reutiliza contenedores de Docker creados por terceros: estos contenedores pueden ejecutar un código arbitrario con sus permisos de usuario. Asegúrese de que el autor del contenedor de terceros es de confianza antes de utilizarlo.

A continuación, se muestran los pasos que debe realizar para configurar análisis de datos periódicos con los datos que han llegado desde que se llevó a cabo el último análisis:

1. Cree un contenedor de Docker que contenga los datos de la aplicación además de las bibliotecas necesarias u otras dependencias.

   La extensión Jupyter de IotAnalytics proporciona una API de creación de contenedores que ayuda en el proceso de creación de contenedores. También puede ejecutar imágenes de su propia creación en las que puede crear o ensamblar el conjunto de herramientas de la aplicación para realizar el análisis de datos o el cálculo que desee. AWS IoT Analytics le permite definir el origen de los datos de entrada a la aplicación contenerizada y el destino de los datos de salida del contenedor de Docker mediante variables. (La sección Variables de entrada/salida del contenedor de Docker personalizado contiene más información sobre el uso de variables con un contenedor personalizado.)
2. Cargue el contenedor en un registro de Amazon ECR.
3. Cree un almacén de datos para recibir y almacenar mensajes (datos) de dispositivos (`iotanalytics: CreateDatastore`)
4. Cree un canal donde se envíen los mensajes (`iotanalytics: CreateChannel`).
5. Cree una canalización para conectar el canal con el almacén de datos (`iotanalytics: CreatePipeline`).

6. Debe crear un rol de IAM que conceda permiso para enviar datos de mensajes a un canal de AWS IoT Analytics (`iam: CreateRole`.)
7. Cree una regla de IoT que utilice una consulta SQL para conectar un canal con el origen de los datos del mensaje (`iot: CreateTopicRule` campo `topicRulePayload:actions:iotAnalytics`). Cuando un dispositivo envía un mensaje con el tema apropiado a través de MQTT, se dirigirá al canal. O bien, puede utilizar `iotanalytics: BatchPutMessage` para enviar mensajes directamente a un canal desde un dispositivo capaz de utilizar el SDK de AWS o la AWS CLI.
8. Cree un conjunto de datos SQL cuya creación se desencadene mediante un cronograma (`iotanalytics: CreateDataset,` campo`actions: queryAction:sqlQuery`).

   Asimismo, debe especificar un filtro previo para aplicárserlo a los datos del mensaje con objeto de limitar los mensajes a los que han llegado desde la última ejecución de la acción. (El campo `actions:queryAction:filters:deltaTime:timeExpression` proporciona una expresión que permite determinar la hora de un mensaje, mientras que el campo `actions:queryAction:filters:deltaTime:offsetSeconds` especifica la posible latencia en la llegada del mensaje.)

   El filtro previo, junto con la programación del desencadenador, determina la ventana diferencial. Cada conjunto de datos SQL nuevo se creará utilizando los mensajes recibidos desde la última vez que se creó el conjunto de datos SQL. (¿Qué pasa con la primera vez que se crea el conjunto de datos SQL? Se hace una estimación de cuándo se habría creado por última vez el conjunto de datos basándose en el calendario y el prefiltro).
9. Cree otro conjunto de datos que se active al crear el primero (CreateDataset campo `trigger:dataset`). Para este conjunto de datos, especifique una acción de contenedor (campo `actions:containerAction`) que apunte al contenedor de Docker creado en el primer paso y proporcione la información necesaria para ejecutarlo. Especifique también:

   - El ARN del contenedor de Docker almacenado en su cuenta (`image`).
   - El ARN del rol que concede permiso al sistema para obtener acceso a los recursos necesarios con el fin de ejecutar la acción de contenedor (`executionRoleArn`).
   - La configuración del recurso que ejecuta la acción de contenedor (`resourceConfiguration`).
   - El tipo de recurso informático utilizado para ejecutar la acción del contenedor (`computeType` con valores posibles: `ACU_1 [vCPU=4, memory=16GiB] or ACU_2 [vCPU=8, memory=32GiB]`).

- El tamaño (GB) del almacenamiento persistente disponible para la instancia del recurso utilizada para ejecutar la acción de contenedor (`volumeSizeInGB`).
- Los valores de las variables utilizadas en el contexto de la ejecución de la aplicación (básicamente, los parámetros que se pasan a la aplicación) (`variables`).

  Estas variables se sustituirán en el momento en el que se ejecute un contenedor. Esto le permite ejecutar el mismo contenedor con diferentes variables (parámetros), que se proporcionan en el momento en que se crea el contenido del conjunto de datos. La extensión Jupyter de IotAnalytics simplifica este proceso reconociendo automáticamente las variables de un bloc de notas y haciendo que estén disponibles como parte del proceso de creación de contenedores. Puede elegir las variables conocidas o añadir sus propias variables personalizadas. Antes de ejecutar un contenedor, el sistema sustituye cada una de estas variables por el valor actual en el momento de la ejecución.
- Una de las variables es el nombre del conjunto de datos cuyo contenido más reciente se utilizará como entrada para la aplicación (este será el nombre del conjunto de datos que ha creado en el paso anterior) (`datasetContentVersionValue:datasetName`).

Con la consulta SQL y la ventana diferencial para generar el conjunto de datos, así como con el contenedor con la aplicación, AWS IoT Analytics crea un conjunto de datos de producción programada que se ejecuta en el intervalo especificado con los datos de la ventana diferencial y que produce la salida deseada y envía notificaciones.

Puede detener la aplicación del conjunto de datos de producción y reanudarla cuando desee. Por defecto, cuando reinicie la aplicación del conjunto de datos de producción, AWS IoT Analytics se pondrá al día con todos los datos que hayan llegado desde la última ejecución pero que aún no se hayan analizado. También puede configurar la forma en que desea reanudar el conjunto de datos de producción (duración de la ventana de trabajo) realizando una serie de ejecuciones consecutivas. También puede reanudar la aplicación del conjunto de datos de producción capturando únicamente los datos que acaban de llegar que se ajusten al tamaño especificado de la ventana diferencial.

Tenga en cuenta las siguientes limitaciones cuando cree o defina un conjunto de datos que se active por la creación de otro conjunto de datos:

- Solo los conjuntos de datos de contenedores se pueden activar mediante conjuntos de datos SQL.
- Un conjunto de datos SQL puede activar un máximo de 10 conjuntos de datos de contenedores.

Es posible que aparezcan los siguientes errores al crear un conjunto de datos de contenedores que se activa mediante un conjunto de datos SQL:

- "Triggering dataset can only be added on a container dataset" (El conjunto de datos desencadenante solo se puede añadir a un conjunto de datos de contenedores)
- "There can only be one triggering dataset" (Solo puede haber un conjunto de datos desencadenante)

  Este error se produce si se intenta definir un conjunto de datos de contenedores que se activa mediante dos conjuntos de datos SQL diferentes.
- "The triggering data set <dataset-name> cannot be triggered by a container dataset" (El conjunto de datos desencadenante <nombre-del-conjunto-de-datos> no se puede activar por un conjunto de datos de contenedores)

  Este error se produce si se intenta definir otro conjunto de datos de contenedores activado por otro conjunto de datos de contenedores.
- "<N> datasets are already dependent on <dataset-name> dataset" (Ya hay <N> conjuntos de datos que dependen del conjunto de datos <nombre-del-conjunto-de-datos>)

  Este error se produce si se intenta definir otro conjunto de datos de contenedores que se activará desde el conjunto de datos SQL que ya activa otros 10 conjuntos de datos de contenedores.
- "Exactly one trigger type should be provided" (Se debe proporcionar exactamente un tipo de desencadenador)

  Este error se produce si se intenta definir un conjunto de datos que se activa mediante desencadenador programado y un desencadenador de conjunto de datos.

Variables de entrada/salida del contenedor de Docker personalizado

En esta sección, se muestra cómo el programa ejecutado por la imagen de Docker personalizada puede leer variables de entrada y cargar su salida.

Archivo de parámetros

Las variables de entrada y los destinos en los que se desea cargar la salida se almacenan en un archivo JSON ubicado en `/opt/ml/input/data/iotanalytics/params` en la instancia que ejecutará la imagen de Docker. A continuación, se muestra un ejemplo del contenido de ese archivo.

```
{
   "Context": {
       "OutputUris": {
           "html": "s3://aws-iot-analytics-dataset-xxxxxxx/notebook/results/
iotanalytics-xxxxxxx/output.html",
           "ipynb": "s3://aws-iot-analytics-dataset-xxxxxxx/notebook/results/
iotanalytics-xxxxxxx/output.ipynb"
       }
   },
   "Variables": {
       "source_dataset_name": "mydataset",
       "source_dataset_version_id": "xxxx",
       "example_var": "hello world!",
       "custom_output": "s3://aws-iot-analytics/dataset-xxxxxxx/notebook/results/
iotanalytics-xxxxxxx/output.txt"
   }
}
```

Además del nombre y el ID de versión del conjunto de datos, la sección `Variables` contiene las variables especificadas en la invocación a `iotanalytics:CreateDataset`; en este ejemplo, a una variable `example_var` se le ha asignado el valor `hello world!`. También se ha proporcionado un URI de salida personalizado en la variable `custom_output`. El campo `OutputUris` contiene las ubicaciones predeterminadas en las que el contenedor puede cargar su salida; en este ejemplo, se han proporcionado URI de salida predeterminados para las salidas ipynb y html.

Variables de entrada

El programa lanzado por la imagen de Docker puede leer variables del archivo `params`. A continuación, se muestra un ejemplo de un programa que abre el archivo `params`, lo analiza e imprime el valor de la variable `example_var`.

```
import json

with open("/opt/ml/input/data/iotanalytics/params") as param_file:
    params = json.loads(param_file.read())
example_var = params["Variables"]["example_var"]
print(example_var)
```

Carga de la salida

El programa lanzado por la imagen de Docker también podría almacenar su salida en una ubicación de Amazon S3. La salida se debe cargar con una lista de control de acceso "bucket-owner-full-control". La lista de acceso concede al servicio AWS IoT Analytics el control de la salida cargada. En este ejemplo, ampliamos el anterior para cargar el contenido de `example_var` en la ubicación de Amazon S3 definida por `custom_output` en el archivo `params`.

```
import boto3
import json
from urllib.parse import urlparse

ACCESS_CONTROL_LIST = "bucket-owner-full-control"

with open("/opt/ml/input/data/iotanalytics/params") as param_file:
    params = json.loads(param_file.read())
example_var = params["Variables"]["example_var"]

outputUri = params["Variables"]["custom_output"]
# break the S3 path into a bucket and key
bucket = urlparse(outputUri).netloc
key = urlparse(outputUri).path.lstrip("/")

s3_client = boto3.client("s3")
s3_client.put_object(Bucket=bucket, Key=key, Body=example_var, ACL=ACCESS_CONTROL_LIST)
```

Permisos

Debe crear dos roles. Un rol concede permiso para lanzar una instancia de SageMaker con objeto de incluir un bloc de notas en contenedores. El otro es necesario para ejecutar un contenedor.

Puede crear manualmente el primer rol o dejar que el sistema lo cree de forma automática. Si crea la instancia de SageMaker nueva con la consola de AWS IoT Analytics, dispondrá de la opción de crear automáticamente un rol nuevo que concede todos los privilegios necesarios para ejecutar instancias de SageMaker e incluir blocs de notas en contenedores. O bien, puede crear un rol con estos privilegios manualmente. Para ello, cree un rol con la política de `AmazonSageMakerFullAccess` asociada y añada la siguiente política:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
```

```
      "Effect": "Allow",
      "Action": [
        "ecr:BatchDeleteImage",
        "ecr:BatchGetImage",
        "ecr:CompleteLayerUpload",
        "ecr:CreateRepository",
        "ecr:DescribeRepositories",
        "ecr:GetAuthorizationToken",
        "ecr:InitiateLayerUpload",
        "ecr:PutImage",
        "ecr:UploadLayerPart"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject"
      ],
      "Resource": "arn:aws:s3:::iotanalytics-notebook-containers/*"
    }
  ]
}
```

Debe crear manualmente el segundo rol que concede permiso para ejecutar un contenedor. Deberá hacerlo incluso si ha utilizado la consola de AWS IoT Analytics para crear el primer rol de forma automática. Cree un rol que tenga asociadas la política y la política de confianza siguientes.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetBucketLocation",
                "s3:PutObject",
                "s3:GetObject",
                "s3:PutObjectAcl"
            ],
            "Resource": "arn:aws:s3:::aws-*-dataset-*/*"
        },
        {
            "Effect": "Allow",
```

```
            "Action": [
                "iotanalytics:*"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ecr:GetAuthorizationToken",
                "ecr:GetDownloadUrlForLayer",
                "ecr:BatchGetImage",
                "ecr:BatchCheckLayerAvailability",
                "logs:CreateLogGroup",
                "logs:CreateLogStream",
                "logs:DescribeLogStreams",
                "logs:GetLogEvents",
                "logs:PutLogEvents"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetBucketLocation",
                "s3:ListBucket",
                "s3:ListAllMyBuckets"
            ],
            "Resource": "*"
        }
    ]
}
```

A continuación, se muestra un ejemplo de una política de confianza.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": ["sagemaker.amazonaws.com", "iotanalytics.amazonaws.com"]
      },
```

```
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Uso de la API CreateDataset mediante Java y AWS CLI

Crea un conjunto de datos. Un conjunto de datos almacena datos recuperados de un almacén de datos aplicando una `queryAction` (una consulta SQL) o una `containerAction` (ejecución de una aplicación en contenedores). Esta operación crea el esqueleto de un conjunto de datos. El conjunto de datos se puede rellenar manualmente llamando `CreateDatasetContent` o automáticamente de acuerdo con un `trigger` especificado. Para obtener más información, consulte CreateDataset y `CreateDatasetContent`.

Temas

Ejemplo 1. Creación de un conjunto de datos SQL (java)

```
CreateDatasetRequest request = new CreateDatasetRequest();
request.setDatasetName(dataSetName);
DatasetAction action = new DatasetAction();

//Create Action
action.setActionName("SQLAction1");
action.setQueryAction(new SqlQueryDatasetAction().withSqlQuery("select * from
 DataStoreName"));

// Add Action to Actions List
```

```
List<DatasetAction> actions = new ArrayList<DatasetAction>();
actions.add(action);

//Create Trigger
DatasetTrigger trigger = new DatasetTrigger();
trigger.setSchedule(new Schedule().withExpression("cron(0 12 * * ? *)"));

//Add Trigger to Triggers List
List<DatasetTrigger> triggers = new ArrayList<DatasetTrigger>();
triggers.add(trigger);

// Add Triggers and Actions to CreateDatasetRequest object
request.setActions(actions);
request.setTriggers(triggers);

// Add RetentionPeriod to CreateDatasetRequest object
request.setRetentionPeriod(new RetentionPeriod().withNumberOfDays(10));
final CreateDatasetResult result = iot.createDataset(request);
```

Salida en caso de éxito:

```
{DatasetName: <datatsetName>, DatasetArn: <datatsetARN>, RetentionPeriod: {unlimited:
 true} or {numberOfDays: 10, unlimited: false}}
```

Ejemplo 2. Creación de un conjunto de datos SQL con una ventana diferencial (java)

```
CreateDatasetRequest request = new CreateDatasetRequest();
request.setDatasetName(dataSetName);
DatasetAction action = new DatasetAction();

//Create Filter for DeltaTime
QueryFilter deltaTimeFilter = new QueryFilter();
deltaTimeFilter.withDeltaTime(
                new DeltaTime()
                .withOffsetSeconds(-1 * EstimatedDataDelayInSeconds)
                .withTimeExpression("from_unixtime(timestamp)"));

//Create Action
action.setActionName("SQLActionWithDeltaTime");
action.setQueryAction(new SqlQueryDatasetAction()
                .withSqlQuery("SELECT * from DataStoreName")
```

```
                .withFilters(deltaTimeFilter));

// Add Action to Actions List
List<DatasetAction> actions = new ArrayList<DatasetAction>();
actions.add(action);

//Create Trigger
DatasetTrigger trigger = new DatasetTrigger();
trigger.setSchedule(new Schedule().withExpression("cron(0 12 * * ? *)"));

//Add Trigger to Triggers List
List<DatasetTrigger> triggers = new ArrayList<DatasetTrigger>();
triggers.add(trigger);

// Add Triggers and Actions to CreateDatasetRequest object
request.setActions(actions);
request.setTriggers(triggers);

// Add RetentionPeriod to CreateDatasetRequest object
request.setRetentionPeriod(new RetentionPeriod().withNumberOfDays(10));
final CreateDatasetResult result = iot.createDataset(request);
```

Salida en caso de éxito:

```
{DatasetName: <datatsetName>, DatasetArn: <datatsetARN>, RetentionPeriod: {unlimited:
 true} or {numberOfDays: 10, unlimited: false}}
```

Ejemplo 3. Creación de un conjunto de datos de contenedores con su propio desencadenador de programación (java)

```
CreateDatasetRequest request = new CreateDatasetRequest();
request.setDatasetName(dataSetName);
DatasetAction action = new DatasetAction();

//Create Action
action.setActionName("ContainerActionDataset");
action.setContainerAction(new ContainerDatasetAction()
        .withImage(ImageURI)
        .withExecutionRoleArn(ExecutionRoleArn)
        .withResourceConfiguration(
                new ResourceConfiguration()
                .withComputeType(new ComputeType().withAcu(1))
```

```
                .withVolumeSizeInGB(1))
        .withVariables(new Variable()
        .withName("VariableName")
        .withStringValue("VariableValue"));

// Add Action to Actions List
List<DatasetAction> actions = new ArrayList<DatasetAction>();
actions.add(action);

//Create Trigger
DatasetTrigger trigger = new DatasetTrigger();
trigger.setSchedule(new Schedule().withExpression("cron(0 12 * * ? *)"));

//Add Trigger to Triggers List
List<DatasetTrigger> triggers = new ArrayList<DatasetTrigger>();
triggers.add(trigger);

// Add Triggers and Actions to CreateDatasetRequest object
request.setActions(actions);
request.setTriggers(triggers);

// Add RetentionPeriod to CreateDatasetRequest object
request.setRetentionPeriod(new RetentionPeriod().withNumberOfDays(10));
final CreateDatasetResult result = iot.createDataset(request);
```

Salida en caso de éxito:

```
{DatasetName: <datatsetName>, DatasetArn: <datatsetARN>, RetentionPeriod: {unlimited:
 true} or {numberOfDays: 10, unlimited: false}}
```

Ejemplo 4. Creación de un conjunto de datos de contenedores con un conjunto de datos SQL como desencadenador (java)

```
CreateDatasetRequest request = new CreateDatasetRequest();
request.setDatasetName(dataSetName);
DatasetAction action = new DatasetAction();

//Create Action
action.setActionName("ContainerActionDataset");
action.setContainerAction(new ContainerDatasetAction()
        .withImage(ImageURI)
        .withExecutionRoleArn(ExecutionRoleArn)
```

```
        .withResourceConfiguration(
                new ResourceConfiguration()
                .withComputeType(new ComputeType().withAcu(1))
                .withVolumeSizeInGB(1))
        .withVariables(new Variable()
        .withName("VariableName")
        .withStringValue("VariableValue"));

// Add Action to Actions List
List<DatasetAction> actions = new ArrayList<DatasetAction>();
actions.add(action);

//Create Trigger
DatasetTrigger trigger = new DatasetTrigger()
        .withDataset(new TriggeringDataset()
                .withName(TriggeringSQLDataSetName));

//Add Trigger to Triggers List
List<DatasetTrigger> triggers = new ArrayList<DatasetTrigger>();
triggers.add(trigger);

// Add Triggers and Actions to CreateDatasetRequest object
request.setActions(actions);
request.setTriggers(triggers);
final CreateDatasetResult result = iot.createDataset(request);
```

Salida en caso de éxito:

```
{DatasetName: <datatsetName>, DatasetArn: <datatsetARN>}
```

Ejemplo 5. Creación de un conjunto de datos SQL (CLI)

```
aws iotanalytics --endpoint <EndPoint>  --region <Region> create-dataset --dataset-
name="<dataSetName>" --actions="[{\"actionName\":\"<ActionName>\", \"queryAction\":
{\"sqlQuery\":\"<SQLQuery>\"}}]" --retentionPeriod numberOfDays=10
```

Salida en caso de éxito:

```
{
    "datasetName": "<datasetName>",
    "datasetArn": "<datatsetARN>",
```

```
    "retentionPeriod": {unlimited: true} or {numberOfDays: 10, unlimited: false}
}
```

Ejemplo 6. Creación de un conjunto de datos SQL con una ventana diferencial (CLI)

Las ventanas diferenciales son una serie de intervalos de tiempo definidos por el usuario, no solapados y continuos. Las ventanas diferenciales le permiten crear el contenido del conjunto de datos y realizar análisis sobre los nuevos datos que han llegado al almacén de datos desde el último análisis. Para crear una ventana diferencial, establezca el valor `deltaTime` en la parte `filters` de una `queryAction` de un conjunto de datos (CreateDataset). Normalmente, querrá crear el contenido del conjunto de datos automáticamente estableciendo un desencadenador de intervalo de tiempo (`triggers:schedule:expression`). Básicamente, esto le permite filtrar los mensajes que han llegado durante un periodo de tiempo específico, para que los datos contenidos en mensajes de periodos de tiempo anteriores no se cuenten dos veces.

En este ejemplo, creamos un nuevo conjunto de datos que crea automáticamente nuevo contenido de conjunto de datos cada 15 minutos únicamente mediante los datos que han llegado desde la última vez. Especificamos un desplazamiento `deltaTime` de 3 minutos (180 segundos) que permite un retraso de 3 minutos para que los mensajes lleguen al almacén de datos especificado. Por lo tanto, si el contenido del conjunto de datos se crea a las 10:30 h, los datos utilizados (que se incluyen en el contenido del conjunto de datos) serían los que tienen marcas de tiempo entre las 10:12 h y las 10:27 (es decir las 10:30 h - 15 minutos - 3 minutos para las 10:30 h - 3 minutos).

```
aws iotanalytics --endpoint <EndPoint>  --region <Region> create-dataset --cli-input-
json file://delta-window.json
```

Donde el archivo `delta-window.json` contiene lo siguiente.

```
{
  "datasetName": "delta_window_example",
  "actions": [
    {
      "actionName": "delta_window_action",
      "queryAction": {
        "sqlQuery": "SELECT temperature, humidity, timestamp FROM my_datastore",
        "filters": [
          {
            "deltaTime": {
```

```
              "offsetSeconds": -180,
              "timeExpression": "from_unixtime(timestamp)"
            }
          }
        ]
      }
    }
  ],
  "triggers": [
    {
      "schedule": {
        "expression": "cron(0/15 * * * ? *)"
      }
    }
  ]
}
```

Salida en caso de éxito:

```
{
    "datasetName": "<datasetName>",
    "datasetArn": "<datatsetARN>",
}
```

Inclusión de un bloc de notas en contenedores

Esta sección incluye información sobre cómo crear un contenedor de Docker mediante un cuaderno de Jupyter. Existe un riesgo para la seguridad si reutiliza blocs de notas creados por terceros: los contenedores incluidos pueden ejecutar un código arbitrario con sus permisos de usuario. Además, el código HTML generado por el bloc de notas se puede mostrar en la consola de AWS IoT Analytics, lo que proporciona un posible vector de ataque en el equipo que muestra el código HTML. Asegúrese de que el autor del bloc de notas de terceros es de confianza antes de utilizarlo.

Una opción para realizar funciones de análisis avanzadas consiste en utilizar el cuaderno de Jupyter. El cuaderno de Jupyter proporciona potentes herramientas de ciencia de datos que pueden llevar a cabo machine learning y una amplia gama de análisis estadísticos. Para obtener más información, consulte Plantillas de bloc de notas. (Tenga en cuenta que la creación de contenedores no es actualmente compatible en JupyterLab). Puede empaquetar su cuaderno de Jupyter y sus bibliotecas en un contenedor que se ejecute periódicamente en un nuevo lote de datos a medida que AWS IoT Analytics lo reciba durante una ventana de tiempo delta definida. Puede programar un trabajo

de análisis que utiliza el contenedor y los datos segmentados nuevos capturados en el periodo de tiempo especificado y, a continuación, almacena la salida del trabajo para futuros análisis programados.

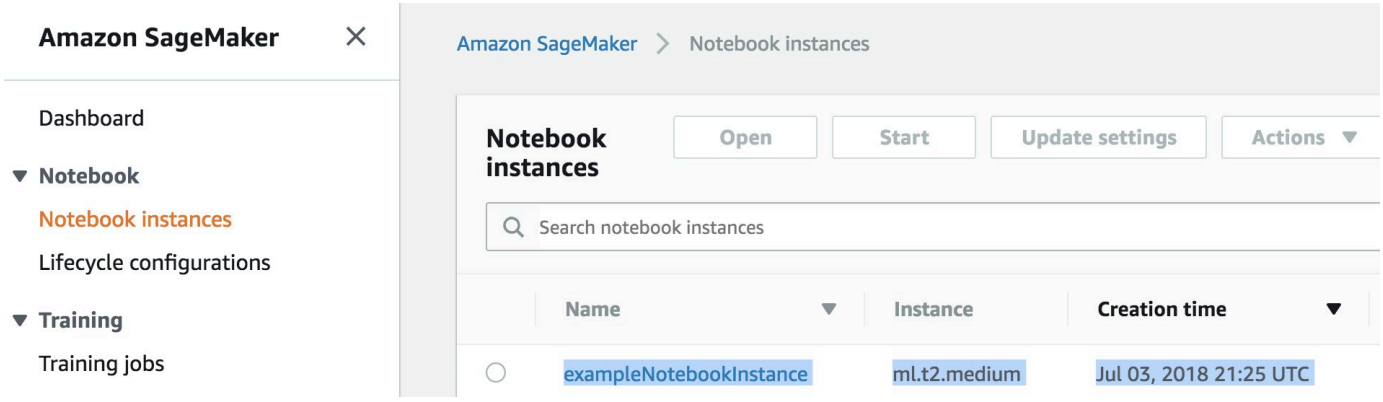
Si ha creado una instancia de SageMaker con la consola de AWS IoT Analytics después del 23 de agosto de 2018, la instalación de la extensión de creación de contenedores se ha llevado a cabo automáticamente y puede comenzar a crear una imagen cargada en contenedores. De lo contrario, siga los pasos que se indican en esta sección para habilitar la creación de contenedores de blocs de notas en la instancia de SageMaker. A continuación, modificará el rol de ejecución de SageMaker de forma que le permita cargar la imagen del contenedor en Amazon EC2 e instalará la extensión de creación de contenedores.

Habilitar la creación de contenedores de instancias de bloc de notas no creadas a través de la consola de AWS IoT Analytics

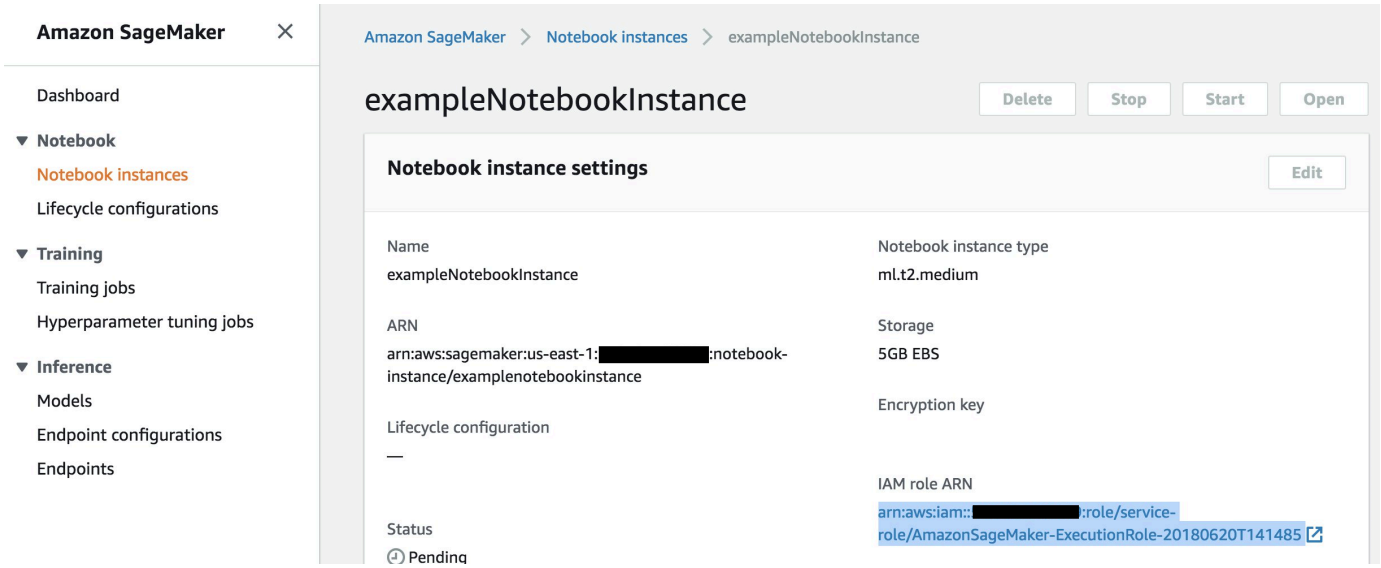
Le recomendamos que cree una nueva instancia de SageMaker mediante la consola de AWS IoT Analytics en lugar de seguir estos pasos. Las instancias nuevas admiten de forma automática la creación de contenedores.

Si reinicia la instancia de SageMaker después de habilitar la creación de contenedores tal y como se muestra aquí, no tendrá que volver a agregar los roles y las políticas de IAM, pero sí tendrá que volver a instalar la extensión, tal como se muestra en el último paso.

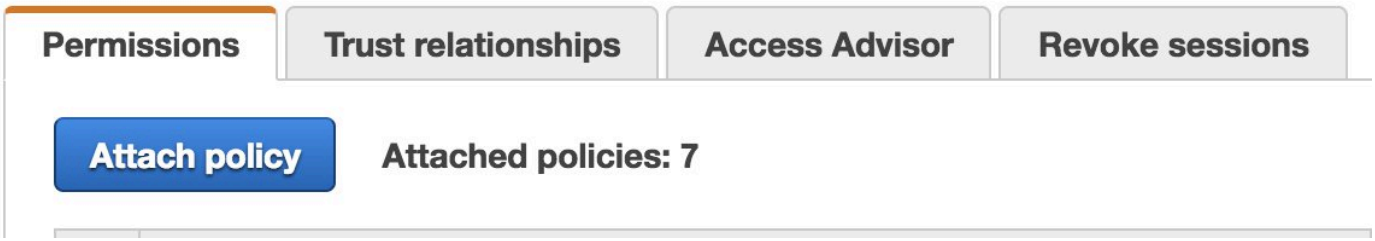
1. Para conceder a la instancia del bloc de notas acceso a Amazon ECS, seleccione la instancia de SageMaker en la página de SageMaker:



2. En ARN del rol de IAM, seleccione el rol de ejecución de SageMaker.

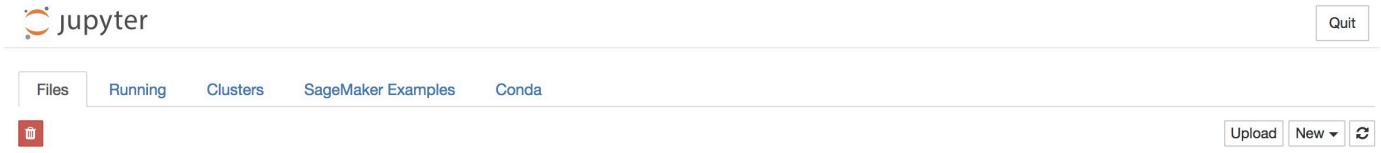


3. Seleccione Attach Policy (Asociar política) y, a continuación, defina y asocie la política mostrada en la sección Permissions (Permisos). Si aún no has añadido la política de `AmazonSageMakerFullAccess`, adjúntela también.



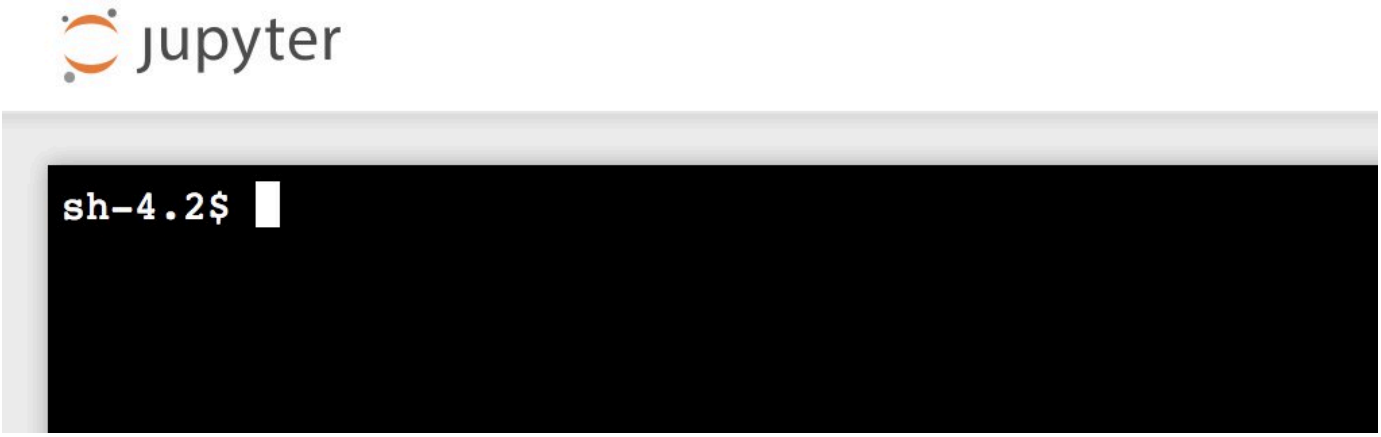
También debe descargar el código de creación de contenedores de Amazon S3 e instalarlo en su instancia con cuaderno. El primer paso es acceder al terminal de la instancia de SageMaker.

1. En Jupyter, seleccione New (Nuevo)



2. En el menú que aparece, seleccione Terminal.

3. En el terminal, introduzca los siguientes comandos para descargar el código, descomprimirlo e instalarlo. Tenga en cuenta que estos comandos finalizarán los procesos que se estén ejecutando en los blocs de notas de esta instancia de SageMaker.




```
cd /tmp

aws s3 cp s3://iotanalytics-notebook-containers/iota_notebook_containers.zip /tmp

unzip iota_notebook_containers.zip

cd iota_notebook_containers

chmod u+x install.sh

./install.sh
```

Espere un minuto o dos hasta que la extensión se valide e instale.

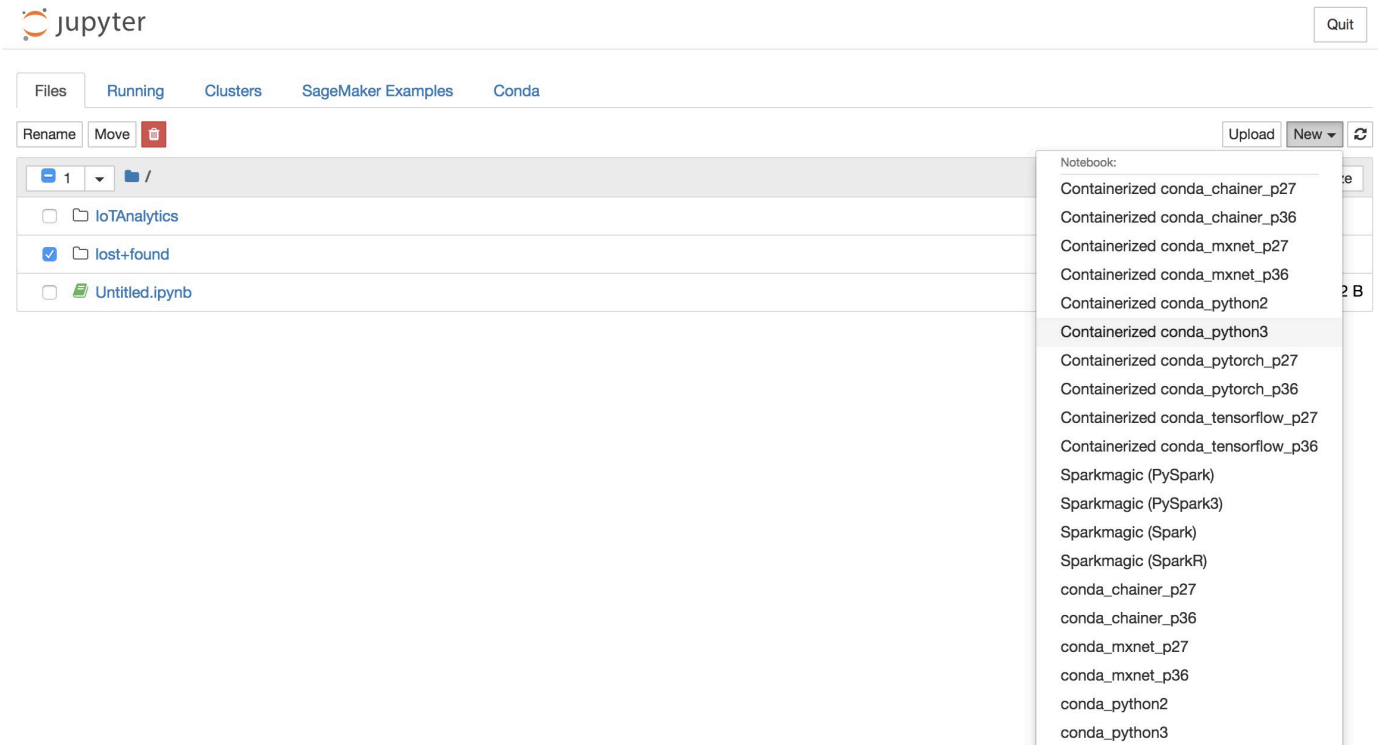
Actualización de la extensión de creación de contenedores del bloc de notas

Si ha creado la instancia de SageMaker mediante la consola de AWS IoT Analytics después del 23 de agosto de 2018, la extensión de creación de contenedores se ha instalado automáticamente. Puede actualizar la extensión reiniciando la instancia desde la consola de SageMaker. Si ha instalado la extensión manualmente, puede actualizarla volviendo a ejecutar los comandos de terminal que se indican en la sección Habilitación de creación de contenedores de instancias de bloc de notas que no se han creado mediante la consola de AWS IoT Analytics.

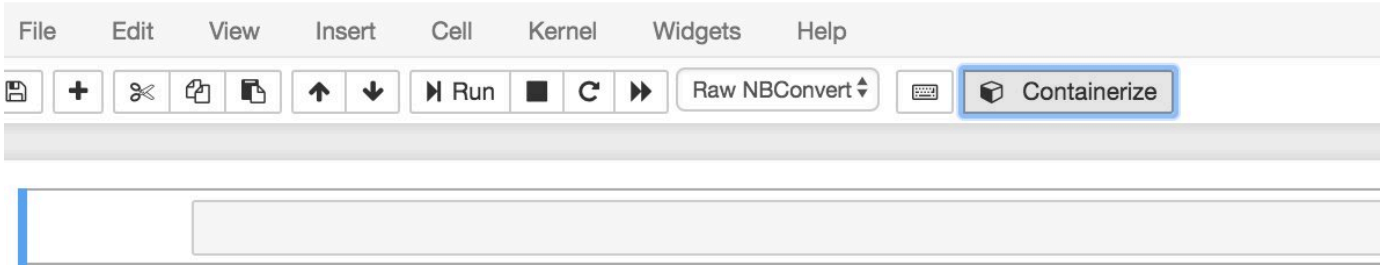
Creación de una imagen en contenedores

En esta sección, se muestran los pasos necesarios para incluir un bloc de notas en contenedores. Para empezar, vaya al cuaderno de Jupyter para crear un bloc de notas con un kernel en contenedores.

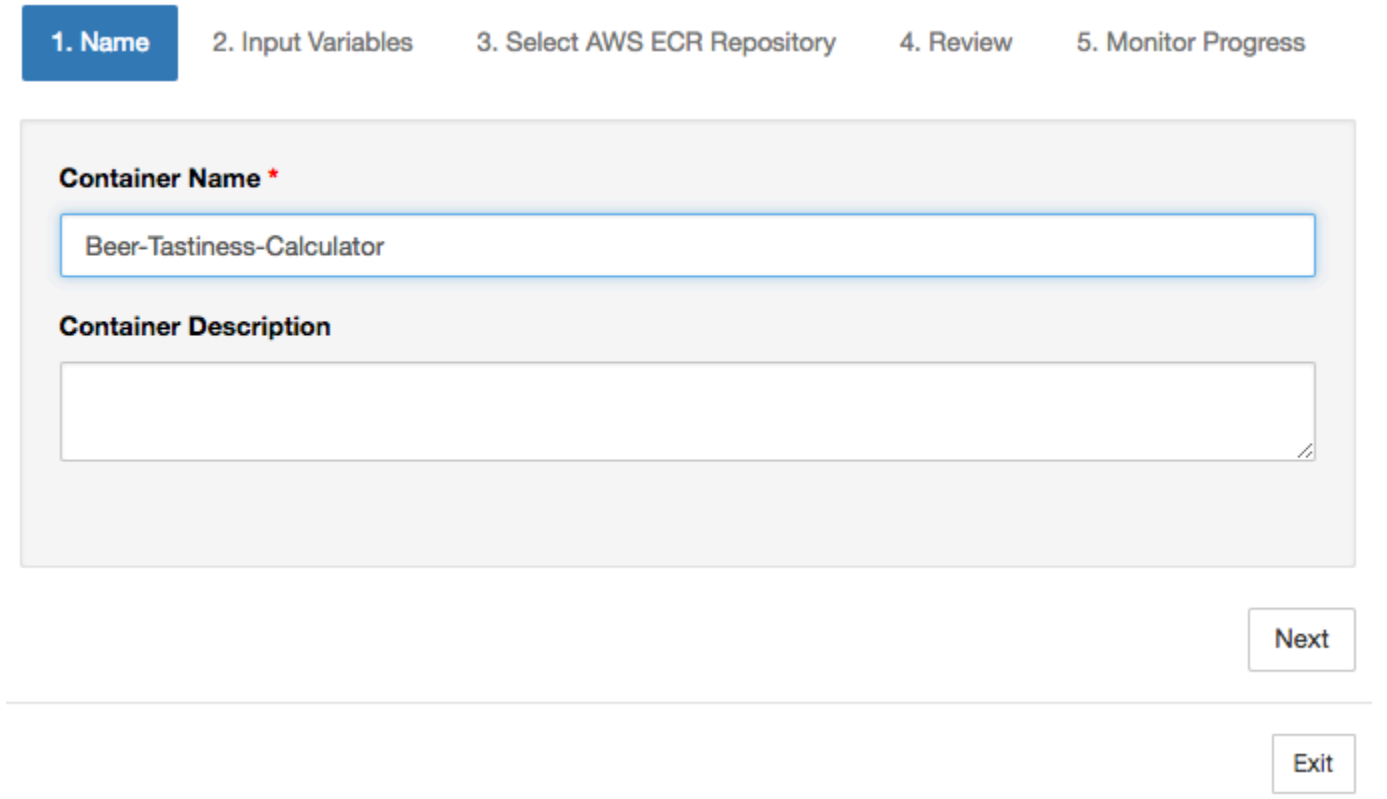
1. En el cuaderno de Jupyter, seleccione New (Nuevo) y, a continuación, seleccione el tipo de kernel que desea en la lista desplegable. (El tipo de kernel debe empezar con "Containerized" (En contenedores) y terminar con el kernel que hubiera seleccionado de otro modo. Por ejemplo, si solo desea un entorno Python 3.0 sencillo como "conda_python3", seleccione "Containerized conda_python3").



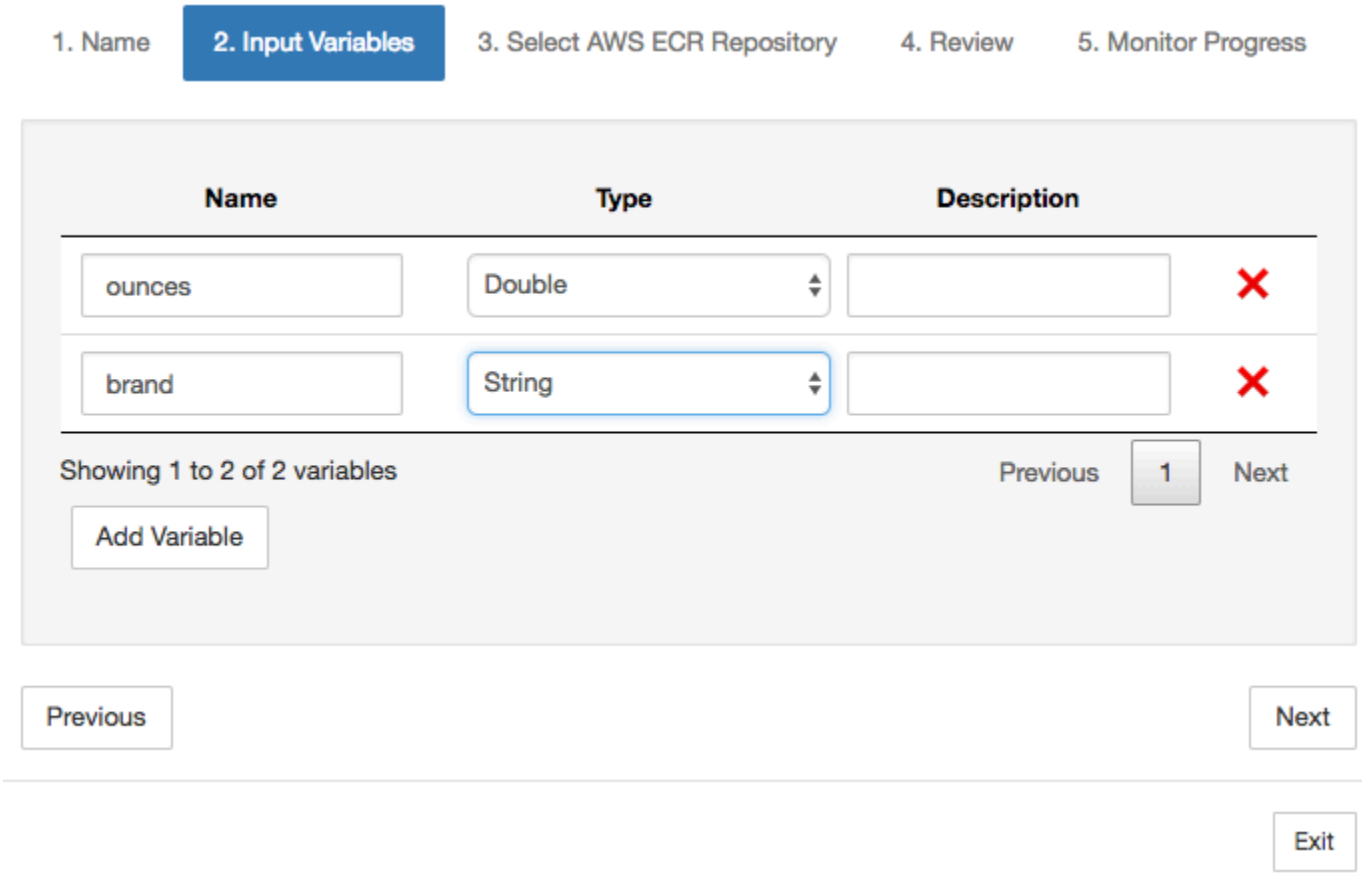
2. Una vez que haya completado el trabajo en el bloc de notas y desee incluirlo en contenedores, seleccione Crear contenedores.



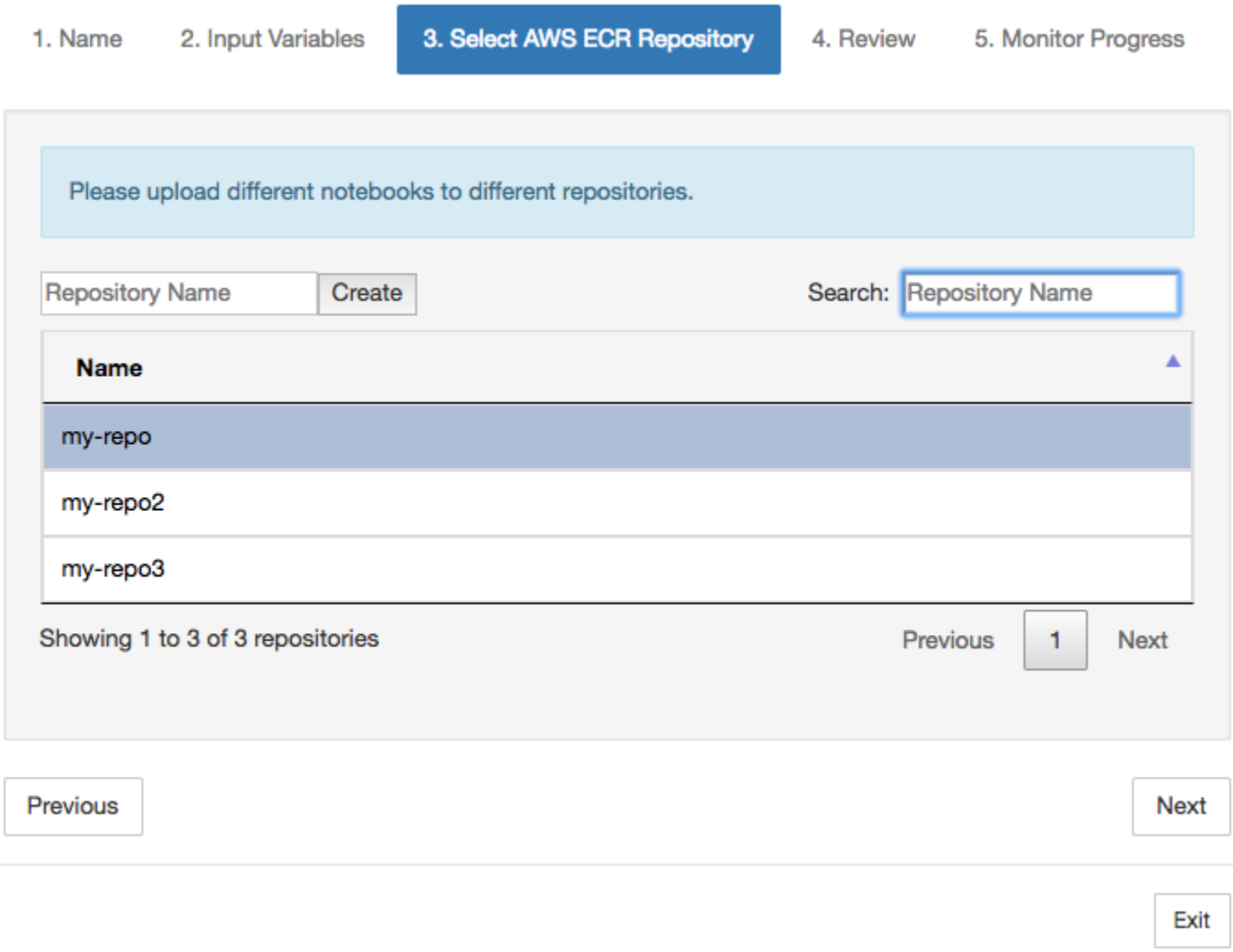
3. Introduzca un nombre para el bloc de notas en contenedores. También puede introducir una descripción opcional.



4. Especifique las Input Variables (Variables de entrada) (los parámetros) con los que debe invocarse el bloc de notas. Puede seleccionar las variables de entrada que se detectan automáticamente en el bloc de notas o definir variables personalizadas. (Tenga en cuenta que las variables de entrada solo se detectan si ha ejecutado previamente el bloc de notas). Elija un tipo para cada variable de entrada. Si lo desea, también puede introducir una descripción opcional de la variable de entrada.

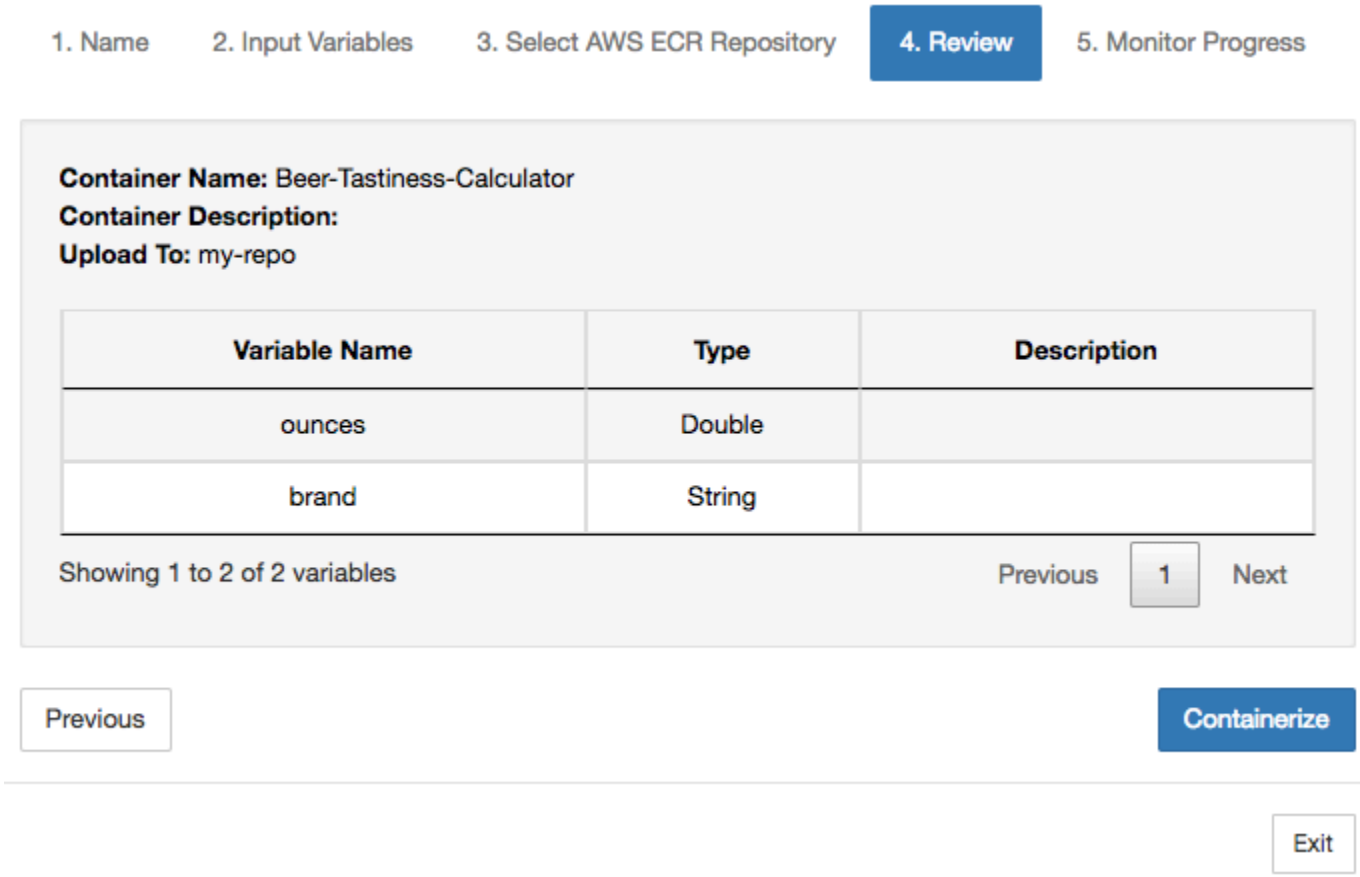


5. Seleccione el repositorio de Amazon ECR donde se debe cargar la imagen creada a partir del bloc de notas.

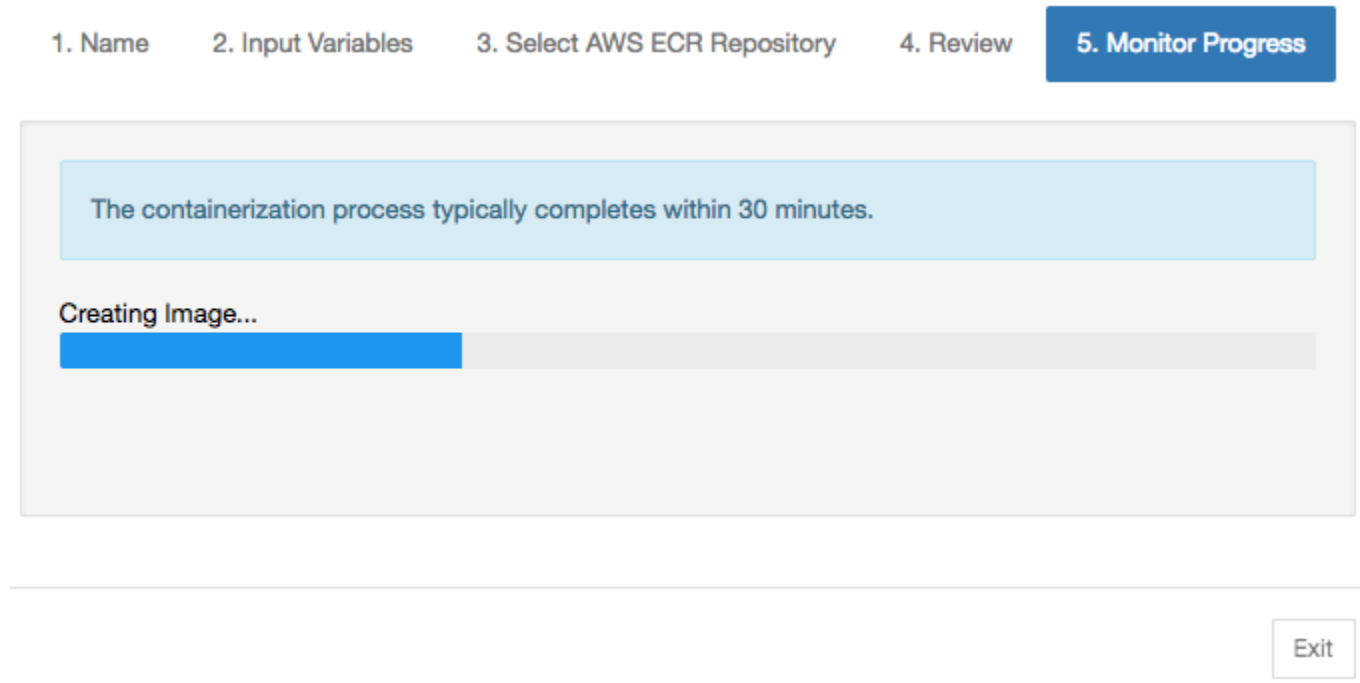


6. Elija Crear de contenedores para iniciar el proceso.

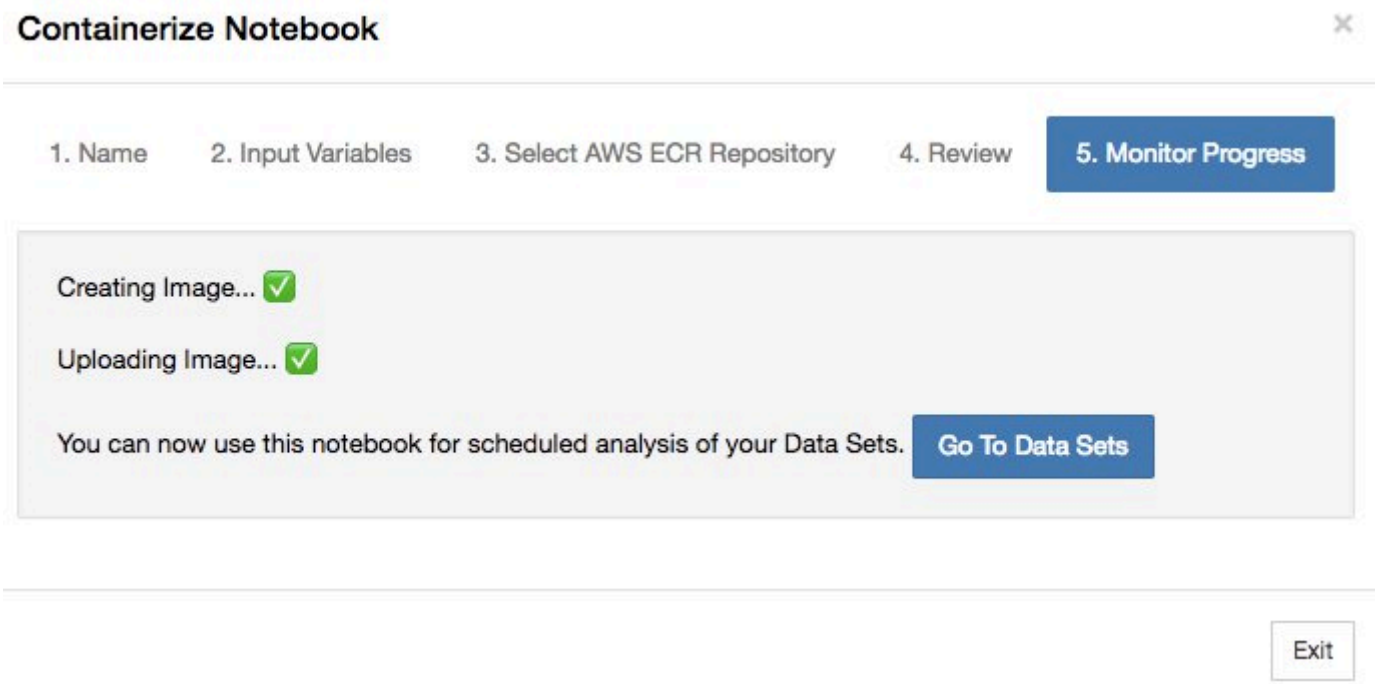
   Se mostrará un resumen de su entrada. Tenga en cuenta que una vez iniciado el proceso no podrá cancelarlo. El proceso puede durar hasta una hora.



7. La siguiente página muestra el progreso.



8. Si cierra el navegador de forma accidental, puede monitorizar el estado del proceso de creación de contenedores en la sección Blocs de notas de la consola de AWS IoT Analytics.
9. Una vez completado el proceso, la imagen en contenedor se almacena en Amazon ECR lista para su uso.



Uso de un contenedor personalizado para el análisis

Esta sección incluye información sobre cómo crear un contenedor de Docker mediante un cuaderno de Jupyter. Existe un riesgo para la seguridad si reutiliza blocs de notas creados por terceros: los contenedores incluidos pueden ejecutar un código arbitrario con sus permisos de usuario. Además, el código HTML generado por el bloc de notas se puede mostrar en la consola de AWS IoT Analytics, lo que proporciona un posible vector de ataque en el equipo que muestra el código HTML. Asegúrese de que el autor del bloc de notas de terceros es de confianza antes de utilizarlo.

Puede crear su propio contenedor personalizado y ejecutarlo con el servicio AWS IoT Analytics. Para ello, configure una imagen de Docker y cárguela en Amazon ECR; a continuación, configure un conjunto de datos y ejecute una acción de contenedor. En esta sección, se muestra un ejemplo del proceso con Octave.

En este tutorial, se supone que:

- Ha instalado Octave en el equipo local.

- Ha configurado una cuenta de Docker en el equipo local.
- Una cuenta de AWS con acceso a Amazon ECR o AWS IoT Analytics

Paso 1: Configurar una imagen de Docker

Existen tres archivos principales que necesita para este tutorial. Sus nombres y su contenido son los siguientes:

- `Dockerfile`: la configuración inicial para el proceso de creación de contenedores de Docker.

```
FROM ubuntu:16.04

# Get required set of software
RUN apt-get update
RUN apt-get install -y software-properties-common
RUN apt-get install -y octave
RUN apt-get install -y python3-pip

# Get boto3 for S3 and other libraries
RUN pip3 install --upgrade pip
RUN pip3 install boto3
RUN pip3 install urllib3

# Move scripts over
ADD moment moment
ADD run-octave.py run-octave.py

# Start python script
ENTRYPOINT ["python3", "run-octave.py"]
```

- `run-octave.py`: analiza el JSON desde AWS IoT Analytics, ejecuta el script de Octave y carga los artefactos en Amazon S3.

```
import boto3
import json
import os
import sys
from urllib.parse import urlparse

# Parse the JSON from IoT Analytics
with open('/opt/ml/input/data/iotanalytics/params') as params_file:
    params = json.load(params_file)
```

```
variables = params['Variables']

order = variables['order']
input_s3_bucket = variables['inputDataS3BucketName']
input_s3_key = variables['inputDataS3Key']
output_s3_uri = variables['octaveResultS3URI']

local_input_filename = "input.txt"
local_output_filename = "output.mat"

# Pull input data from S3...
s3 = boto3.resource('s3')
s3.Bucket(input_s3_bucket).download_file(input_s3_key, local_input_filename)

# Run Octave Script
os.system("octave moment {} {} {}".format(local_input_filename,
 local_output_filename, order))

# # Upload the artifacts to S3
output_s3_url = urlparse(output_s3_uri)
output_s3_bucket = output_s3_url.netloc
output_s3_key = output_s3_url.path[1:]

s3.Object(output_s3_bucket, output_s3_key).put(Body=open(local_output_filename,
 'rb'), ACL='bucket-owner-full-control')
```

- `moment`: un script de Octave sencillo que calcula el momento en función de un archivo de entrada o salida y un orden especificado.

```
#!/usr/bin/octave -qf

arg_list = argv ();
input_filename = arg_list{1};
output_filename = arg_list{2};
order = str2num(arg_list{3});

[D,delimiterOut]=importdata(input_filename)
M = moment(D, order)

save(output_filename,'M')
```

1. Descargue el contenido de cada archivo. Cree un nuevo directorio y coloque todos los archivos en él y luego `cd` a ese directorio.
2. Ejecute el siguiente comando.

```
docker build -t octave-moment .
```

3. Debería ver una imagen nueva en el repositorio de Docker. Verifíquelo ejecutando el siguiente comando.

```
docker image ls | grep octave-moment
```

Paso 2: Cargar la imagen de Docker en un repositorio de Amazon ECR

1. Cree un repositorio en Amazon ECR.

```
aws ecr create-repository --repository-name octave-moment
```

2. Obtenga el inicio de sesión para el entorno de Docker.

```
aws ecr get-login
```

3. Copie la salida y ejecútela. El resultado debería tener un aspecto similar al siguiente.

```
docker login -u AWS -p password -e none https://your-aws-account-
id.dkr.ecr..amazonaws.com
```

4. Etiquete la imagen que ha creado con la etiqueta del repositorio de Amazon ECR.

```
docker tag your-image-id  your-aws-account-id.dkr.ecr.region.amazonaws.com/octave-
moment
```

5. Envíe la imagen a Amazon ECR.

```
docker push your-aws-account-id.dkr.ecr.region.amazonaws.com/octave-moment
```

Paso 3: Cargar los datos de ejemplo en un bucket de Amazon S3

1. Descargue lo siguiente en el archivo `input.txt`.

```
0.857549  -0.987565  -0.467288  -0.252233  -2.298007
 0.030077  -1.243324  -0.692745   0.563276   0.772901
-0.508862  -0.404303  -1.363477  -1.812281  -0.296744
-0.203897   0.746533   0.048276   0.075284   0.125395
 0.829358   1.246402  -1.310275  -2.737117   0.024629
 1.206120   0.895101   1.075549   1.897416   1.383577
```

2. Cree un bucket de Amazon S3 denominado `octave-sample-data-`*`your-aws-account-id`*.
3. Cargue el archivo `input.txt` en el bucket de Amazon S3 que acaba de crear. Ahora debería tener un bucket denominado `octave-sample-data-`*`your-aws-account-id`* que contiene el archivo `input.txt`.

Paso 4: Crear un rol de ejecución de contenedores

1. Copie lo siguiente en un archivo denominado `role1.json`. Sustituya *`your-aws-account-id`* por su ID de cuenta de AWS y *`región de AWS`* por la región de AWS de sus recursos de AWS.

> **Note**
>
> Este ejemplo incluye una clave de contexto de condición global para protegerse contra el problema de seguridad de suplente confuso. Para obtener más información, consulte the section called “Prevención del suplente confuso entre servicios”.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "sagemaker.amazonaws.com",
                    "iotanalytics.amazonaws.com"
                ]
            },
            "Action": "sts:AssumeRole",
            "Condition": {
              "StringEquals": {
                "aws:SourceAccount": "your-aws-account-id"
```

```
            },
            "ArnLike": {
              "aws:SourceArn": "arn:aws:iotanalytics:aws-region:your-aws-account-
id:dataset/DOC-EXAMPLE-DATASET"
            }
          }
    ]
}
```

2. Cree un rol que otorgue permisos de acceso a SageMaker y a AWS IoT Analytics, utilizando el archivo `role1.json` que descargó.

```
aws iam create-role --role-name container-execution-role --assume-role-policy-
document file://role1.json
```

3. Descargue lo siguiente en un archivo con el nombre `policy1.json` y sustituya *your-account-id* por su ID de cuenta (consulte el segundo ARN en `Statement:Resource`).

```
{
 "Version": "2012-10-17",
 "Statement": [
   {
     "Effect": "Allow",
     "Action": [
       "s3:GetBucketLocation",
       "s3:PutObject",
       "s3:GetObject",
       "s3:PutObjectAcl"
     ],
     "Resource": [
       "arn:aws:s3:::*-dataset-*/*",
       "arn:aws:s3:::octave-sample-data-your-account-id/*"
   },
   {
     "Effect": "Allow",
     "Action": [
       "iotanalytics:*"
     ],
     "Resource": "*"
   },
   {
     "Effect": "Allow",
     "Action": [
```

```
        "ecr:GetAuthorizationToken",
        "ecr:GetDownloadUrlForLayer",
        "ecr:BatchGetImage",
        "ecr:BatchCheckLayerAvailability",
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:DescribeLogStreams",
        "logs:GetLogEvents",
        "logs:PutLogEvents"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetBucketLocation",
        "s3:ListBucket",
        "s3:ListAllMyBuckets"
      ],
      "Resource" : "*"
    }
  ]
}
```

4. Cree una política de IAM utilizando el archivo `policy.json` que acaba de descargar.

```
aws iam create-policy --policy-name ContainerExecutionPolicy --policy-document
 file://policy1.json
```

5. Asocie la política al rol.

```
aws iam attach-role-policy --role-name container-execution-role --policy-arn
 arn:aws:iam::your-account-id:policy/ContainerExecutionPolicy
```

Paso 5: Crear un conjunto de datos con una acción de contenedor

1. Descargue lo siguiente en un archivo con el nombre `cli-input.json` y sustituya todas las instancias de *`your-account-id`* y de *`region`* por los valores correspondientes.

```
{
    "datasetName": "octave_dataset",
    "actions": [
```

```
        {
            "actionName": "octave",
            "containerAction": {
                "image": "your-account-id.dkr.ecr.region.amazonaws.com/octave-
moment",
                "executionRoleArn": "arn:aws:iam::your-account-id:role/container-
execution-role",
                "resourceConfiguration": {
                    "computeType": "ACU_1",
                    "volumeSizeInGB": 1
                },
                "variables": [
                    {
                        "name": "octaveResultS3URI",
                        "outputFileUriValue": {
                            "fileName": "output.mat"
                        }
                    },
                    {
                        "name": "inputDataS3BucketName",
                        "stringValue": "octave-sample-data-your-account-id"
                    },
                    {
                        "name": "inputDataS3Key",
                        "stringValue": "input.txt"
                    },
                    {
                        "name": "order",
                        "stringValue": "3"
                    }
                ]
            }
        }
    ]
}
```

2. Cree un conjunto de datos utilizando el archivo `cli-input.json` que acaba de descargar y editar.

```
aws iotanalytics create-dataset –cli-input-json file://cli-input.json
```

Paso 6: Invocar la generación del contenido del conjunto de datos

1. Ejecute el siguiente comando.

```
aws iotanalytics create-dataset-content --dataset-name octave-dataset
```

Paso 7: Obtener el contenido del conjunto de datos

1. Ejecute el siguiente comando.

```
aws iotanalytics get-dataset-content --dataset-name octave-dataset --version-id \
$LATEST
```

2. Es posible que tenga que esperar varios minutos hasta que `DatasetContentState` sea `SUCCEEDED`.

Paso 8: Imprimir la salida en Octave

1. Utilice el intérprete de comandos de Octave para imprimir la salida del contenedor ejecutando el siguiente comando.

```
bash> octave
octave> load output.mat
octave> disp(M)
-0.016393 -0.098061 0.380311 -0.564377 -1.318744
```

Visualizar datos de AWS IoT Analytics

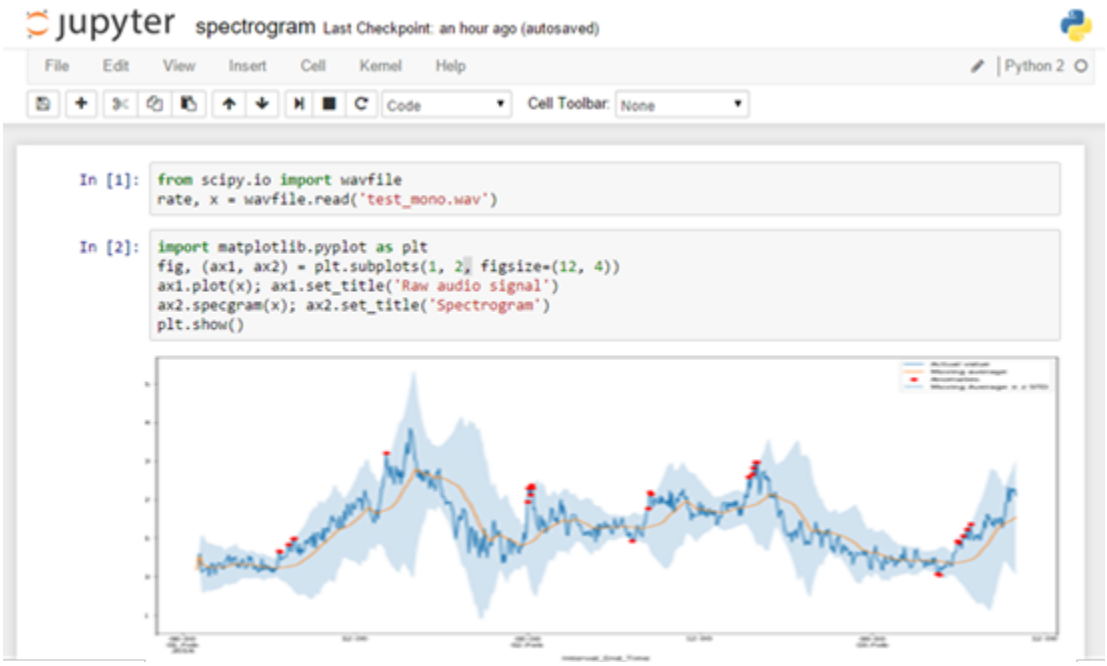
Para visualizar los datos de AWS IoT Analytics, puede utilizar la consola de AWS IoT Analytics o Amazon QuickSight.

Temas

- Visualización de datos de AWS IoT Analytics con la consola
- Visualización de datos de AWS IoT Analytics con Amazon QuickSight

Visualización de datos de AWS IoT Analytics con la consola

AWS IoT Analytics puede integrar la salida HTML del conjunto de datos del contenedor (que se encuentra en el archivo `output.html`) en la página del contenido del conjunto de datos del contenedor de la consola de AWS IoT Analytics. Por ejemplo, si define un conjunto de datos del contenedor que ejecuta un cuaderno de Jupyter y crea una visualización en su cuaderno de Jupyter, su conjunto de datos podría tener el siguiente aspecto.



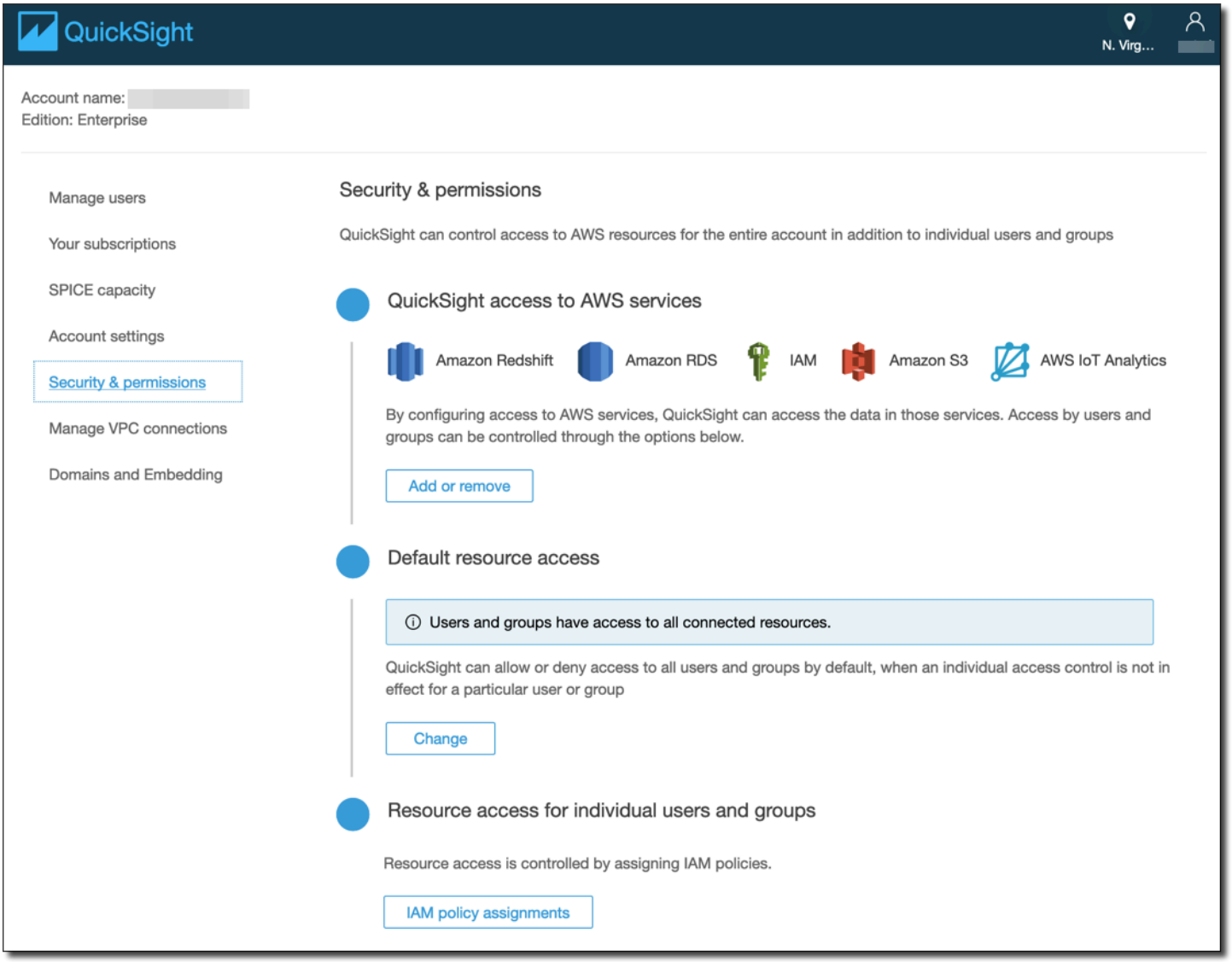
Una vez creado el contenido del conjunto de datos del contenedor, verá la página de contenido Data Set de la consola.

Para obtener información sobre la creación de un conjunto de datos del contenedor que ejecute un cuaderno de Jupyter, consulte Automatización del flujo de trabajo.

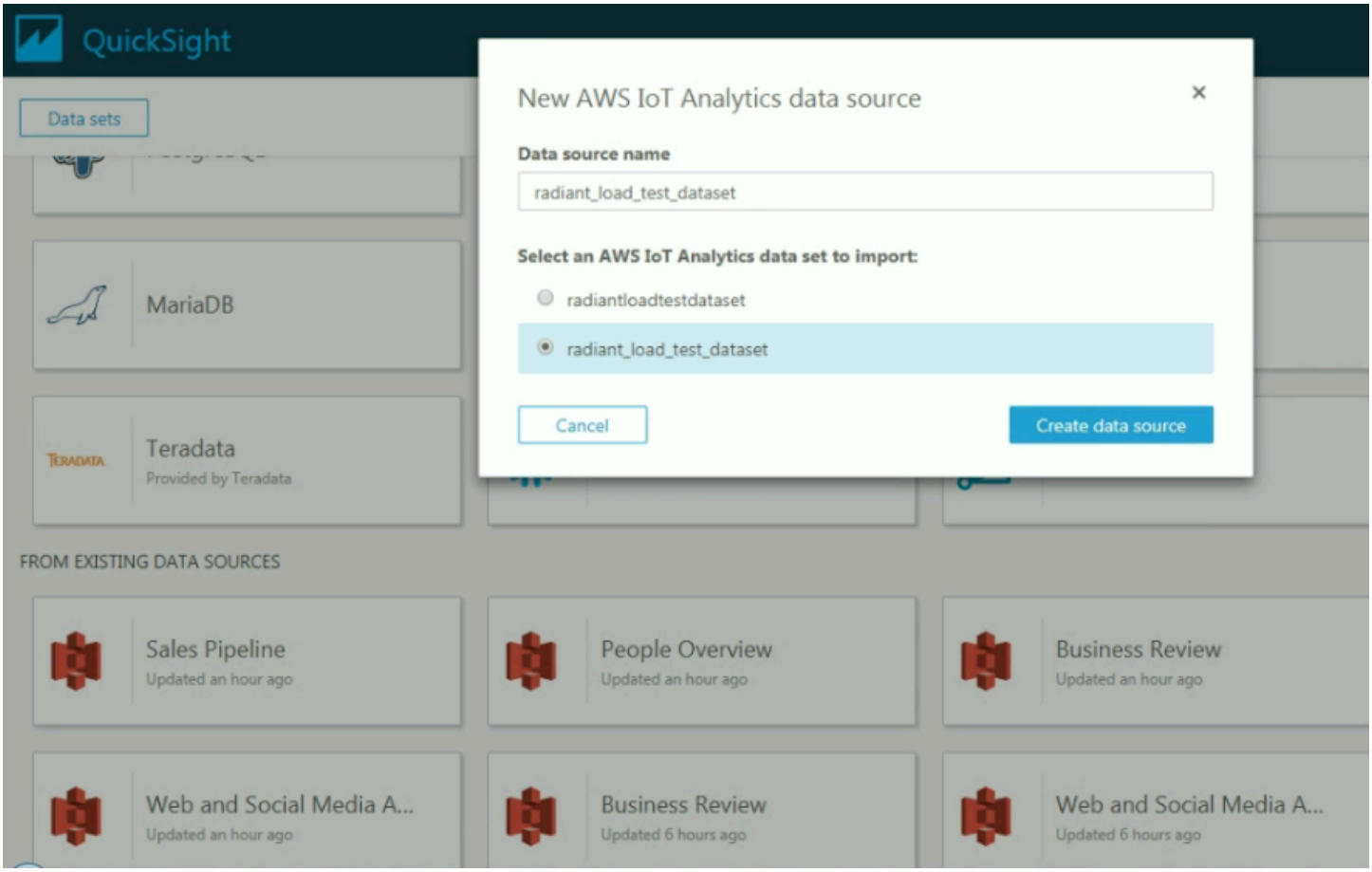
Visualización de datos de AWS IoT Analytics con Amazon QuickSight

AWS IoT Analytics ofrece integración directa con Amazon QuickSight. Amazon QuickSight es un rápido servicio de análisis empresariales que puede utilizar para crear visualizaciones, realizar análisis ad hoc y obtener rápidamente información empresarial útil a partir de sus datos. Amazon QuickSight permite a las organizaciones escalar de cientos a miles de usuarios y ofrece un rendimiento fiable gracias a su sólido motor en memoria (SPICE). Puede seleccionar conjuntos de datos de AWS IoT Analytics en la consola de Amazon QuickSight y comenzar a crear paneles y visualizaciones. Amazon QuickSight está disponible en estas regiones.

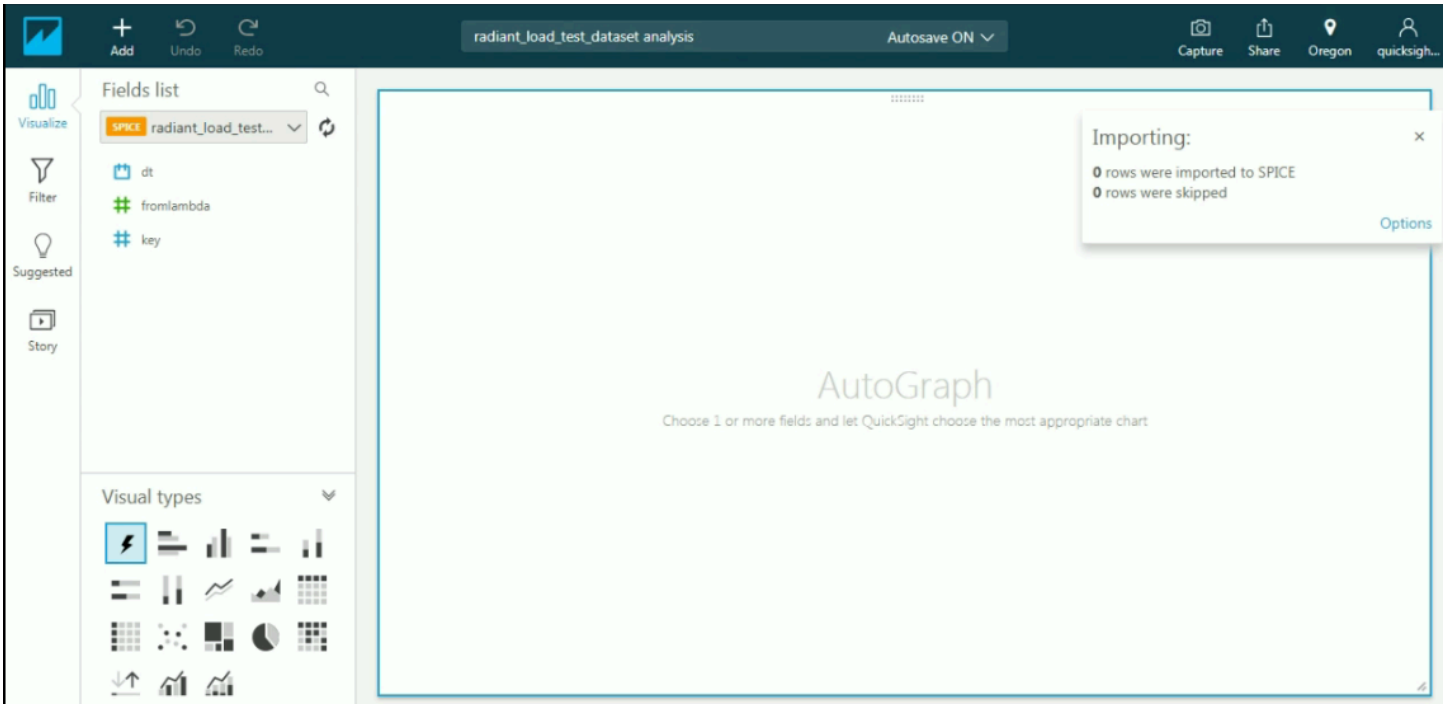
Para comenzar con sus visualizaciones de Amazon QuickSight, debe crear una cuenta de Amazon QuickSight. Asegúrese de dar acceso a Amazon QuickSight a sus datos de AWS IoT Analytics al configurar su cuenta. Si ya tiene una cuenta, proporcione a Amazon QuickSight acceso a sus datos de AWS IoT Analytics seleccionando Administrador, Administrar QuickSight, Seguridad y permisos. En Acceso de QuickSight a los servicios de AWS, seleccione Añadir o quitar y, a continuación, seleccione la casilla situada junto a AWS IoT Analytics y seleccione Actualizar.



Una vez configurada su cuenta, en la página de la consola de administración de Amazon QuickSight elija Nuevo análisis y Nuevo conjunto de datos, y después elija AWS IoT Analytics como origen. Escriba un nombre para su origen de datos, seleccione un conjunto de datos para importar y, a continuación, seleccione Crear origen de datos.



Una vez creado el origen de datos, puede crear visualizaciones en Amazon QuickSight.



Para obtener información sobre los paneles y conjuntos de datos de Amazon QuickSight, consulte la documentación de Amazon QuickSight.

Etiquetado de los recursos de AWS IoT Analytics

Para ayudarle a administrar los canales, los conjuntos de datos, los almacenes de datos y las canalizaciones, opcionalmente puede asignar sus propios metadatos a cada uno de estos recursos en forma de etiquetas. Este capítulo describe qué son las etiquetas y cómo crearlas.

Temas

Conceptos básicos de etiquetas

Las etiquetas le permiten clasificar los recursos de AWS IoT Analytics de diversas maneras, por ejemplo, según su finalidad, propietario o entorno. Esto es útil cuando tiene muchos recursos del mismo tipo, ya que puede identificar rápidamente un recurso específico en función de las etiquetas que le haya asignado. Cada etiqueta está formada por una clave y un valor opcional, ambos definidos por el usuario. Por ejemplo, podría definir un conjunto de etiquetas para los canales que le ayuden a realizar un seguimiento del tipo de dispositivo responsable del origen de los mensajes de cada canal. Recomendamos que idee un conjunto de claves de etiqueta que cumpla sus necesidades para cada tipo de recurso. Mediante el uso de un conjunto coherente de claves de etiquetas, podrá administrar los recursos más fácilmente. Puede buscar y filtrar los recursos en función de las etiquetas que agregue.

También puede utilizar etiquetas para categorizar y realizar un seguimiento de los costos. Cuando se aplican etiquetas a los canales, los conjuntos de datos, los almacenes de datos o las canalizaciones, AWS genera un informe de asignación de costos en forma de archivo CSV (valores separados por comas) con el total del uso y los costos por etiqueta. Puede aplicar etiquetas que representen categorías de negocio (p. ej., centros de costos, nombres de aplicación o propietarios) para estructurar los costos entre diferentes servicios. Para obtener más información sobre el uso de etiquetas para la asignación de costos, consulte Uso de etiquetas de asignación de costos en la Guía del usuario de AWS Billing.

Para facilitar el uso, utilice Editor de etiquetas en la consola de AWS Billing and Cost Management, que proporciona un método unificado y centralizado para crear y administrar etiquetas. Para obtener más información, consulte Uso de Tag Editor en Introducción a la AWS Management Console.

También puede trabajar con etiquetas utilizando la AWS CLI y la API de AWS IoT Analytics. Puede asociar etiquetas con canales, conjuntos de datos, almacenes de datos y canalizaciones al crearlos; utilice el campo Tags en los siguientes comandos:

- CreateChannel
- CreateDataset
- CreateDatastore
- CreatePipeline

Puede añadir, modificar o eliminar etiquetas para recursos existentes que admitan el uso de etiquetas. Use los siguientes comandos:

- TagResource
- ListTagsForResource
- UntagResource

Puede editar las claves y los valores de las etiquetas y también puede eliminar etiquetas de un recurso en cualquier momento. Puede establecer el valor de una etiqueta como una cadena vacía, pero no puede asignarle un valor nulo. Si añade una etiqueta con la misma clave que una etiqueta existente en ese recurso, el nuevo valor sobrescribirá al antiguo. Si elimina un recurso, también se eliminarán todas las etiquetas que este tenga asociadas.

Uso de etiquetas con políticas de IAM

Puede utilizar el elemento `Condition` (también llamado bloque `Condition`) junto con las siguientes claves/valores de contexto de condición en una política de IAM para controlar el acceso del usuario (permiso) en función de las etiquetas de un usuario:

- Utilice `iotanalytics:ResourceTag/<tag-key>: <tag-value>` para permitir o denegar acciones de los usuarios en recursos con etiquetas específicas.
- Utilice `aws:RequestTag/<tag-key>: <tag-value>` para exigir (o impedir) el uso de una etiqueta específica al realizar una solicitud de API para crear o modificar un recurso que permita etiquetas.
- Utilice `aws:TagKeys: [<tag-key>, ...]` para exigir (o impedir) el uso de un conjunto de claves de etiquetas al realizar una solicitud de API para crear o modificar un recurso que permita etiquetas.

> **Note**
>
> Las claves de contexto de condición y los valores de una política de IAM se aplican únicamente a las acciones de AWS IoT Analytics en las que un identificador de un recurso que se puede etiquetar es un parámetro obligatorio. Por ejemplo, el uso de DescribeLoggingOptions no está permitido/denegado según las claves/vales de contexto de la condición porque no se hace referencia a ningún recurso etiquetable (canal, conjunto de datos, almacén de datos o canalización) en esta solicitud.

Para obtener más información, consulte Control del acceso mediante etiquetas en la Guía del usuario de IAM. La sección Referencia de políticas JSON de IAM de esta guía incluye sintaxis, descripciones y ejemplos detallados de los elementos, variables y lógica de evaluación de las políticas JSON de IAM.

La siguiente política de ejemplo aplica dos restricciones basadas en etiquetas. Un usuario restringido por esta política:

1. No se puede dar a un recurso la etiqueta "env=prod" (consulte la línea `"aws:RequestTag/env" : "prod"` en el ejemplo).
2. No se puede modificar u obtener acceso a un recurso que tiene una etiqueta existente "env=prod" (consulte la línea `"iotanalytics:ResourceTag/env" : "prod"` en el ejemplo).

```
{
  "Version" : "2012-10-17",
  "Statement" :
  [
    {
      "Effect" : "Deny",
      "Action" : "iotanalytics:*",
      "Resource" : "*",
      "Condition" : {
        "StringEquals" : {
          "aws:RequestTag/env" : "prod"
        }
      }
    },
    {
```

```
      "Effect" : "Deny",
      "Action" : "iotanalytics:*",
      "Resource" : "*",
      "Condition" : {
        "StringEquals" : {
          "iotanalytics:ResourceTag/env" : "prod"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": [
        "iotanalytics:*"
      ],
      "Resource": "*"
    }
  ]
}
```

También puede especificar varios valores de etiqueta para una determinada clave de etiqueta encerrándolos en una lista, como en el siguiente ejemplo.

```
"StringEquals" : {
  "iotanalytics:ResourceTag/env" : ["dev", "test"]
}
```

> **Note**
>
> Si permite/deniega a los usuarios acceso a recursos en función de etiquetas, es importante considerar denegar explícitamente a los usuarios la posibilidad de agregar estas etiquetas o retirarlas de los mismos recursos. De lo contrario, es posible que un usuario eluda sus restricciones y obtenga acceso a un recurso modificando sus etiquetas.

Restricciones de las etiquetas

Se aplican las siguientes restricciones básicas a las etiquetas:

- Número máximo de etiquetas por recurso: 50
- Longitud máxima de la clave: 127 caracteres Unicode en UTF-8

- Longitud máxima del valor: 255 caracteres Unicode en UTF-8
- Las claves y los valores de las etiquetas distinguen entre mayúsculas y minúsculas.
- No utilice el `aws: prefix` en los nombres o valores de las etiquetas, porque está reservado para uso de AWS. Los nombres 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 fuente.
- Si se utiliza su esquema de etiquetado en múltiples servicios y recursos, recuerde que otros servicios pueden tener otras restricciones sobre caracteres permitidos. Los caracteres generalmente permitidos son: letras, espacios y números representables en UTF-8, además de los siguientes caracteres especiales: + - = . _ : / @.

Expresiones SQL en AWS IoT Analytics

Los conjuntos de datos se generan mediante expresiones SQL en los datos de un almacén de datos. AWS IoT Analytics utiliza las mismas consultas, funciones y operadores de SQL que Amazon Athena.

AWS IoT Analytics admite un subconjunto de sintaxis SQL en estándar ANSI.

```
SELECT [ ALL | DISTINCT ] select_expression [, ...]
[ FROM from_item [, ...] ]
[[ INNER | OUTER ] LEFT | RIGHT | FULL | CROSS JOIN join_item [ ON join_condition ]]
[ WHERE condition ]
[ GROUP BY [ ALL | DISTINCT ] grouping_element [, ...] ]
[ HAVING condition ]
[ UNION [ ALL | DISTINCT ] union_query ]
[ ORDER BY expression [ ASC | DESC ] [ NULLS FIRST | NULLS LAST] [, ...] ]
[ LIMIT [ count | ALL ] ]
```

Para obtener una descripción de los parámetros, consulte Parámetros en la documentación de Amazon Athena.

AWS IoT Analytics y Amazon Athena no admiten lo siguiente:

- Cláusulas de `WITH`.
- Instrucciones `CREATE TABLE AS SELECT`
- Instrucciones `INSERT INTO`
- Instrucciones preparadas, no puede ejecutar `EXECUTE` con `USING`.
- `CREATE TABLE LIKE`
- `DESCRIBE INPUT` y `DESCRIBE OUTPUT`
- Instrucciones `EXPLAIN`
- Funciones definidas por el usuario (UDF o UDAF)
- Procedimientos almacenados
- Conectores federados

Temas

- Funcionalidades de SQL compatibles con AWS IoT Analytics

- Solución de problemas comunes con las consultas SQL en AWS IoT Analytics

Funcionalidades de SQL compatibles con AWS IoT Analytics

Los conjuntos de datos se generan utilizando expresiones SQL sobre los datos de un almacén de datos. Las consultas que ejecuta en AWS IoT Analytics se basan en Presto 0.217.

Tipos de datos compatibles

AWS IoT Analytics y Amazon Athena admiten estos tipos de datos.

- primitive_type
  - `TINYINT`
  - `SMALLINT`
  - `INT`
  - `BIGINT`
  - `BOOLEAN`
  - `DOUBLE`
  - `FLOAT`
  - `STRING`
  - `TIMESTAMP`
  - `DECIMAL(precision, scale)`
  - `DATE`
  - `CHAR` (datos de caracteres de longitud fija con una longitud específica)
  - `VARCHAR` (datos de caracteres de longitud variable con una longitud específica)
- array_type
  - `ARRAY<data_type>`
- map_type
  - `MAP<primitive_type, data_type>`
- struct_type
  - `STRUCT<col_name:data_type[COMMENT col_comment][,...]>`

> **Note**
>
> AWS IoT Analytics y Amazon Athena no admiten algunos tipos de datos.

Funciones compatibles

Las funcionalidades de SQL de Amazon Athena y AWS IoT Analytics se basan en Presto 0.217. Para obtener información sobre funciones, operadores y expresiones relacionadas, consulte Funciones y operadores y las siguientes secciones específicas de la documentación de Presto.

- Logical operators (Operadores lógicos)
- Comparison functions and operators (Funciones y operadores de comparación)
- Conditional expressions (Expresiones condicionales)
- Conversion functions (Funciones de conversión)
- Mathematical functions and operators (Funciones y operadores matemáticos)
- Bitwise functions (Funciones Bitwise)
- Decimal functions and operators (Funciones y operadores decimales)
- String functions and operators (Funciones y operadores de cadena)
- Binary functions (Funciones binarias)
- Date and time functions and operators (Funciones y operadores de fecha y hora)
- Regular expression functions (Funciones de expresión regular)
- JSON functions and operators (Funciones y operadores JSON)
- URL functions (Funciones de URL)
- Aggregate functions (Funciones de agregación)
- Window functions (Funciones de ventana)
- Color functions (Funciones de color)
- Array functions and operators (Funciones y operadores de matriz)
- Map functions and operators (Funciones y operadores de mapas)
- Lambda expressions and functions (Expresiones y funciones de Lambda)
- Teradata functions (Funciones de teradatos)

> **Note**
>
> AWS IoT Analytics y Amazon Athena no admiten funciones definidas por el usuario (UDF o UDAF) ni procedimientos almacenados.

Solución de problemas comunes con las consultas SQL en AWS IoT Analytics

Utilice la siguiente información para solucionar problemas con sus consultas SQL en AWS IoT Analytics.

- Para evitar una comilla simple, preceda a esta de otra comilla simple. No confunda esto con una comilla doble.

  Example Ejemplo

  ```
  SELECT 'O''Reilly'
  ```

- Para evitar los guiones bajos, utilice puntos suspensivos para encerrar los nombres de columna del almacén de datos que comiencen con un guion bajo.

  Example Ejemplo

  ```
  SELECT `_myMessageAttribute` FROM myDataStore
  ```

- Para evitar usar nombres con números, escriba entre comillas dobles los nombres de los almacenes de datos que incluyan números.

  Example Ejemplo

  ```
  SELECT * FROM "myDataStore123"
  ```

- Para evitar las palabras clave reservadas, escriba entre comillas dobles las palabras clave reservadas. Para obtener más información, consulte la Lista de palabras clave reservadas en Instrucciones SQL SELECT.

Seguridad en AWS IoT Analytics

La seguridad en la nube AWS es la máxima prioridad. Como AWS cliente, usted se beneficia de una arquitectura de centro de datos y red diseñada para cumplir con los requisitos de las organizaciones más sensibles a la seguridad.

La seguridad es una responsabilidad compartida entre usted 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 AWS los servicios en la AWS nube. AWS también le proporciona servicios que puede utilizar de forma segura. Auditores externos prueban y verifican periódicamente la eficacia de nuestra seguridad en el marco de los programas de conformidad de AWS. Para obtener más información sobre los programas de cumplimiento aplicables AWS IoT Analytics, consulte AWS los servicios clasificados por programa de cumplimiento.
- Seguridad en la nube: su responsabilidad viene determinada por el AWS servicio que utilice. Usted 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 ayudará a entender cómo aplicar el modelo de responsabilidad compartida cuando lo utilice AWS IoT Analytics. Los siguientes temas muestran cómo configurarlo AWS IoT Analytics para cumplir sus objetivos de seguridad y conformidad. También aprenderá a utilizar otros AWS servicios que pueden ayudarle a supervisar y proteger sus AWS IoT Analytics recursos.

AWS Identity and Access Management en AWS IoT Analytics

AWS Identity and Access Management (IAM) es un AWS servicio que ayuda al administrador a controlar de forma segura el acceso a los AWS recursos. Los administradores de IAM controlan quién puede autenticarse (iniciar sesión) y quién puede autorizarse (tener permisos) para usar los recursos. AWS IoT Analytics La IAM es un AWS servicio que puede utilizar sin coste adicional.

Público

La forma de usar AWS Identity and Access Management (IAM) varía según el trabajo en el que se desempeñe. AWS IoT Analytics

Usuario del servicio: si utiliza el AWS IoT Analytics servicio para realizar su trabajo, el administrador le proporcionará las credenciales y los permisos que necesita. A medida que vaya utilizando más AWS IoT Analytics funciones para realizar su trabajo, es posible que 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 AWS IoT Analytics, consulte Solución de problemas AWS IoT Analytics de identidad y acceso.

Administrador de servicios: si estás a cargo de AWS IoT Analytics los recursos de tu empresa, probablemente tengas acceso total a ellos AWS IoT Analytics. Su trabajo consiste en determinar a qué AWS IoT Analytics funciones y recursos 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 la IAM AWS IoT Analytics, consulte¿Cómo AWS IoT Analytics funciona con IAM.

Administrador de IAM: si es un administrador de IAM, es posible que quiera conocer más detalles sobre cómo escribir políticas para administrar el acceso a AWS. Para ver ejemplos de políticas AWS IoT Analytics basadas en la identidad que puede utilizar en IAM, consulte. AWS IoT Analytics ejemplos de políticas basadas en la identidad

Autenticación con identidades

La autenticación es la forma de iniciar sesión AWS con sus credenciales de identidad. Debe estar autenticado (con quien haya iniciado sesión AWS) como usuario de IAM o asumiendo una función de IAM. Usuario raíz de la cuenta de AWS

Puede iniciar sesión AWS como una identidad federada mediante las credenciales proporcionadas a través de una fuente de identidad. AWS IAM Identity Center Los usuarios (IAM Identity Center), 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 accedes AWS mediante la federación, estás asumiendo un rol de forma indirecta.

Según el tipo de usuario que sea, puede iniciar sesión en el portal AWS Management Console o en el de AWS acceso. Para obtener más información sobre cómo iniciar sesión AWS, consulte Cómo iniciar sesión Cuenta de AWS en su Guía del AWS Sign-In usuario.

Si accede AWS mediante programación, AWS proporciona un kit de desarrollo de software (SDK) y una interfaz de línea de comandos (CLI) para firmar criptográficamente sus solicitudes con sus

credenciales. Si no utilizas AWS herramientas, debes firmar las solicitudes tú mismo. Para obtener más información sobre cómo usar el método recomendado para firmar las solicitudes usted mismo, consulte Firmar las solicitudes de la AWS API 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, le AWS recomienda que utilice 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.

Cuenta de AWS usuario root

Al crear una Cuenta de AWS, comienza con una identidad de inicio de sesión que tiene acceso completo a todos Servicios de AWS los recursos de la cuenta. Esta identidad se denomina usuario Cuenta de AWS raíz y se accede a ella iniciando sesión con la dirección de correo electrónico y la contraseña que utilizaste 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.

Usuarios y grupos de IAM

Un usuario de IAM es una identidad propia Cuenta de AWS que tiene 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 dentro de usted Cuenta de AWS que tiene permisos específicos. Es similar a un usuario de IAM, pero no está asociado a una determinada persona. Puede asumir temporalmente una función de IAM en el AWS Management Console cambiando de función. Puede asumir un rol llamando a una operación de AWS API AWS CLI 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. Sin embargo, con algunas Servicios de AWS, puedes adjuntar una política directamente a un recurso (en lugar de usar un rol como proxy). 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 funciones en 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 al servicio.

- Sesiones de acceso directo (FAS): cuando utilizas un usuario o un rol de IAM para realizar acciones en ellas AWS, se te considera director. Cuando utiliza algunos servicios, es posible que realice una acción que desencadene otra acción en un servicio diferente. El FAS utiliza los permisos del principal que llama Servicio de AWS y los solicita Servicio de AWS para realizar solicitudes a los servicios descendentes. Las solicitudes de FAS solo se realizan cuando un servicio recibe una solicitud que requiere interacciones con otros Servicios de AWS 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.
- Función vinculada al servicio: una función vinculada a un servicio es un tipo de función de servicio que está vinculada a un. Servicio de AWS El servicio puede asumir el rol para realizar una acción en su nombre. Los roles vinculados al servicio aparecen en usted 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 usar un rol de IAM para administrar las credenciales temporales de las aplicaciones que se ejecutan en una instancia EC2 y realizan AWS CLI solicitudes a la API. AWS Es preferible hacerlo de este modo a almacenar claves de acceso en la instancia de EC2. Para asignar un AWS rol a una instancia EC2 y ponerlo a disposición de todas sus aplicaciones, debe crear un perfil de instancia adjunto 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

El acceso se controla AWS creando políticas y adjuntándolas a AWS identidades o recursos. Una política es un objeto AWS que, cuando se asocia a una identidad o un recurso, define sus permisos. AWS evalúa estas políticas cuando un director (usuario, usuario raíz o sesión de rol) realiza una solicitud. Los permisos en las políticas determinan si la solicitud se permite o se deniega. La mayoría de las políticas se almacenan 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 usar las políticas de AWS JSON 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 esa política puede obtener información sobre el rol de la API AWS Management Console AWS CLI, la o la AWS API.

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 empresa. Las políticas administradas incluyen políticas AWS administradas y 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.

Otros tipos de políticas

AWS admite tipos de políticas adicionales y menos comunes. 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 servicios (SCP): las SCP son políticas de JSON que especifican los permisos máximos para una organización o unidad organizativa (OU). AWS Organizations AWS Organizations es un servicio para agrupar y gestionar de forma centralizada varios de los Cuentas de AWS que son propiedad de su empresa. Si habilita todas las características en una organización, entonces podrá aplicar políticas de control de servicio (SCP) a una o a todas sus cuentas. El SCP limita los permisos de las entidades en las cuentas de los miembros, incluidas las de cada una. 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 saber cómo AWS determinar si se debe permitir una solicitud cuando se trata de varios tipos de políticas, consulte la lógica de evaluación de políticas en la Guía del usuario de IAM.

¿Cómo AWS IoT Analytics funciona con IAM

Antes de utilizar IAM para gestionar el acceso AWS IoT Analytics, debe comprender las funciones de IAM disponibles para su uso. AWS IoT AnalyticsPara obtener una visión general de cómo funcionan con IAM AWS IoT Analytics y otros AWS servicios, consulte los AWS servicios que funcionan con IAM en la Guía del usuario de IAM.

Temas en esta página:

AWS IoT Analytics políticas basadas en la identidad

Con las políticas de IAM basadas en la identidad, puede especificar las acciones y los recursos permitidos o denegados y las condiciones en las que se permiten o deniegan las acciones. AWS IoT Analytics admite acciones, recursos y claves de condición específicos. Para obtener más información acerca de los elementos que utiliza en una política de JSON, consulte Referencia de los elementos de las políticas de JSON de IAM en la Guía del usuario de IAM.

Acciones

El elemento `Action` de una política basada en identidad de IAM describe la acción o las acciones específicas que la política permitirá o denegará. Las acciones políticas suelen tener el mismo nombre que la operación de AWS API asociada. Las acciones se utilizan en una política para conceder permisos y así llevar a cabo la operación asociada.

La acción política AWS IoT Analytics utiliza el siguiente prefijo antes de la acción: `iotanalytics:` por ejemplo, para conceder a alguien permiso para crear un AWS IoT Analytics canal con la operación de la AWS IoT Analytics `CreateChannel` API, debes incluir la `iotanalytics:BatchPuMessage` acción en su política. Las declaraciones de política deben incluir un `NotAction` elemento `Action` o. AWS IoT Analytics define su propio conjunto de acciones que describen las tareas que puede realizar con este servicio.

Para especificar varias acciones de en una única instrucción, sepárelas con comas del siguiente modo.

```
"Action": [
    "iotanalytics:action1",
    "iotanalytics: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": "iotanalytics:Describe*"
```

Para ver una lista de AWS IoT Analytics acciones, consulte las acciones definidas AWS IoT Analytics en la Guía del usuario de IAM.

Recursos

El elemento `Resource` especifica el objeto u objetos a los que se aplica la acción. Las instrucciones deben contener un elemento `Resource` o `NotResource`. Especifique un recurso con un ARN o el carácter comodín (*) para indicar que la instrucción se aplica a todos los recursos.

El recurso del AWS IoT Analytics conjunto de datos tiene el siguiente ARN.

```
arn:${Partition}:iotanalytics:${Region}:${Account}:dataset/${DatasetName}
```

Para obtener más información acerca del formato de los ARN, consulte Nombres de recursos de Amazon (ARN) y espacios de nombres de servicios de AWS.

Por ejemplo, para especificar el conjunto de datos `Foobar` en su instrucción, utilice el siguiente ARN.

```
"Resource": "arn:aws:iotanalytics:us-east-1:123456789012:dataset/Foobar"
```

Para especificar todas las instancias que pertenecen a una cuenta específica, utilice el carácter comodín (*).

```
"Resource": "arn:aws:iotanalytics:us-east-1:123456789012:dataset/*"
```

Algunas AWS IoT Analytics acciones, como las de creación de recursos, no se pueden realizar en un recurso específico. En dichos casos, debe utilizar el carácter comodín (*).

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

Algunas acciones AWS IoT Analytics de la API implican varios recursos. Por ejemplo, `CreatePipeline` hace referencia tanto a un canal como a un conjunto de datos, por lo que un usuario debe tener permisos para utilizar el canal y el conjunto de datos. Para especificar varios recursos en una única instrucción, separe los ARN con comas.

```
"Resource": [
    "resource1",
    "resource2"
     ]
```

Para ver una lista de los tipos de AWS IoT Analytics recursos y sus ARN, consulte los recursos definidos AWS IoT Analytics en la Guía del usuario de IAM. Para obtener información sobre las acciones con las que puede especificar el ARN de cada recurso, consulte Acciones definidas por AWS IoT Analytics.

Claves de condición

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 utilizan operadores de condición, tales como igual o menor que, para que coincida la condición de la política con 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 a un usuario permiso para acceder a un recurso solo si está etiquetado con su nombre de usuario. Para obtener más información, consulte Elementos de la política de IAM: Variables y etiquetas en la Guía del usuario de IAM.

AWS IoT Analytics no proporciona ninguna clave de condición específica del servicio, pero sí admite el uso de algunas claves de condición globales. Para ver todas las claves de condición AWS globales, consulte las claves de contexto de condición AWS globales en la Guía del usuario de IAM.

Ejemplos

Para ver ejemplos de políticas AWS IoT Analytics basadas en la identidad, consulte. AWS IoT Analytics ejemplos de políticas basadas en la identidad

AWS IoT Analytics políticas basadas en recursos

AWS IoT Analytics no admite políticas basadas en recursos. Para ver un ejemplo de una página detallada de políticas basadas en recursos, consulte Uso de políticas basadas en recursos AWS Lambda en la Guía del desarrollador de AWS Lambda .

Autorización basada en etiquetas AWS IoT Analytics

Puede adjuntar etiquetas a AWS IoT Analytics los recursos o pasarles etiquetas en una solicitud AWS IoT Analytics. Para controlar el acceso en función de las etiquetas, proporcione información sobre etiquetas en el elemento de condición de una política mediante las claves de condición `iotanalytics:ResourceTag/{key-name}`, `aws:RequestTag/{key-name}` o `aws:TagKeys`. Para obtener más información sobre cómo etiquetar AWS IoT Analytics los recursos, consulta Cómo etiquetar AWS IoT Analytics los recursos.

Para ver un ejemplo de política basada en la identidad para limitar el acceso a un recurso en función de las etiquetas de ese recurso, consulte Visualización de AWS IoT Analytics canales en función de las etiquetas.

AWS IoT Analytics Funciones de IAM

Un rol de IAM es una entidad de la Cuenta de AWS que dispone de permisos específicos.

Usar una credencial temporal con AWS IoT Analytics

Puede utilizar credenciales temporales para iniciar sesión con federación, asumir un rol de IAM o asumir un rol de acceso entre cuentas. Las credenciales de seguridad temporales se obtienen llamando a AWS Security Token Service (AWS STS) operaciones de la API, como AssumeRoleo GetFederationToken.

AWS IoT Analytics no admite el uso de credenciales temporales.

Roles vinculados al servicio

Los roles relacionados con el AWS servicio permiten al servicio acceder a los recursos de otros servicios para completar una acción en tu nombre. Los roles vinculados a servicios aparecen en la cuenta de IAM y son propiedad del servicio. Un administrador de IAM puede ver, pero no editar, los permisos de los roles vinculados a servicios.

AWS IoT Analytics no admite funciones vinculadas al servicio.

Roles de servicio

Esta característica permite que un servicio asuma un rol de servicio en su nombre. Este rol permite que el servicio obtenga acceso a los recursos de otros servicios para completar una acción en su nombre. Los roles de servicio aparecen en su cuenta de IAM y son propiedad de la cuenta. Esto significa que un administrador de IAM puede cambiar los permisos de este rol. Sin embargo, hacerlo podría deteriorar la funcionalidad del servicio.

AWS IoT Analytics admite funciones de servicio.

Prevención del suplente confuso entre servicios

El problema del suplente confuso es un problema de seguridad en el que una entidad que no tiene permiso para realizar una acción puede obligar a una entidad con más privilegios a realizar la acción. En AWS, la suplantación entre servicios puede dar lugar al problema del suplente confuso. La suplantación entre servicios puede producirse cuando un servicio (el servicio que lleva a cabo las llamadas) llama a otro servicio (el servicio al que se llama). El servicio que lleva a cabo las llamadas se puede manipular para utilizar sus permisos a fin de actuar en función de los recursos de otro cliente de una manera en la que no debe tener permiso para acceder. Para evitarlo, AWS proporciona herramientas que lo ayudan a proteger sus datos para todos los servicios con entidades principales de servicio a las que se les ha dado acceso a los recursos de su cuenta.

Recomendamos utilizar las claves de contexto de condición global de `aws:SourceArn` y `aws:SourceAccount` en las políticas de recursos. Esto limita los permisos que AWS IoT Analytics otorga a otro servicio para el recurso. Si se utilizan ambas claves de contexto de condición global, el valor `aws:SourceAccount` y la cuenta del valor `aws:SourceArn` deben utilizar el mismo ID de cuenta cuando se utilicen en la misma declaración de política.

La forma más eficaz de protegerse contra el problema del suplente confuso es utilizar la clave de contexto de condición global de `aws:SourceArn` con el nombre de recurso de Amazon (ARN) completo del recurso. Si no conoce el ARN completo del recurso o si especifica varios recursos, utilice la clave de condición de contexto global `aws:SourceArn` con comodines (*) para las partes desconocidas del ARN. Por ejemplo, `arn:aws:iotanalytics::123456789012:*`.

Temas

- Prevención de los buckets de Amazon S3
- Prevención con Registros de Amazon CloudWatch
- Prevención del suplente confuso en los recursos de AWS IoT Analytics administrados por el cliente

Prevención de los buckets de Amazon S3

Si utiliza el almacenamiento Amazon S3 gestionado por el cliente para su almacén de datos de AWS IoT Analytics, el bucket de Amazon S3 que almacena sus datos podría estar expuesto a problemas de suplentes confusos.

Por ejemplo, Nikki Wolf usa un bucket de Amazon S3 de propiedad del cliente llamado *DOC-EXAMPLE-BUCKET*. El bucket almacena información para un almacén de datos de AWS IoT Analytics que se creó en la región *us-east-1*. Especifica una política que permite a la entidad principal del servicio de AWS IoT Analytics consultar *DOC-EXAMPLE-BUCKET* en su nombre. La compañera de trabajo de Nikki, Li Juan, consulta *DOC-EXAMPLE-BUCKET* desde su propia cuenta y crea un conjunto de datos con los resultados. Como resultado, la entidad principal del servicio de AWS IoT Analytics consultó el bucket de Amazon S3 de Nikki en nombre de Li, a pesar de que Li ejecutó la consulta desde su cuenta.

Para evitar esto, Nikki puede especificar la condición `aws:SourceAccount` o la condición `aws:SourceArn` en la política para *DOC-EXAMPLE-BUCKET*.

Especificar la condición **`aws:SourceAccount`**: el siguiente ejemplo de una política de bucket especifica que solo los recursos de AWS IoT Analytics de la cuenta de Nikki (*123456789012*) pueden acceder a *DOC-EXAMPLE-BUCKET*.

```
{
    "Version": "2012-10-17",
    "Id": "MyPolicyID",
    "Statement": [
        {
            "Sid": "ConfusedDeputyPreventionExamplePolicy",
            "Effect": "Allow",
            "Principal": {
                "Service": "iotanalytics.amazonaws.com"
            },
            "Action": [
                "s3:GetBucketLocation",
                "s3:GetObject",
                "s3:ListBucket",
                "s3:ListBucketMultipartUploads",
                "s3:ListMultipartUploadParts",
                "s3:AbortMultipartUpload",
                "s3:PutObject",
                "s3:DeleteObject"
```

```
            ],
            "Resource": [
                "arn:aws:s3:::DOC-EXAMPLE-BUCKET",
                "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"
            ],
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "123456789012"
                }
            }
        }
    ]
}
```

Especificar la condición **`aws:SourceArn`**: como alternativa, Nikki puede usar la condición `aws:SourceArn`.

```
{
    "Version": "2012-10-17",
    "Id": "MyPolicyID",
    "Statement": [
        {
            "Sid": "ConfusedDeputyPreventionExamplePolicy",
            "Effect": "Allow",
            "Principal": {
                "Service": "iotanalytics.amazonaws.com"
            },
            "Action": [
                "s3:GetBucketLocation",
                "s3:GetObject",
                "s3:ListBucket",
                "s3:ListBucketMultipartUploads",
                "s3:ListMultipartUploadParts",
                "s3:AbortMultipartUpload",
                "s3:PutObject",
                "s3:DeleteObject"
            ],
            "Resource": [
                "arn:aws:s3:::DOC-EXAMPLE-BUCKET",
                "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"
            ],
            "Condition": {
                "ArnLike": {
```

```
                    "aws:SourceArn": [
                        "arn:aws:iotanalytics:us-east-1:123456789012:dataset/DOC-
EXAMPLE-DATASET",
                        "arn:aws:iotanalytics:us-east-1:123456789012:datastore/DOC-
EXAMPLE-DATASTORE"
                    ]
                }
            }
        }
    ]
}
```

Prevención con Registros de Amazon CloudWatch

Al monitorizar con Registros de Amazon CloudWatch puede evitar el problema de suplentes confusos. La siguiente política de recursos muestra cómo evitar el problema de suplente confuso con:

- Clave de contexto de condición global, `aws:SourceArn`
- La `aws:SourceAccount` con su ID de cuenta de AWS;
- El recurso de cliente asociado a la solicitud de `sts:AssumeRole` en AWS IoT Analytics.

En el siguiente ejemplo, sustituya *123456789012* por su ID de cuenta de AWS y *us-east-1* por la región de su cuenta de AWS IoT Analytics.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "iotanalytics.amazonaws.com"
            },
            "Action": "logs:PutLogEvents",
            "Resource": "*",
            "Condition":{
                "ArnLike":{
                    "aws:SourceArn":"arn:aws:iotanalytics:us-east-1:123456789012:*/*"
                },
                "StringEquals":{
                    "aws:SourceAccount":"123456789012"
```

```
                }
            }
    ]
}
```

Para obtener más información acerca de la habilitación y configuración de Registros de Amazon CloudWatch, consulte the section called “Registro y monitoreo”.

Prevención del suplente confuso en los recursos de AWS IoT Analytics administrados por el cliente

Si concede permiso a AWS IoT Analytics para que realice acciones con sus recursos de AWS IoT Analytics, es posible que los recursos se vean expuestos a problemas de suplentes confusos. Para evitar el problema de suplentes confusos, puede limitar los permisos que se conceden a AWS IoT Analytics con las siguientes políticas de recursos de ejemplo.

Temas

Prevención en canales y almacenes de datos de AWS IoT Analytics

Utilice los roles de IAM para controlar los recursos de AWS a los que AWS IoT Analytics puede acceder en su nombre. Para evitar exponer su rol al problema de suplentes confusos, puede especificar la cuenta de AWS en el elemento `aws:SourceAccount` y el ARN del recurso de AWS IoT Analytics en el elemento `aws:SourceArn` de la política de confianza que asocie a un rol.

En el siguiente ejemplo, sustituya *123456789012* por su ID de cuenta de AWS y *arn:aws:iotanalytics:región de AWS:123456789012:channel/DOC-EXAMPLE-CHANNEL* por el ARN de un canal o almacén de datos de AWS IoT Analytics.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ConfusedDeputyPreventionExamplePolicy",
      "Effect": "Allow",
      "Principal": {
```

```
        "Service": "iotanalytics.amazonaws.com"
       },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "123456789012"
        },
        "ArnLike": {
          "aws:SourceArn": "arn:aws:iotanalytics:aws-region:123456789012:channel/DOC-
EXAMPLE-CHANNEL"
        }
      }
    }
  ]
}
```

Para obtener más información sobre las opciones de almacenamiento S3 administrado por el cliente para canales y almacenes de datos, consulte `CustomerManagedChannelS3Storage` y `CustomerManagedDatastoreS3Storage` en la Referencia de la API de AWS IoT Analytics.

Prevención del suplente confuso entre servicios para las reglas de entrega de contenido de conjunto de datos de AWS IoT Analytics

El rol de IAM que AWS IoT Analytics asume para entregar los resultados de las consultas del conjunto de datos a Amazon S3 o a AWS IoT Events puede estar expuesto a problemas de suplentes confusos. Para evitar el problema de suplentes confusos, especifique la cuenta de AWS en el elemento `aws:SourceAccount` y el ARN del recurso de AWS IoT Analytics en el elemento `aws:SourceArn` de la política de confianza que asocia a su rol.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ConfusedDeputyPreventionExampleTrustPolicyDocument",
      "Effect": "Allow",
      "Principal": {
        "Service": "iotanalytics.amazonaws.com"
       },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "123456789012"
```

```
        },
        "ArnLike": {
          "aws:SourceArn": "arn:aws:iotanalytics:aws-region:123456789012:dataset/DOC-
EXAMPLE-DATASET"
        }
      }
    }
  ]
}
```

Para obtener más información sobre la configuración de las reglas de entrega del contenido del conjunto de datos, consulte `contentDeliveryRules` en la Referencia de la API de AWS IoT Analytics.

AWS IoT Analytics ejemplos de políticas basadas en la identidad

De forma predeterminada, los usuarios y roles no tienen permiso para crear, ver ni modificar recursos de AWS IoT Analytics . Tampoco pueden realizar tareas con la API AWS Management Console AWS CLI, o AWS . Un administrador de IAM debe crear políticas de IAM que concedan permisos a los usuarios y a los roles para realizar operaciones de la API concretas en los recursos especificados que necesiten. El administrador debe asociar esas políticas a los usuarios o grupos que necesiten esos permisos.

Para obtener más información acerca de cómo crear una política basada en identidad de IAM con estos documentos de políticas de JSON de ejemplo, consulte Creación de políticas en la pestaña JSON en la Guía del usuario de IAM

Temas en esta página:

Prácticas recomendadas sobre las políticas

Las políticas basadas en identidad son muy eficaces. Determinan si alguien puede crear AWS IoT Analytics recursos de tu cuenta, acceder a ellos o eliminarlos. 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 a utilizar las políticas AWS gestionadas: para empezar a AWS IoT Analytics utilizarlas rápidamente, utilice las políticas AWS gestionadas para conceder a sus empleados los permisos que necesitan. Estas políticas ya están disponibles en su cuenta y son mantenidas y actualizadas por AWS. Para obtener más información, consulte Cómo empezar a usar permisos con políticas AWS administradas en la Guía del usuario de IAM.
- Conceder privilegios mínimos: al crear políticas personalizadas, conceda solo los permisos necesarios para llevar a cabo una tarea. Comience con un conjunto mínimo de permisos y conceda permisos adicionales según sea necesario. Por lo general, es más seguro que comenzar con permisos demasiado tolerantes e intentar hacerlos más estrictos más adelante. Para obtener más información, consulte Conceder privilegios mínimos en la Guía del usuario de IAM.
- Habilitar MFA para operaciones confidenciales: para mayor seguridad, obligue a los usuarios a utilizar la autenticación multifactor (MFA) para acceder a los recursos u operaciones confidenciales de la API. Para obtener más información, consulte Uso de la autenticación multifactor (MFA) en AWS en la Guía del usuario de IAM.
- Utilizar condiciones de política para mayor seguridad: en la medida en que sea práctico, defina las condiciones en las que las políticas basadas en identidad permitan el acceso a un recurso. Por ejemplo, puede escribir condiciones para especificar un rango de direcciones IP permitidas desde el que debe proceder una solicitud. También puede escribir condiciones para permitir solicitudes solo en un intervalo de hora o fecha especificado o para solicitar el uso de SSL o MFA. Para obtener más información, consulte Elementos de la política JSON de IAM: condición en la Guía del usuario de IAM.

Uso de la consola AWS IoT Analytics

Para acceder a la AWS IoT Analytics consola, debe tener un conjunto mínimo de permisos. Estos permisos deben permitirle enumerar y ver detalles sobre los AWS IoT Analytics recursos de su cuenta Cuenta de AWS. Si crea una política basada en identidad 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.

Para garantizar que esas entidades puedan seguir utilizando la AWS IoT Analytics consola, adjunte también la siguiente política AWS gestionada a las entidades. Para más información, consulte Adición de permisos a un usuario en la Guía del usuario de IAM:

```
{
     "Version": "2012-10-17",
     "Statement": [
         {
             "Effect": "Allow",
             "Action": [
                 "iotanalytics:BatchPutMessage",
                 "iotanalytics:CancelPipelineReprocessing",
                 "iotanalytics:CreateChannel",
                 "iotanalytics:CreateDataset",
                 "iotanalytics:CreateDatasetContent",
                 "iotanalytics:CreateDatastore",
                 "iotanalytics:CreatePipeline",
                 "iotanalytics:DeleteChannel",
                 "iotanalytics:DeleteDataset",
                 "iotanalytics:DeleteDatasetContent",
                 "iotanalytics:DeleteDatastore",
                 "iotanalytics:DeletePipeline",
                 "iotanalytics:DescribeChannel",
                 "iotanalytics:DescribeDataset",
                 "iotanalytics:DescribeDatastore",
                 "iotanalytics:DescribeLoggingOptions",
                 "iotanalytics:DescribePipeline",
                 "iotanalytics:GetDatasetContent",
                 "iotanalytics:ListChannels",
                 "iotanalytics:ListDatasetContents",
                 "iotanalytics:ListDatasets",
                 "iotanalytics:ListDatastores",
                 "iotanalytics:ListPipelines",
                 "iotanalytics:ListTagsForResource",
                 "iotanalytics:PutLoggingOptions",
                 "iotanalytics:RunPipelineActivity",
                 "iotanalytics:SampleChannelData",
                 "iotanalytics:StartPipelineReprocessing",
                 "iotanalytics:TagResource",
                 "iotanalytics:UntagResource",
                 "iotanalytics:UpdateChannel",
                 "iotanalytics:UpdateDataset",
                 "iotanalytics:UpdateDatastore",
                 "iotanalytics:UpdatePipeline"
             ],
             "Resource": "arn:${Partition}:iotanalytics:${Region}:${Account}:channel/
${channelName}",
```

```
            "Resource": "arn:${Partition}:iotanalytics:${Region}:${Account}:dataset/
${datasetName}",
            "Resource": "arn:${Partition}:iotanalytics:${Region}:${Account}:datastore/
${datastoreName}",
            "Resource": "arn:${Partition}:iotanalytics:${Region}:${Account}:pipeline/
${pipelineName}"
         }
     ]
 }
```

No es necesario conceder permisos mínimos de consola a los usuarios que solo realizan llamadas a la API AWS CLI o a la AWS API. En su lugar, permite acceso únicamente a las acciones que coincidan con la operación de API que intenta realizar.

Permitir a los usuarios consultar sus propios permisos

En este ejemplo, se muestra cómo podría crear una política que permita a los usuarios ver las políticas administradas e insertadas que se asocian a la identidad de sus usuarios. Esta política incluye permisos para completar esta acción en la consola o mediante programación mediante la API AWS CLI o 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": "*"
        }
    ]
}
```

Acceder a una entrada AWS IoT Analytics

En este ejemplo, quieres conceder a un usuario tu Cuenta de AWS acceso a uno de tus AWS IoT Analytics canales,`exampleChannel`. También desea permitir el uso para añadir, actualizar y eliminar canales.

La política concede los permisos `iotanalytics:ListChannels`, `iotanalytics:DescribeChannel`, `iotanalytics:CreateChannel`, `iotanalytics:DeleteChannel`, and `iotanalytics:UpdateChannel` al usuario. Para ver un tutorial de ejemplo para el servicio de Amazon S3 en el que se conceden permisos a los usuarios y se prueban con la consola, consulte Tutorial de ejemplo: uso de las políticas del usuario para controlar el acceso al bucket.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"ListChannelsInConsole",
         "Effect":"Allow",
         "Action":[
            "iotanalytics:ListChannels"
         ],
         "Resource":"arn:aws:iotanalytics:::*"
      },
      {
         "Sid":"ViewSpecificChannelInfo",
         "Effect":"Allow",
         "Action":[
```

```
            "iotanalytics:DescribeChannel"
         ],
         "Resource":"arn:aws:iotanalytics:::exampleChannel"
      },
      {
         "Sid":"ManageChannels",
         "Effect":"Allow",
         "Action":[
            "iotanalytics:CreateChannel",
            "iotanalytics:DeleteChannel",
            "iotanalytics:DescribeChannel",
            "iotanalytics:ListChannels",
            "iotanalytics:UpdateChannel"
         ],
         "Resource":"arn:aws:iotanalytics:::exampleChannel/*"
      }
   ]
}
```

Visualización AWS IoT Analytics de los canales en función de las etiquetas

Puedes usar las condiciones de tu política basada en la identidad para controlar el acceso a AWS IoT Analytics los recursos en función de las etiquetas. En este ejemplo, se muestra cómo crear una política que permita visualizar una `channel`. Sin embargo, los permisos solo se conceden si el `Owner` de la etiqueta del `channel` tiene el valor del nombre de usuario de ese usuario. Esta política también proporciona los permisos necesarios para llevar a cabo esta acción en la consola.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ListChannelsInConsole",
            "Effect": "Allow",
            "Action": "iotanalytics:ListChannels",
            "Resource": "*"
        },
        {
            "Sid": "ViewChannelsIfOwner",
            "Effect": "Allow",
            "Action": "iotanalytics:ListChannels",
            "Resource": "arn:aws:iotanalytics:*:*:channel/*",
            "Condition": {
                "StringEquals": {"iotanalytics:ResourceTag/Owner": "${aws:username}"}
```

```
            }
        }
    ]
}
```

Puede adjuntar esta política a los usuarios de su cuenta. Si un usuario llamado `richard-roe` intenta ver una AWS IoT Analytics `channel`, `channel` debe estar etiquetada. `0wner=richard-roe or owner=richard-roe` De lo contrario, se le deniega el acceso. La clave de la etiqueta de condición `0wner` coincide con los nombres de las claves de condición `0wner` y `owner` porque no distinguen entre mayúsculas y minúsculas. Para obtener más información, consulte Elementos de la política de JSON de IAM: Condición en la Guía del usuario de IAM.

Solución de problemas AWS IoT Analytics de identidad y acceso

Utilice la siguiente información como ayuda para diagnosticar y solucionar problemas comunes que puedan surgir al trabajar con ellos AWS IoT Analytics.

Temas

- No estoy autorizado a realizar ninguna acción en AWS IoT Analytics
- No tengo autorización para realizar iam:PassRole
- Quiero permitir que personas ajenas a mí accedan Cuenta de AWS a mis AWS IoT Analytics recursos

No estoy autorizado a realizar ninguna acción en AWS IoT Analytics

Si AWS Management Console le indica que no está autorizado a realizar una acción, debe ponerse en contacto con su administrador para obtener ayuda. Su administrador es la persona que le facilitó su nombre de usuario y contraseña.

El siguiente ejemplo de error se produce cuando el usuario de `mateojackson` intenta utilizar la consola para ver detalles sobre un `channel`, pero no tiene permisos de `iotanalytics:ListChannels`.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 iotanalytics:``ListChannels`` on resource: ``my-example-channel``
```

En este caso, Mateo pide a su administrador que actualice sus políticas de forma que pueda obtener acceso al recurso `my-example-channel` mediante la acción `iotanalytics:ListChannel`.

No tengo autorización para realizar **`iam:PassRole`**

Si recibe un error que indica que no tiene autorización para realizar la acción `iam:PassRole`, las políticas deben actualizarse a fin de permitirle pasar un rol a AWS IoT Analytics.

Algunas Servicios de AWS permiten transferir una función existente a ese servicio en lugar de crear una nueva función de servicio o una función vinculada a un 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 AWS IoT Analytics. 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. AWS El administrador es la persona que le proporcionó las credenciales de inicio de sesión.

Quiero permitir que personas ajenas a mí accedan Cuenta de AWS a mis AWS IoT Analytics recursos

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 saber si AWS IoT Analytics es compatible con estas funciones, consulta Cómo AWS IoT Analytics funciona con IAM.
- Para obtener información sobre cómo proporcionar acceso a los recursos de su Cuentas de AWS propiedad, consulte Proporcionar acceso a un usuario de IAM en otro Cuenta de AWS de su propiedad en la Guía del usuario de IAM.

- Para obtener información sobre cómo proporcionar acceso a tus recursos a terceros Cuentas de AWS, consulta Cómo proporcionar acceso a recursos que Cuentas de AWS 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.

Registro y monitoreo en AWS IoT Analytics

AWS proporciona herramientas que puede utilizar para monitorear AWS IoT Analytics. Puede configurar algunas de estas herramientas para que realicen la monitorización por usted. Algunas de las herramientas requieren intervención manual. Le recomendamos que automatice las tareas de monitorización en la medida de lo posible.

Herramientas de monitoreo automatizadas

Puede utilizar las siguientes herramientas de monitorización automatizado para vigilar AWS IoT e informar cuando haya algún problema:

- Registros de Amazon CloudWatch: monitoree, almacene y tenga acceso a los archivos de registro de AWS CloudTrail u otras fuentes. Para más información, consulte ¿Qué es monitoreo de archivos de registro de AWS CloudTrail en la Guía del usuario de Amazon CloudWatch.
- Monitoreo de registros de AWS CloudTrail: comparta archivos de registro entre cuentas, monitoree los archivos de registro de CloudTrail en tiempo real enviándolos a CloudWatch Logs, escriba aplicaciones de procesamiento de registros en Java y compruebe que los archivos de registro no hayan cambiado después de que CloudTrail los entregara. Para obtener más información, consulte Uso de archivos de registro de CloudTrail en la Guía del usuario de AWS CloudTrail.

Herramientas de monitoreo manuales

Otra parte importante del monitoreo de AWS IoT implica el monitoreo manual de los elementos que no cubren las alarmas de CloudWatch. Los paneles de la consola de servicios AWS IoT, CloudWatch

y otros servicios de AWS proporcionan una visión general del estado de su entorno de AWS. Le recomendamos que también compruebe los archivos de registro en AWS IoT Analytics.

- La consola de AWS IoT Analytics muestra:
  - Canales
  - Canalizaciones
  - Almacenes de datos
  - Conjuntos de datos
  - Cuadernos
  - Configuración
  - Aprendizaje
- La página principal de CloudWatch muestra:
  - Alarmas y estado actual
  - Gráficos de alarmas y recursos
  - Estado de los servicios

  Además, puede utilizar CloudWatch para hacer lo siguiente:
  - Crear paneles personalizados para monitorear los servicios que le interesan.
  - Realizar un gráfico con los datos de las métricas para resolver problemas y descubrir tendencias
  - Buscar y examinar todas sus métricas de recursos de AWS.
  - Crear y editar las alarmas de notificación de problemas

Monitorización con Registros de Amazon CloudWatch

AWS IoT Analytics admite el registro con Amazon CloudWatch. Puede habilitar y configurar el registro de Amazon CloudWatch para AWS IoT Analytics mediante la operación de `PutLoggingOptions` de la API. En esta sección se explica cómo se puede utilizar `PutLoggingOptions` con AWS Identity and Access Management (IAM) para configurar y habilitar el registro de Amazon CloudWatch para AWS IoT Analytics.

Para obtener más información acerca de los Registros de CloudWatch, consulte la Guía del usuario de los Registros de Amazon CloudWatch. Para obtener más información sobre IAM de AWS, consulte la Guía del usuario de AWS Identity and Access Management.

> **Note**
>
> Antes de habilitar el registro de AWS IoT Analytics, asegúrese de comprender bien los permisos de acceso a CloudWatch Logs. Los usuarios con acceso a CloudWatch Logs podrán consultar la información de depuración. Para obtener más información, consulte Autenticación y control de acceso de Registros de Amazon CloudWatch.

Creación de un rol de IAM para habilitar registros

Para crear un rol de IAM que permita la habilitación de registros en Amazon CloudWatch

1. Utilice la consola de IAM de AWS o el siguiente comando de la CLI de IAM de AWS, CreateRole, para crear un nuevo rol de IAM con una política de relación de confianza (política de confianza). La política de confianza otorga permiso a una entidad, como Amazon CloudWatch, para asumir la función.

   ```
   aws iam create-role --role-name exampleRoleName --assume-role-policy-document
         exampleTrustPolicy.json
   ```

   El archivo `exampleTrustPolicy.json` contiene el siguiente contenido.

   > **Note**
   >
   > Este ejemplo incluye una clave de contexto de condición global para protegerse contra el problema de seguridad de suplente confuso. Sustituya *123456789012* con el ID de su cuenta de AWS y *región de AWS* con la región de AWS de sus recursos de AWS. Para obtener más información, consulte the section called “Prevención del suplente confuso entre servicios”.

   ```
   {
      "Version": "2012-10-17",
      "Statement": [
         {
            "Effect": "Allow",
            "Principal": {
               "Service": "iotanalytics.amazonaws.com"
   ```

```
        },
        "Action": "sts:AssumeRole",
        "Condition": {
          "StringEquals": {
            "aws:SourceAccount": "123456789012"
          },
          "ArnLike": {
            "aws:SourceArn": "arn:aws:iotanalytics:aws-region:123456789012:*"
          }
        }

    }
  ]
}
```

Utilizará el ARN de este rol más adelante cuando llame al comando `PutLoggingOptions` de AWS IoT Analytics.

2. Utilice PutRolePolicy de IAM de AWS para adjuntar una política de permisos (una `role policy`) al rol que creó en el paso 1.

   ```
   aws iam put-role-policy --role-name exampleRoleName --policy-name
   examplePolicyName --policy-document exampleRolePolicy.json
   ```

   El archivo exampleRolePolicy.json tiene el siguiente contenido.

   ```
   {
     "Version": "2012-10-17",
     "Statement": [
     {
     "Effect": "Allow",
     "Action": [
      "logs:CreateLogGroup",
      "logs:CreateLogStream"
     ],
     "Resource": [
      "arn:aws:logs:*:*:*"
     ]
     }
     ]
   }
   ```

3. Para conceder a AWS IoT Analytics permiso para colocar eventos de registro en Amazon CloudWatch, utilice el comando PutreSourcePolicy de Amazon CloudWatch.

> **Note**
>
> Para evitar el problema de seguridad del suplente confuso, le recomendamos que especifique `aws:SourceArn` en su política de recursos. Esto restringe el acceso y admite solo las solicitudes que provienen de una cuenta específica. Para obtener más información sobre el problema del suplente confuso, consulte the section called "Prevención del suplente confuso entre servicios".

```
aws logs put-resource-policy --policy-in-json
exampleResourcePolicy.json
```

El archivo `exampleResourcePolicy.json` contiene la siguiente política de recursos.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "iotanalytics.amazonaws.com"
            },
            "Action": "logs:PutLogEvents",
            "Resource": "*",
            "Condition":{
                "ArnLike":{
                    "aws:SourceArn":"arn:aws:iotanalytics:us-east-1:123456789012:*/
*"
                },
                "StringEquals":{
                    "aws:SourceAccount":"123456789012"
                }
            }
    ]
}
```

Configurar y habilitar el registro

Utilice el comando `PutLoggingOptions` para configurar y habilitar el registro de Amazon CloudWatch para AWS IoT Analytics. El valor `roleArn` del campo `loggingOptions` deber ser el ARN del rol creado en la sección anterior. También puede utilizar el comando `DecribeLoggingOptions` para comprobar la configuración de las opciones de registro.

PutLoggingOptions

Establece o actualiza las opciones de registro de AWS IoT Analytics. Si actualiza el valor de cualquier campo `loggingOptions`, los cambios tardarán hasta un minuto en surtir efecto. Por otra parte, si cambia la política asociada al rol que ha especificado en el campo `roleArn` (p. ej., para corregir una política que no es válida), el cambio puede tardar hasta cinco minutos en surtir efecto. Para obtener más información, consulte `PutLoggingOptions`.

DescribeLoggingOptions

Recupera la configuración actual de las opciones de registro de AWS IoT Analytics. Para obtener más información, consultar `DescribeLoggingOptions`

Espacio de nombres, métricas y dimensiones

AWS IoT Analytics coloca las siguientes métricas en el repositorio de Amazon CloudWatch:

Espacio de nombres
AWS/IoTAnalytics

Métrica	Descripción
ActionExecution	El número de acciones ejecutadas.
ActionExecutionThrottled	El número de acciones que se han limitado.
ActivityExecutionError	El número de errores generados mientras se ejecuta la actividad de canalización.
IncomingMessages	El número de mensajes que entran en el canal.

Métrica	Descripción
PipelineConcurrentExecutionCount	El número de actividades de canalización que se han ejecutado simultáneamente.

Dimensión	Descripción
ActionType	El tipo de acción que se está monitorizando.
ChannelName	El nombre del canal que se está monitorizando.
DatasetName	El nombre del conjunto de datos que se está monitorizando.
DatastoreName	El nombre del almacén de datos que se está monitorizando.
PipelineActivityName	El nombre de la actividad de canalización que se está monitorizando.
PipelineActivityType	El tipo de la actividad de canalización que se está monitorizando.
PipelineName	El nombre de la canalización que se está monitorizando.

Supervisión con Eventos de Amazon CloudWatch

AWS IoT Analytics publica automáticamente un evento en Eventos de Amazon CloudWatch cuando se produce un error de tiempo de ejecución durante una actividad de AWS Lambda. Este evento contiene un mensaje de error detallado y las claves de los objetos de Amazon Simple Storage Service (Amazon S3) que almacenan mensajes de canal sin procesar. Puede utilizar las claves de Amazon S3 para volver a procesar los mensajes del canal sin procesar. Para obtener más información, consulte Reprocesamiento de los mensajes de canal, la API StartPipelineReprocessing en la Referencia de la API de AWS IoT Analytics y Qué son los Eventos de Amazon CloudWatch en la Guía del usuario de Eventos de Amazon CloudWatch.

También puede configurar destinos que permitan a Eventos de Amazon CloudWatch enviar notificaciones o realizar otras acciones. Por ejemplo, puede enviar la notificación a una cola de Amazon Simple Queue Service (Amazon SQS) y, a continuación, invocar la API `StartReprocessingMessage` para procesar los mensajes de canal guardados en los objetos de Amazon S3. Eventos de Amazon CloudWatch admite muchos tipos de destinos, como los siguientes:

- Amazon Kinesis Streams
- Funciones de AWS Lambda
- Temas de Amazon Simple Notification Service (Amazon SNS)
- Colas de Amazon Simple Queue Service (Amazon SQS)

Para ver una lista de los objetivos admitidos, consulte Destinos de Amazon EventBridge en la Guía del usuario de Amazon EventBridge.

Sus recursos de Eventos de CloudWatch y los destinos asociados deben estar en la región de AWS en la que creó los recursos de AWS IoT Analytics. Para obtener más información, consulte Puntos de enlace y cuotas en la Referencia general de AWS.

La notificación que se envía a Eventos de Amazon CloudWatch sobre errores de tiempo de ejecución en la actividad AWS Lambda utiliza el siguiente formato.

```
{
    "version": "version-id",
    "id": "event-id",
    "detail-type": "IoT Analytics Pipeline Failure Notification",
    "source": "aws.iotanalytics",
    "account": "aws-account",
    "time": "timestamp",
    "region": "aws-region",
    "resources": [
        "pipeline-arn"
    ],
    "detail": {
        "event-detail-version": "1.0",
        "pipeline-name": "pipeline-name",
        "error-code": "LAMBDA_FAILURE",
        "message": "error-message",
        "channel-messages": {
            "s3paths": [
                "s3-keys"
```

```
            ]
        },
        "activity-name": "lambda-activity-name",
        "lambda-function-arn": "lambda-function-arn"
    }
}
```

Notificaciones de ejemplo:

```
{
    "version": "0",
    "id": "204e672e-ef12-09af-4cfd-de3b53673ec6",
    "detail-type": "IoT Analytics Pipeline Failure Notification",
    "source": "aws.iotanalytics",
    "account": "123456789012",
    "time": "2020-10-15T23:47:02Z",
    "region": "ap-southeast-2",
    "resources": [
        "arn:aws:iotanalytics:ap-southeast-2:123456789012:pipeline/
test_pipeline_failure"
    ],
    "detail": {
        "event-detail-version": "1.0",
        "pipeline-name": "test_pipeline_failure",
        "error-code": "LAMBDA_FAILURE",
        "message": "Temp unavaliable",
        "channel-messages": {
        "s3paths": [
            "test_pipeline_failure/channel/cmr_channel/__dt=2020-10-15
 00:00:00/1602805530000_1602805560000_123456789012_cmr_channel_0_257.0.json.gz"
        ]
    },
    "activity-name": "LambdaActivity_33",
    "lambda-function-arn": "arn:aws:lambda:ap-
southeast-2:123456789012:function:lambda_activity"
    }
}
```

Obtención de notificaciones de datos atrasados a través de Eventos de Amazon CloudWatch

Al crear contenidos de conjuntos de datos con datos de mensajes de un periodo de tiempo determinado, es posible que algunos de estos datos no lleguen a tiempo para ser procesados. Para permitir un retraso, puede especificar un desplazamiento de `deltaTime` para `QueryFilter` cuando cree un conjunto de datos mediante la aplicación de una `queryAction` (una consulta SQL). AWS IoT Analytics sigue procesando los datos que llegan dentro del tiempo delta y el contenido del conjunto de datos tiene un desfase temporal. La característica de notificación de datos atrasados permite a AWS IoT Analytics enviar notificaciones a través de Eventos de Amazon CloudWatch cuando los datos llegan después del tiempo delta.

Puede usar la consola de AWS IoT Analytics, la API, AWS Command Line Interface (AWS CLI) o el SDK de AWS para especificar las reglas de datos atrasados para un conjunto de datos.

En la API de AWS IoT Analytics, el objeto `LateDataRuleConfiguration` representa la configuración de las reglas de datos atrasados de un conjunto de datos. Este objeto forma parte del objeto `Dataset` asociado a las operaciones `CreateDataset` y `UpdateDataset` de la API.

Parámetros

Cuando cree una regla de datos tardíos para un conjunto de datos con AWS IoT Analytics, debe especificar la siguiente información:

`ruleConfiguration` (**`LateDataRuleConfiguration`**)

Una estructura que contiene la información de configuración de una regla de datos tardíos.

`deltaTimeSessionWindowConfiguration`

Una estructura que contiene la información de configuración de una ventana de sesión de tiempo delta.

DeltaTime (Tiempo delta) especifica un intervalo de tiempo. Puede utilizar `DeltaTime` para crear contenido de un conjunto de datos con los datos que hayan llegado al almacén de datos desde la última ejecución. Para ver un ejemplo de `DeltaTime`, consulte Creación de un conjunto de datos SQL con una ventana diferencial (CLI).

`timeoutInMinutes`

Un intervalo de tiempo. Puede utilizar `timeoutInMinutes` para que AWS IoT Analytics envíe por lotes las notificaciones de datos tardíos que se hayan generado desde la última

ejecución. AWS IoT Analytics envía un lote de notificaciones a Eventos de CloudWatch al mismo tiempo.

Tipo: entero

Rango válido: 1-60

`ruleName`

El nombre de la regla de datos tardíos.

Tipo: String

> **Important**
>
> Para especificar `lateDataRules`, el conjunto de datos debe utilizar un filtro `DeltaTime`.

Configuración de las reglas de datos atrasados (consola)

En el siguiente procedimiento se explica cómo configurar la regla de datos atrasados de un conjunto de datos en la consola de AWS IoT Analytics.

Para configurar las reglas de datos atrasados

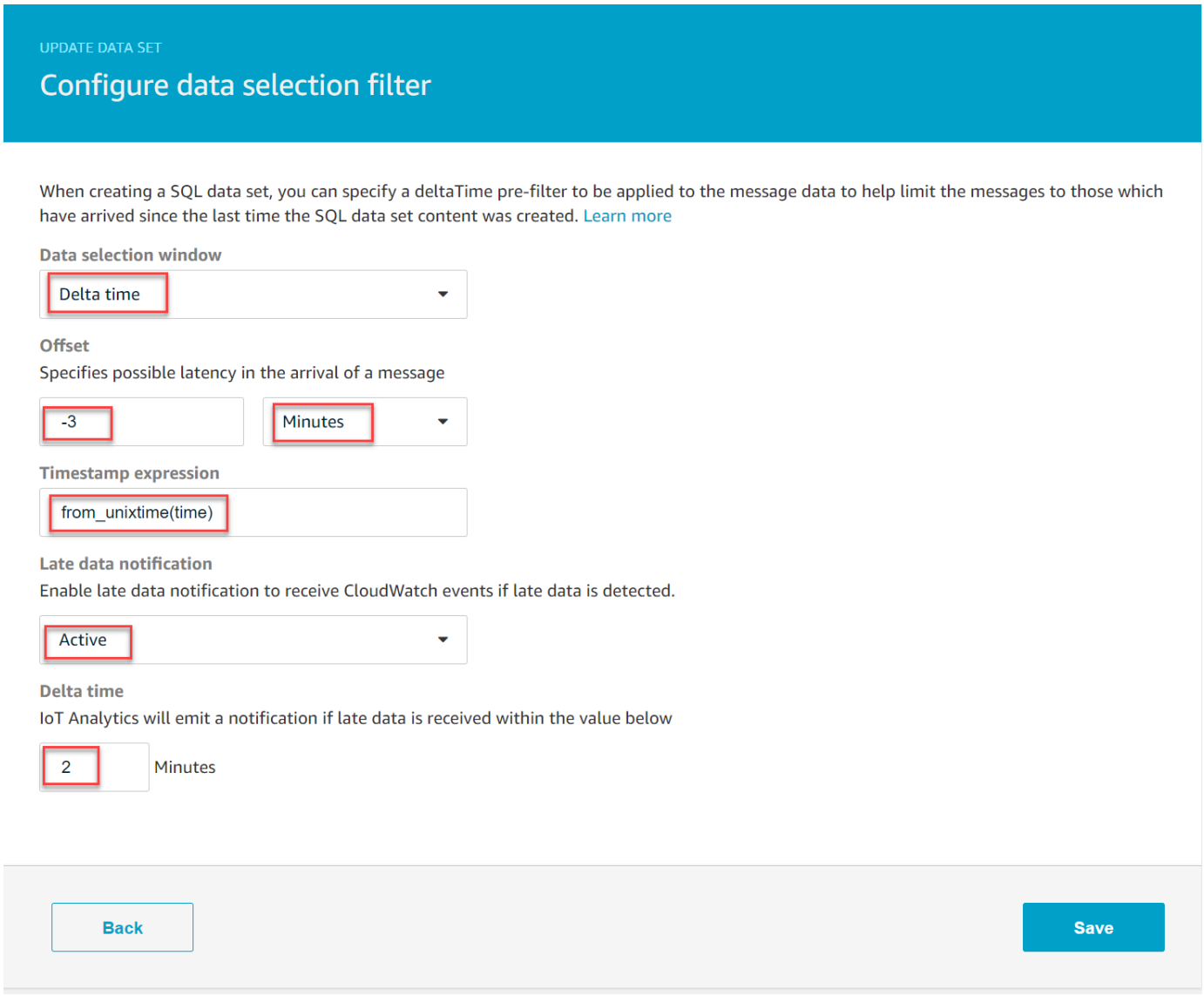
1. Inicie sesión en la consola de AWS IoT Analytics.
2. En el panel de navegación, seleccione Conjuntos de datos.
3. En Conjuntos de datos, seleccione el conjunto de datos de destino.
4. En el panel de navegación, seleccione Detalles.
5. En la sección Ventana diferencial, seleccione Editar.
6. En Configurar el filtro de selección de datos, haga lo siguiente:
   a. En la ventana de Selección de datos, seleccione Tiempo delta.
   b. En Desplazamiento, introduzca un período de tiempo y, a continuación, seleccione una unidad.
   c. En Expresión de marca de tiempo, introduzca una expresión. Puede ser el nombre de un campo de marca de tiempo o una expresión SQL que puede obtener la hora, como por ejemplo, *`from_unixtime(time)`*.

Para obtener más información sobre cómo escribir una expresión de marca de tiempo, consulte Funciones y operadores de fecha y hora, en la documentación de Presto 0.172.

d. En Notificación de datos atrasados, seleccione Activo.

e. En Tiempo diferencia, introduzca un número entero. El rango válido es de 1 a 60.

f. Seleccione Save (Guardar).



Configuración de reglas de datos atrasados (CLI)

En la API de AWS IoT Analytics, el objeto `LateDataRuleConfiguration` representa la configuración de las reglas de datos atrasados de un conjunto de datos. Este objeto forma parte del

objeto `Dataset` asociado a `CreateDataset` y `UpdateDataset`. Puede usar la API, AWS CLI, o el SDK de AWS para especificar reglas de datos atrasados para un conjunto de datos. El siguiente ejemplo utiliza la AWS CLI.

Use el siguiente comando para crear un conjunto de datos atrasados específicos. El comando da por hecho que el archivo `dataset.json` está en el directorio actual.

> **Note**
>
> Puede usar la API UpdateDataset para actualizar un conjunto de datos existente.

```
aws iotanalytics create-dataset --cli-input-json file://dataset.json
```

El archivo `dataset.json` debe contener lo siguiente:

- Sustituya *`demo_dataset`* por el nombre del conjunto de datos de destino.
- Sustituya *`demo_datastore`* por el nombre del almacén de datos de destino.
- Sustituya *`from_unixtime(time)`* por el nombre de un campo de marca de tiempo o una expresión SQL que pueda obtener la hora.

  Para obtener más información sobre cómo escribir una expresión de marca de tiempo, consulte Funciones y operadores de fecha y hora, en la documentación de Presto 0.172.
- Sustituya *`tiemeout`* por un número entero comprendido entre 1 y 60.
- Sustituya *`demo_rule`* por cualquier nombre.

```
{
    "datasetName": "demo_dataset",
    "actions": [
        {
            "actionName": "myDatasetAction",
            "queryAction": {
                "filters": [
                    {
                        "deltaTime": {
                            "offsetSeconds": -180,
                            "timeExpression": "from_unixtime(time)"
```

```
                        }
                    }
                ],
                "sqlQuery": "SELECT * FROM demo_datastore"
            }
        }
    ],
    "retentionPeriod": {
        "unlimited": false,
        "numberOfDays": 90
    },
    "lateDataRules": [
        {
            "ruleConfiguration": {
                "deltaTimeSessionWindowConfiguration": {
                    "timeoutInMinutes": timeout
                }
            },
            "ruleName": "demo_rule"
        }
    ]
}
```

Suscripción para recibir notificaciones de datos atrasados

Puede crear reglas en Eventos de CloudWatchque que definan cómo procesar las notificaciones de datos atrasados enviadas desde AWS IoT Analytics. Cuando Eventos de CloudWatch recibe las notificaciones, invoca las acciones de destino especificadas y definidas en sus reglas.

Requisitos previos para crear las reglas de Eventos de CloudWatch

Antes de crear una regla de Eventos de CloudWatch para AWS IoT Analytics, debe hacer lo siguiente:

- Familiarizarse con los eventos, las reglas y los destinos de Eventos de CloudWatch.
- Cree y configure los destinos que las reglas de Eventos de CloudWatch han invocado. Las reglas pueden invocar muchos tipos de destinos, como los siguientes:
  - Amazon Kinesis Streams
  - Funciones de AWS Lambda
  - Temas de Amazon Simple Notification Service (Amazon SNS)
  - Colas de Amazon Simple Queue Service (Amazon SQS)

La regla de Eventos de CloudWatch y los destinos asociados deben estar en la región de AWS en la que creó sus recursos de AWS IoT Analytics. Para obtener más información, consulte Puntos de enlace y cuotas en la Referencia general de AWS.

Para obtener más información, consulte ¿Qué es Eventos de Amazon CloudWatch? e Introducción a Eventos de Amazon CloudWatch en la Guía del usuario de Eventos de Amazon CloudWatch.

Evento de notificaciones de datos atrasados

El evento de notificaciones de datos atrasados utiliza el siguiente formato.

```
{
 "version": "0",
 "id": "7f51dfa7-ffef-97a5-c625-abddbac5eadd",
 "detail-type": "IoT Analytics Dataset Lifecycle Notification",
 "source": "aws.iotanalytics",
 "account": "123456789012",
 "time": "2020-05-14T02:38:46Z",
 "region": "us-east-2",
 "resources": ["arn:aws:iotanalytics:us-east-2:123456789012:dataset/demo_dataset"],
 "detail": {
  "event-detail-version": "1.0",
  "dataset-name": "demo_dataset",
  "late-data-rule-name": "demo_rule",
  "version-ids": ["78244852-8737-4650-aa4d-3071a01338fa"],
  "message": null
 }
}
```

Cree una regla de Eventos de CloudWatch para recibir notificaciones de datos atrasados.

En el siguiente procedimiento se explica cómo crear una regla que envíe notificaciones de datos atrasados de AWS IoT Analytics a una cola de Amazon SQS.

Para crear manualmente una regla de Eventos de CloudWatch

1. Inicie sesión en la consola de Amazon CloudWatch.
2. En el panel de navegación, en Events (Eventos), seleccione Rules (Reglas).
3. En la página Reglas, seleccione Crear una regla.
4. En Origen de eventos, seleccione Patrón de eventos.

5. En la sección Crear patrón de eventos para buscar eventos coincidentes por servicio, haga lo siguiente:

   a. En Nombre del servicio, seleccione IoT Analytics

   b. En Tipo de evento, seleccione Notificación del ciclo de vida del conjunto de datos de IoT Analytics.

   c. Seleccione Nombre o nombres de conjuntos de datos específicos y, a continuación, introduzca el nombre del conjunto de datos de destino.

6. En Destinos, seleccione Añadir objetivo*.
7. Seleccione Cola SQS y, a continuación, haga lo siguiente:

   • En Cola*, seleccione la cola de destino.

8. Seleccione Configure details (Configurar detalles).
9. En la página Paso 2: Configurar detalles de la regla, escriba el nombre y una descripción.
10. Seleccione Create rule (Crear regla).

Registrar llamadas a la API de AWS IoT Analytics con AWS CloudTrail

AWS IoT Analytics se integra con AWS CloudTrail, un servicio que proporciona un registro de las acciones hechas por un usuario, un rol o un servicio de AWS en AWS IoT Analytics. CloudTrail captura un subconjunto de llamadas a la API de AWS IoT Analytics como eventos, incluidas las llamadas procedentes de la consola de AWS IoT Analytics y las llamadas de código a las API de AWS IoT Analytics. Si crea un registro de seguimiento, puede habilitar la entrega continua de eventos de CloudTrail a un bucket de Amazon S3, incluidos los eventos para AWS IoT Analytics. Si no configura un registro de seguimiento, puede ver los eventos más recientes de la consola de CloudTrail en el Historial de eventos. Mediante la información recopilada por CloudTrail, puede determinar la solicitud que se realizó a AWS IoT Analytics, la dirección IP desde la que se realizó, quién la realizó y cuándo, etc.

Para obtener más información acerca de CloudTrail, consulte la Guía del usuario de AWS CloudTrail.

Información de AWS IoT Analytics en AWS CloudTrail

CloudTrail se habilita en su cuenta de AWS cuando la crea. Cuando se produce una actividad en AWS IoT Analytics, esa actividad se registra en un evento de CloudTrail junto con otros eventos de servicio de AWS en Event history (Historial de eventos). Puede ver, buscar y descargar los últimos

eventos de la cuenta de AWS. Para obtener más información, consulte Visualización de eventos con el historial de eventos de CloudTrail.

Para mantener un registro continuo de eventos en la cuenta de AWS, incluidos los eventos de AWS IoT Analytics, cree un registro de seguimiento. Un registro de seguimiento permite a CloudTrail enviar archivos de registro a un bucket de Amazon S3. De manera predeterminada, cuando crea un registro de seguimiento en la consola, el registro de seguimiento se aplica a todas las regiones. 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 Amazon S3 especificado. También es posible configurar otros servicios de AWS para analizar en profundidad y actuar en función de los datos de eventos recopilados en los registros de CloudTrail. Para obtener más información, consulte:

- Introducción a la creación de registros de seguimiento
- Servicios e integraciones compatibles con CloudTrail
- Configuración de notificaciones de Amazon SNS para CloudTrail
- Recibir archivos de registros de CloudTrail de varias regiones y Recepción de archivos de registros de CloudTrail de varias cuentas

AWS IoT Analytics admite el registro de las siguientes acciones como eventos en archivos de registros de CloudTrail:

- CancelPipelineReprocessing
- CreateChannel
- CreateDataset
- CreateDatasetContent
- CreateDatastore
- CreatePipeline
- DeleteChannel
- DeleteDataset
- DeleteDatasetContent
- DeleteDatastore
- DeletePipeline
- DescribeChannel
- DescribeDataset

- DescribeDatastore
- DescribeLoggingOptions
- DescribePipeline
- GetDatasetContent
- ListChannels
- ListDatasets
- ListDatastores
- ListPipelines
- PutLoggingOptions
- RunPipelineActivity
- SampleChannelData
- StartPipelineReprocessing
- UpdateChannel
- UpdateDataset
- UpdateDatastore
- UpdatePipeline

Cada entrada de registro o evento contiene información sobre quién generó la solicitud. La información de identidad del usuario le ayuda a determinar lo siguiente:

- Si la solicitud se realizó con las credenciales raíz o del usuario de AWS Identity and Access Management.
- 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.

Descripción de las entradas de los archivos de registro de AWS IoT Analytics

Un registro de seguimiento es una configuración que permite la entrega de eventos como archivos de log al bucket de S3 que se especifique. Los archivos de registro de CloudTrail contienen una o

más entradas de registro. Un evento representa la solicitud específica realizada desde un origen y contiene información sobre la acción solicitada, la fecha y la hora de la acción, los parámetros de la solicitud, etc. Los archivos de registro de CloudTrail no son un rastro de pila ordenado de las llamadas a las API públicas, por lo que no aparecen en ningún orden específico.

En el ejemplo siguiente, se muestra una entrada de registro de CloudTrail que ilustra la acción `CreateChannel`.


```
{
"eventVersion": "1.05",
"userIdentity": {
"type": "AssumedRole",
"principalId": "ABCDE12345FGHIJ67890B:AnalyticsChannelTestFunction",
"arn": "arn:aws:sts::123456789012:assumed-role/AnalyticsRole/
AnalyticsChannelTestFunction",
"accountId": "123456789012",
"accessKeyId": "ABCDE12345FGHIJ67890B",
"sessionContext": {
"attributes": {
 "mfaAuthenticated": "false",
 "creationDate": "2018-02-14T23:43:12Z"
},
"sessionIssuer": {
 "type": "Role",
 "principalId": "ABCDE12345FGHIJ67890B",
 "arn": "arn:aws:iam::123456789012:role/AnalyticsRole",
 "accountId": "123456789012",
 "userName": "AnalyticsRole"
}
}
},
"eventTime": "2018-02-14T23:55:14Z",
"eventSource": "iotanalytics.amazonaws.com",
"eventName": "CreateChannel",
"awsRegion": "us-east-1",
"sourceIPAddress": "198.162.1.0",
"userAgent": "aws-internal/3 exec-env/AWS_Lambda_java8",
"requestParameters": {
"channelName": "channel_channeltest"
},
"responseElements": {
"retentionPeriod": {
"unlimited": true
```

```
},
"channelName": "channel_channeltest",
"channelArn": "arn:aws:iotanalytics:us-east-1:123456789012:channel/channel_channeltest"
},
"requestID": "7f871429-11e2-11e8-9eee-0781b5c0ac59",
"eventID": "17885899-6977-41be-a6a0-74bb95a78294",
"eventType": "AwsApiCall",
"recipientAccountId": "123456789012"
}
```

En el ejemplo siguiente, se muestra una entrada de registro de CloudTrail que ilustra la acción `CreateDataset`.

```
{
"eventVersion": "1.05",
"userIdentity": {
"type": "AssumedRole",
"principalId": "ABCDE12345FGHIJ67890B:AnalyticsDatasetTestFunction",
"arn": "arn:aws:sts::123456789012:assumed-role/AnalyticsRole/
AnalyticsDatasetTestFunction",
"accountId": "123456789012",
"accessKeyId": "ABCDE12345FGHIJ67890B",
"sessionContext": {
"attributes": {
 "mfaAuthenticated": "false",
 "creationDate": "2018-02-14T23:41:36Z"
},
"sessionIssuer": {
 "type": "Role",
 "principalId": "ABCDE12345FGHIJ67890B",
 "arn": "arn:aws:iam::123456789012:role/AnalyticsRole",
 "accountId": "123456789012",
 "userName": "AnalyticsRole"
}
}
},
"eventTime": "2018-02-14T23:53:39Z",
"eventSource": "iotanalytics.amazonaws.com",
"eventName": "CreateDataset",
"awsRegion": "us-east-1",
"sourceIPAddress": "198.162.1.0",
"userAgent": "aws-internal/3 exec-env/AWS_Lambda_java8",
"requestParameters": {
```

```
"datasetName": "dataset_datasettest"
},
"responseElements": {
"datasetArn": "arn:aws:iotanalytics:us-east-1:123456789012:dataset/
dataset_datasettest",
"datasetName": "dataset_datasettest"
},
"requestID": "46ee8dd9-11e2-11e8-979a-6198b668c3f0",
"eventID": "5abe21f6-ee1a-48ef-afc5-c77211235303",
"eventType": "AwsApiCall",
"recipientAccountId": "123456789012"
}
```


Validación de conformidad para AWS IoT Analytics

Para saber si uno Servicio de AWS está dentro del ámbito de aplicación de programas de cumplimiento específicos, consulte Servicios de AWS Alcance por programa de cumplimiento Servicios de AWS de cumplimiento y elija el programa de cumplimiento que le interese. Para obtener información general, consulte Programas de AWS cumplimiento > Programas AWS .

Puede descargar informes de auditoría de terceros utilizando AWS Artifact. Para obtener más información, consulte Descarga de informes en AWS Artifact .

Su responsabilidad de cumplimiento al Servicios de AWS utilizarlos viene determinada por la confidencialidad de sus datos, los objetivos de cumplimiento de su empresa y las leyes y reglamentos aplicables. AWS proporciona los siguientes recursos para ayudar con el cumplimiento:

- Guías de inicio rápido sobre seguridad y cumplimiento: estas guías de implementación analizan las consideraciones arquitectónicas y proporcionan los pasos para implementar entornos básicos centrados en AWS la seguridad y el cumplimiento.
- Diseño de arquitectura para garantizar la seguridad y el cumplimiento de la HIPAA en Amazon Web Services: en este documento técnico se describe cómo pueden utilizar AWS las empresas para crear aplicaciones aptas para la HIPAA.

> **Note**
>
> No Servicios de AWS todas cumplen los requisitos de la HIPAA. Para más información, consulte la Referencia de servicios compatibles con HIPAA.

- AWS Recursos de de cumplimiento: esta colección de libros de trabajo y guías puede aplicarse a su industria y ubicación.
- AWS Guías de cumplimiento para clientes: comprenda el modelo de responsabilidad compartida desde la perspectiva del cumplimiento. Las guías resumen las mejores prácticas para garantizar la seguridad Servicios de AWS y orientan los controles de seguridad en varios marcos (incluidos el Instituto Nacional de Estándares y Tecnología (NIST), el Consejo de Normas de Seguridad del Sector de Tarjetas de Pago (PCI) y la Organización Internacional de Normalización (ISO)).
- Evaluación de los recursos con reglas en la guía para AWS Config desarrolladores: el AWS Config servicio evalúa en qué medida las configuraciones de los recursos cumplen con las prácticas internas, las directrices del sector y las normas.
- AWS Security Hub— Esto Servicio de AWS proporciona una visión completa del estado de su seguridad interior AWS. Security Hub utiliza controles de seguridad para evaluar sus recursos de AWS y comprobar su cumplimiento con los estándares y las prácticas recomendadas del sector de la seguridad. Para obtener una lista de los servicios y controles compatibles, consulte la Referencia de controles de Security Hub.
- Amazon GuardDuty: Servicio de AWS detecta posibles amenazas para sus cargas de trabajo Cuentas de AWS, contenedores y datos mediante la supervisión de su entorno para detectar actividades sospechosas y maliciosas. GuardDuty puede ayudarlo a cumplir con varios requisitos de conformidad, como el PCI DSS, al cumplir con los requisitos de detección de intrusiones exigidos por ciertos marcos de cumplimiento.
- AWS Audit Manager— Esto le Servicio de AWS ayuda a auditar continuamente su AWS uso para simplificar la gestión del riesgo y el cumplimiento de las normativas y los estándares del sector.

Resiliencia en AWS IoT Analytics

La infraestructura AWS global se basa en AWS regiones y zonas de disponibilidad. 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 conmutan automáticamente entre zonas de disponibilidad sin interrupción. 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 AWS las regiones y las zonas de disponibilidad, consulte infraestructura AWS global.

Seguridad de la infraestructura en AWS IoT Analytics

Como servicio gestionado, AWS IoT Analytics está protegido por la seguridad de la red AWS global. Para obtener información sobre los servicios AWS de seguridad y cómo se AWS protege la infraestructura, consulte Seguridad AWS en la nube. Para diseñar su AWS entorno utilizando las mejores prácticas de seguridad de la infraestructura, consulte Protección de infraestructuras en un marco de buena AWS arquitectura basado en el pilar de la seguridad.

Utiliza las llamadas a la API AWS publicadas para acceder a través de la red. Los clientes deben admitir lo siguiente:

- Seguridad de la capa de transporte (TLS). Exigimos TLS 1.2 y recomendamos TLS 1.3.
- Conjuntos de cifrado con confidencialidad directa total (PFS) 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.

Cuotas de AWS IoT Analytics

La Guía de Referencia general de AWS proporciona las cuotas predeterminadas de AWS IoT Analytics para una cuenta de AWS. A menos que se especifique, cada cuota es por región de AWS. Para obtener más información, consulte AWS IoT AnalyticsPuntos de conexión y cuotas y Service Quotas de AWS en la Guía de Referencia general de AWS.

Para solicitar un aumento de Service Quotas, envíe un caso de soporte a la consola del Centro de soporte. Para obtener más información, consulte Solicitud de un aumento de cuota en la Guía del usuario de Service Quotas.

Comandos de la AWS IoT Analytics

Lea este tema para obtener información sobre las operaciones de la API para AWS IoT Analytics, incluidas solicitudes, respuestas y errores de muestra de los protocolos de servicios web compatibles.

Acciones de AWS IoT Analytics

Puede usar los comandos de la API de AWS IoT Analytics para recopilar, procesar, almacenar y analizar sus datos de IoT. Para obtener más información, consulta las acciones que se admiten en AWS IoT Analytics en la Referencia de la API de AWS IoT Analytics.

Las secciones de AWS IoT Analytics de la Referencia de comandos de AWS CLI incluyen los comandos de AWS CLI que puede utilizar para administrar y manipular AWS IoT Analytics.

Datos de AWS IoT Analytics

Puede usar los comandos de la API de datos de AWS IoT Analytics para realizar actividades avanzadas con `channel`, `pipeline`,`datastore`, y `dataset` de AWS IoT Analytics. Para obtener más información, consulte los tipos de datos compatibles con los datos de AWS IoT Analytics en la Referencia de la API de AWS IoT Analytics.

Solución de problemas de AWS IoT Analytics

Consulte la siguiente sección para solucionar errores y encontrar posibles soluciones para resolverlos con AWS IoT Analytics.

Temas

¿Cómo sé si mis mensajes están llegando a AWS IoT Analytics?

Compruebe que la regla está correctamente configurada para inyectar datos en el canal a través del motor de reglas.

```
aws iot get-topic-rule --rule-name your-rule-name
```

La respuesta debe ser similar a la siguiente.

```
{
```

```
    "ruleArn": "arn:aws:iot:us-west-2:your-account-id:rule/your-rule-name",
    "rule": {
        "awsIotSqlVersion": "2016-03-23",
        "sql": "SELECT * FROM 'iot/your-rule-name'",
        "ruleDisabled": false,
        "actions": [
            {
                "iotAnalytics": {
                    "channelArn":
 "arn:aws:iotanalytics:region:your_account_id:channel/your-channel-name"
                }
            }
        ],
        "ruleName": "your-rule-name"
    }
}
```

Asegúrese de que el nombre de la región y del canal utilizados en la regla sean correctos. Para asegurarse de que los datos llegan al motor de reglas y de que la regla se está ejecutando correctamente, conviene que añada un destino nuevo para almacenar los mensajes entrantes en el bucket de Amazon S3 temporalmente.

¿Por qué pierde mensajes mi canalización? ¿Cómo lo soluciono?

- Una actividad ha recibido una entrada JSON no válida:

  Todas las actividades, excepto las de Lambda, requieren específicamente una cadena JSON válida como entrada. Si el JSON recibido por una actividad no es válido, el mensaje se descarta y no llega al almacén de datos. Asegúrese de que se adquieren mensajes JSON válidos en el servicio. En caso de los datos binarios, asegúrese de que la primera actividad de la canalización es una actividad de Lambda que convierte los datos binarios en JSON válido antes de pasarlos a la siguiente actividad o de almacenarlos en el almacén de datos. Para obtener más información, consulte Ejemplo de funciones de Lambda 2.

- Una función de Lambda invocada por una actividad de Lambda no tiene permisos suficientes:

  Asegúrese de que cada función de Lambda de una actividad de Lambda tenga permisos para ser invocada desde el servicio de AWS IoT Analytics. Puede utilizar el siguiente comando de AWS CLI para conceder el permiso.

```
aws lambda add-permission --function-name <name> --region <region> --statement-id
 <id> --principal iotanalytics.amazonaws.com --action lambda:InvokeFunction
```

- Una actividad filter o removeAttribute no se ha definido correctamente:

  Asegúrese de que las definiciones si alguna de las actividades `filter` o `removeAttribute` son correctas. Si filtra un mensaje o elimina todos los atributos de un mensaje, dicho mensaje no se añade al almacén de datos.

¿Por qué no hay datos en mi almacén de datos?

- Existe un retraso entre la ingesta de datos y la disponibilidad de datos:

  Puede que pasen unos minutos desde que los datos se adquieren en un canal hasta que están disponibles en el almacén de datos. El tiempo depende del número de actividades de la canalización y de la definición de las actividades de Lambda personalizadas de la canalización.
- Los mensajes se excluyen de la canalización:

  Asegúrese de que no se están descartando mensajes de la canalización. (Consulte la pregunta anterior y su respuesta).
- La consulta del conjunto de datos es incorrecta:

  Asegúrese de que la consulta que genera el conjunto de datos desde el almacén de datos es correcta. Elimine los filtros innecesarios de la consulta para asegurarse de que los datos lleguen al almacén de datos.

¿Por qué mi conjunto de datos acaba de mostrar **__dt**?

- Esta columna la añade automáticamente el servicio y contiene el tiempo de adquisición aproximado de los datos. Puede que se utilice para optimizar las consultas. Si el conjunto de datos únicamente contiene esta columna, consulte la pregunta anterior y su respuesta.

¿Cómo puedo utilizar un evento controlado por la finalización de un conjunto de datos?

- Debe configurar un sondeo basado en el comando `describe-dataset` para comprobar si el estado del conjunto de datos con una marca de tiempo determinada es LOGRADO.

¿Cómo puedo configurar correctamente mi instancia del bloc de notas para utilizar el servicio AWS IoT Analytics?

Siga estos pasos para asegurarse de que el rol de IAM que está utilizando para crear la instancia del bloc de notas tiene los permisos necesarios:

1. Vaya a la consola de SageMaker y cree una instancia del bloc de notas.
2. Rellene los detalles y elija Create a new role (Crear un nuevo rol). Anote el ARN del rol.
3. Cree la instancia del bloc de notas. Esto también crea un rol que SageMaker puede utilizar.
4. Vaya a la consola de IAM y modifique el rol de Sagemaker recién creado. Al abrir dicho rol, este debe tener una política administrada.
5. Haga clic en Añadir política insertada, seleccione IoTAnalytics como servicio y, como permiso de lectura, seleccione GetDatasetContent.
6. Revise la política, asígnele un nombre y elija Create (Crear) para crearla. Esto otorga al rol recién creado un permiso de política para leer un conjunto de datos de AWS IoT Analytics.
7. Vaya a la consola de AWS IoT Analytics y cree blocs de notas en la instancia del bloc de notas.
8. Espere a que la instancia del bloc de notas esté en el estado "In service" (En servicio).
9. Seleccione create notebooks (Crear blocs de notas) y seleccione la instancia del bloc de notas que ha creado. Esto crea un cuaderno de Jupyter con la plantilla seleccionada que puede obtener acceso a sus conjuntos de datos.

¿Por qué no puedo crear blocs de notas en una instancia?

- Asegúrese de crear una instancia del bloc de notas con la política de IAM correcta. (Siga los pasos de la pregunta anterior).
- Asegúrese de que la instancia del bloc de notas esté en el estado "In service" (En servicio). Al crear una instancia, esta comienza con el estado "Pending” (Pendiente). Suele tardar alrededor de

cinco minutos en pasar al estado "In Service” (En servicio). Si la instancia del bloc de notas pasa al estado "Failed” (Error) después de unos cinco minutos, comprobar los permisos de nuevo.

¿Por qué no veo mis conjuntos de datos en Amazon QuickSight?

Es posible que Amazon QuickSight necesite permiso para leer el contenido de su conjunto de datos de AWS IoT Analytics. Para conceder permiso, siga estos pasos.

1. Seleccione el nombre de su cuenta en la esquina superior derecha de Amazon QuickSight y seleccione Administrar QuickSight.
2. En el panel de navegación izquierdo, seleccione Seguridad y permisos. En Acceso a los servicios de AWS, compruebe que se concede el acceso a AWS IoT Analytics.
   a. Si AWS IoT Analytics no tiene acceso, seleccione Agregar o quitar.
   b. Seleccione la casilla situada junto a AWS IoT Analytics, y, a continuación, seleccione Actualizar. Esto concede permisos a Amazon QuickSight para leer el contenido del conjunto de datos.
3. Inténtelo de nuevo para visualizar sus datos.

Asegúrese de elegir la misma región de AWS para AWS IoT Analytics y Amazon QuickSight. De lo contrario, podría tener problemas para acceder a los recursos de AWS. Para ver la lista de regiones compatibles, consulte puntos de conexión y cuotas de AWS IoT Analytics y puntos de conexión y cuotas de Amazon QuickSight en Referencia general de Amazon Web Services.

¿Por qué no veo el botón de inclusión en contenedores en mi cuaderno de Jupyter existente?

- Esto se debe a que falta un complemento de creación de contenedores de AWS IoT Analytics. Si ha creado la instancia del bloc de notas de SageMaker antes del 23 de agosto de 2018, debe instalar manualmente el complemento siguiendo las instrucciones que figuran en Inclusión de un bloc de notas en contenedores.
- Si no ve el botón de inclusión en contenedores después de crear la instancia del bloc de notas de SageMaker desde la consola de AWS IoT Analytics o de haberlo instalado manualmente, póngase en contacto con el soporte técnico de AWS IoT Analytics.

¿Por qué da error la instalación de mi complemento de creación de contenedores?

- Normalmente, la instalación del complemento produce errores debido a una falta de permisos en la instancia del bloc de notas de SageMaker. Para obtener información sobre los permisos necesarios para la instancia del bloc de notas, consulte Permisos y añada los permisos necesarios al rol de instancia del bloc de notas. Si el problema persiste, cree una nueva instancia del bloc de notas desde la consola de AWS IoT Analytics.
- Puede hacer caso omiso del siguiente mensaje del registro si aparece durante la instalación del complemento: "To initialize this nbextension in the browser every time the notebook (or other app) loads" (Para inicializar esta nbextension en el navegador cada vez que se cargue el bloc de notas [u otra aplicación]).

¿Por qué da error mi complemento de creación de contenedores?

- La creación de contenedores puede fallar y generar errores por varios motivos. Asegúrese de que está utilizando el kernel correcto antes de incluir el bloc de notas en contenedores. Los kernels en contenedores comienzan por el prefijo "Containerized" (En contenedores).
- Dado que el complemento crea y guarda una imagen de Docker en un repositorio de ECR, asegúrese de que el rol de instancia del bloc de notas tiene permisos suficientes para leer, mostrar y crear repositorios de ECR. Para obtener información sobre los permisos necesarios para la instancia del bloc de notas, consulte Permisos y añada los permisos necesarios al rol de instancia del bloc de notas.
- Asimismo, asegúrese de que el nombre del repositorio cumple con los requisitos de ECR. Los nombres de repositorio de ECR deben comenzar por una letra y solo pueden contener letras minúsculas, números, guiones, guiones bajos y barras inclinadas.
- Si el proceso de creación de contenedores falla con el error: "This instance has insufficient free space to run containerization" (Esta instancia no tiene suficiente espacio libre para ejecutar la creación de contenedores), intente utilizar una instancia de mayor tamaño para solucionar el problema.
- Si ve errores de conexión o un error de creación de imágenes, vuelva a intentarlo. Si el problema persiste, reinicie la instancia e instale la última versión del complemento.

¿Por qué no veo las variables durante la creación de contenedores?

- El complemento de creación de contenedores de AWS IoT Analytics reconoce automáticamente todas las variables de su bloc de notas después de ejecutarlo con el kernel "Containerized" (En contenedores). Utilice uno de los kernels en contenedores para ejecutar el bloc de notas y, a continuación, lleve a cabo la creación de contenedores.

¿Qué variables puedo añadir a mi contenedor como entrada?

- Puede añadir cualquier variable cuyo valor desee modificar durante el tiempo de ejecución como entrada para el contenedor. Esto le permite ejecutar el mismo contenedor con diferentes parámetros que se deben proporcionar en el momento en que se crea el conjunto de datos. El complemento Jupyter de creación de contenedores de AWS IoT Analytics simplifica este proceso reconociendo automáticamente las variables del bloc de notas y haciendo que estén disponibles como parte del proceso de creación de contenedores.

¿Cómo puedo definir la salida de mi contenedor como entrada para un análisis posterior?

- Para cada ejecución del conjunto de datos de contenedores, se crea una ubicación de S3 específica donde se pueden almacenar los artefactos ejecutados. Para obtener acceso a esta ubicación de salida, cree una variable del tipo `outputFileUriValue` en el conjunto de datos de contenedores. El valor de esta variable debe ser una ruta de S3 que se utilizará para almacenar los archivos de salida adicionales. Para acceder a estos artefactos guardados en ejecuciones posteriores, puede utilizar la API de `getDatasetContent` y seleccionar el archivo de salida adecuado necesario para la ejecución posterior.

¿Por qué genera errores mi conjunto de datos de contenedores?

- Asegúrese de que está pasando el `executionRole` correcto al conjunto de datos de contenedores. La política de confianza de `executionRole` debe incluir `iotanalytics.amazonaws.com` y `sagemaker.amazonaws.com`.

- Si aparece `AlgorithmError` como motivo del error, intente reparar manualmente el código del contenedor. Esto ocurre si hay un error en el código del contenedor o el rol de ejecución no tiene permiso para ejecutar el contenedor. Si ha llevado a cabo la inclusión en contenedores mediante el complemento Jupyter de AWS IoT Analytics, cree una instancia del bloc de notas de SageMaker nueva con el mismo rol que el executionRole del containerDataset y pruebe a ejecutar el bloc de notas manualmente. Si el contenedor se ha creado fuera del complemento Jupyter, pruebe a ejecutar manualmente el código y a limitar el permiso al executionRole.

Historial de documentos

En la siguiente tabla se describen los cambios importantes en la Guía del usuario de AWS IoT Analytics después del 3 de noviembre de 2020. Para obtener más información sobre las actualizaciones de esta documentación, puede suscribirse a una fuente RSS.

Cambio	Descripción	Fecha
Lanzamiento por región	A partir de ahora, AWS IoT Analytics está disponible en la región Asia-Pacífico (Bombay).	18 de agosto de 2021
Consulta para `JOIN`	Esta actualización le permite utilizar `JOIN` para consultar un conjunto de datos de AWS IoT Analytics.	27 de julio de 2021
Integración con el AWS IoT SiteWise	Ahora se puede utilizar AWS IoT Analytics para consultar datos de AWS IoT SiteWise.	27 de julio de 2021
Particiones personalizadas	AWS IoT Analytics por lo general, ahora admite la partición de los datos según los atributos de mensajes o los atributos añadidos a través de las actividades de canalizac ión.	14 de junio de 2021
Reprocesamiento de los mensajes de canal	Esta actualización le permite volver a procesar los datos de canal en los objetos de Amazon S3 especificados.	15 de diciembre de 2020
Esquema de Parquet	Los almacenes de datos de AWS IoT Analytics ahora	15 de diciembre de 2020

	admiten el formato de archivo Parquet.	
Monitorización con Eventos de CloudWatch	AWS IoT Analytics publica automáticamente un evento en Eventos de Amazon CloudWatch cuando se produce un error de tiempo de ejecución durante una actividad de AWS Lambda.	15 de diciembre de 2020
Notificaciones de datos atrasados	Puede utilizar esta característica para recibir notificaciones a través de Eventos de Amazon CloudWatch cuando lleguen datos atrasados.	9 de noviembre de 2020
Lanzamiento por región	Se lanzó AWS IoT Analytics en China (Pekín).	4 de noviembre de 2020

Actualizaciones anteriores

En la siguiente tabla se describen los cambios importantes en la Guía del usuario de AWS IoT Analytics antes del 4 de noviembre de 2020.

Cambio	Descripción	Fecha
Lanzamiento por región	Se lanzó AWS IoT Analytics en la región Asia Pacífico (Sídney).	16 de julio de 2020
Actualización	Documentación reorganizada.	7 de mayo de 2020

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