Guía del usuario

AWS IoT TwinMaker

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

AWS IoT TwinMaker: Guía del usuario

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

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

Table of Contents

¿Qué es AWS IoT TwinMaker? 1
Cómo funcionan 1
Componentes y conceptos clave 2
Espacio de trabajo 3
Modelo de componente de una entidad 4
Visualización 6
Empezando con AWS IoT TwinMaker 8
Crear y administrar un rol de servicio para AWS IoT TwinMaker 9
Asignar confianza 9
Permisos de Amazon S3 9
Asignar permisos a un bucket de Amazon S3 específico 11
Permisos para conectores integrados 12
Permisos para un conector a un origen de datos externo 15
Modifique el rol de IAM de su espacio de trabajo para utilizar el conector de datos Athena 17
Creación de un espacio de trabajo 18
Crear su primera entidad 21
Configuración de una AWS cuenta 24
Inscríbase en un Cuenta de AWS 24
Creación de un usuario con acceso administrativo 25
Uso y creación de tipos de componentes 27
Tipos de componentes integrados 27
Características principales de los tipos de AWS IoT TwinMaker componentes 28
Creación de definiciones de propiedades 30
Creación de funciones 31
Tipos de componentes de ejemplo 31
Alarma (abstracta) 32
Telemetría de Timestream 33
Alarma (se hereda de una alarma abstracta) 34
Ejemplos de equipamiento 35
Operaciones masivas 38
Conceptos y terminología clave 38
AWS IoT TwinMaker metadataTransferJob funcionalidad 39
Realizar operaciones de importación y exportación masivas 40
metadataTransferJob requisitos previos 41

Permisos de IAM 41
Ejecute una operación masiva 45
Gestión de errores 48
Importa plantillas de metadatos 49
AWS IoT TwinMaker metadataTransferJob ejemplos 53
AWS IoT TwinMaker esquema de trabajo de transferencia de metadatos 54
Conectores de datos 72
Conectores de datos 72
Conector de inicialización de esquemas 73
DataReaderByEntity 74
DataReaderByComponentType 75
DataReader 77
AttributePropertyValueReaderByEntity 78
DataWriter 79
Ejemplos 80
Conector de datos tabulares de Athena 89
AWS IoT TwinMaker Requisitos previos del conector de datos Athena 89
Uso del conector de datos de Athena 90
Uso de la referencia JSON del conector de datos tabulares de Athena 94
Uso del conector de datos de Athena 95
Visualice los datos tabulares de Athena en Grafana 96
AWS IoT TwinMaker conector de datos de series temporales 98
AWS IoT TwinMaker requisitos previos del conector de datos de series temporales 98
Antecedentes del conector de datos de series temporales 99
Desarrollo de un conector de datos de series temporales 100
Mejora de su conector de datos 109
Prueba del conector 110
Seguridad 110
Creación de AWS IoT TwinMaker recursos 110
Siguientes pasos 112
AWS IoT TwinMaker conector de datos de fábrica de cookies 112
Creación de AWS IoT TwinMaker escenas 118
Antes de crear escenas 118
Optimice sus recursos antes de importarlos a AWS IoT TwinMaker 118
Mejores prácticas de rendimiento en AWS IoT TwinMaker 119
Más información 119

Cargar recursos en AWS IoT TwinMaker 120
Cargue los archivos a la biblioteca de recursos mediante la consola 120
Cree sus escenas 120
Usa la navegación 3D AWS IoT TwinMaker en tus escenas 122
Agregue cámaras fijas 124
Edición mejorada 124
Colocación dirigida de los objetos de la escena 125
Selección de submodelos 125
Edite las entidades en la jerarquía de escenas 126
Agregue anotaciones a las entidades 127
Agregue superposiciones a las Etiquetas 132
Edite sus escenas 140
Agregue modelos 140
Agregar widgets 141
Agregar etiquetas. 145
Optimice su modelo 3D 145
Uso de mosaicos 3D en la escena 145
Escenas dinámicas 148
Escenas estáticas versus dinámicas 148
Tipos y entidades de componentes de la escena 149
Conceptos de escena dinámica 150
AWS IoT TwinMaker integración del kit de aplicaciones 151
Cambiar AWS IoT TwinMaker los modos de precios 152
Gráfico de conocimiento 154
AWS IoT TwinMaker conceptos básicos del gráfico de conocimiento 154
Uso del gráfico de conocimiento 155
Generar un gráfico de escena 158
AWS IoT TwinMaker requisitos previos del gráfico de escena 158
Enlace nodos 3D en su escena 159
Crear una aplicación web 162
Gráfico de conocimiento: panel Grafana 164
AWS IoT TwinMaker requisitos previos del editor de consultas 164
Gráfico de conocimiento: permisos de Grafana 164
Recursos adicionales de gráfico de conocimiento 169
Sincronización de activos con AWS IoT SiteWise 182
Uso de la sincronización de activos con AWS IoT SiteWise 182

Uso de un espacio de trabajo personalizado 183
Uso del lo TSite WiseDefaultWorkspace 188
Diferencias entre los espacios de trabajo personalizados y predeterminados 189
Recursos sincronizados desde AWS IoT SiteWise 190
Espacios de trabajo personalizados y predeterminados 190
Solo espacio de trabajo predeterminado 191
Los recursos no están sincronizados 192
Utilice entidades sincronizadas y tipos de componentes en AWS IoT TwinMaker 192
Analice el estado y los errores de la sincronización 193
Estados de los trabajos de sincronización 193
Eliminar un trabajo de sincronización 195
Límites de la sincronización de activos 197
Configuración de los paneles de Grafana 199
Configuración de CORS 200
Configuración del entorno de Grafana 201
Amazon Managed Grafana 201
Grafana autoadministrado 202
Creación de un rol de panel 203
Creación de una política de IAM 203
Suba un vídeo desde Edge 207
Añadir más permisos 207
Creación del rol de IAM del panel de control de Grafana 209
Creación de una política AWS IoT TwinMaker de reproductores de vídeo 210
Limite el acceso a sus recursos 211
Limite permisos GET 212
Limite el permiso AWS IoT SiteWise BatchPutAssetPropertyValue 213
Conecte alarmas a los paneles de Grafana 216
AWS IoT SiteWise requisitos previos de configuración de alarmas 216
Defina la función de AWS IoT SiteWise IAM del componente de alarma 217
Realice consultas y actualícelas a través de la AWS IoT TwinMaker API 218
Configura tu panel de control de Grafana para las alarmas 220
Utilice el panel de control de Grafana para la visualización de las alarmas 222
Integración de Matterport 225
Información general de la integración 226
Requisitos previos para la integración de Matterport 227
Credenciales del SDK de Matterport 229

Guarde las credenciales de Matterport en AWS Secrets Manager 230
Matterport escanea las escenas AWS IoT TwinMaker 233
Matterport en tu panel de control de Grafana AWS IoT TwinMaker 239
Integración de Matterport con el kit de aplicaciones AWS IoT 239
Transmisión de vídeo a AWS IoT TwinMaker 241
Utilice el conector perimetral para la transmisión de vídeo de Kinesis para transmitir vídeo en AWS IoT TwinMaker 241
Requisitos previos 241
Cree componentes de vídeo para las escenas AWS IoT TwinMaker 242
Agregue videos y metadatos de la secuencia de vídeo de Kinesis a un panel de Grafana 243
Uso de la biblioteca Flink de AWS IoT TwinMaker 245
Registro y monitoreo 246
Monitoreo con métricas de Amazon CloudWatch 247
Métricas 247
Registrar llamadas a la API con AWS CloudTrail 250
Información de AWS IoT TwinMaker en CloudTrail 251
Seguridad 253
Protección de los datos 254
Cifrado en reposo 255
Cifrado en tránsito 255
Identity and Access Management 255
Público 256
Autenticación con identidades 257
Administración de acceso mediante políticas 260
¿Cómo AWS IoT TwinMaker funciona con IAM 263
Ejemplos de políticas basadas en identidades 270
Solución de problemas 273
Cómo utilizar roles vinculados a servicios 275
AWS políticas gestionadas 278
Puntos de conexión de VPC (AWS PrivateLink) 282
Consideraciones sobre los puntos AWS IoT TwinMaker finales de VPC 283
Creación de un punto de conexión de VPC de interfaz para AWS IoT TwinMaker 284
Acceso a AWS IoT TwinMaker través de un punto final de VPC de interfaz 285
Crear una política de puntos de conexión de VPC para AWS IoT TwinMaker 287
Validación de la conformidad 288
Resiliencia 289

Seguridad de infraestructuras 289
Cuotas y puntos de conexión 291
AWS IoT TwinMaker puntos finales y cuotas 291
Información adicional sobre los puntos de conexión 291
Historial de documentos 292
..... ccxciii

¿Qué es AWS IoT TwinMaker?

AWS IoT TwinMaker es un AWS IoT servicio que puede utilizar para crear gemelos digitales operativos de sistemas físicos y digitales. AWS IoT TwinMaker crea visualizaciones digitales mediante mediciones y análisis de una variedad de sensores, cámaras y aplicaciones empresariales del mundo real para ayudarlo a realizar un seguimiento de su fábrica, edificio o planta industrial física. Puede utilizar estos datos del mundo real para supervisar las operaciones, diagnosticar y corregir errores y optimizar las operaciones.

Un gemelo digital es una representación digital en vivo de un sistema y de todos sus componentes físicos y digitales. Se actualiza de forma dinámica con datos para imitar la estructura, el estado y el comportamiento reales del sistema. Puede utilizarlos para impulsar los resultados empresariales.

Los usuarios finales interactúan con los datos de su gemelo digital mediante una aplicación de interfaz de usuario.

Cómo funcionan

Para cumplir con los requisitos mínimos para crear un gemelo digital, tendrá que hacer lo siguiente.

- Modelar dispositivos, equipos, espacios y procesos en una ubicación física.
- Conectar estos modelos a orígenes de datos que almacenan información contextual importante, como los datos de los sensores de la señal de las cámaras.
- Crear visualizaciones que ayuden a los usuarios a comprender los datos y la información para tomar decisiones empresariales de forma más eficiente.
- Hacer que los gemelos digitales estén disponibles para los usuarios finales a fin de impulsar los resultados empresariales.

AWS IoT TwinMaker aborda estos desafíos al proporcionar las siguientes capacidades.

- Gráfico de conocimiento del sistema de componentes de la entidad: AWS IoT TwinMaker proporciona herramientas para modelar dispositivos, equipos, espacios y procesos en un gráfico de conocimiento.

  Este gráfico de conocimiento contiene metadatos sobre el sistema y puede conectarse a datos en diferentes ubicaciones. AWS IoT TwinMaker proporciona conectores integrados para los

datos almacenados en Kinesis Video Streams AWS IoT SiteWise y Kinesis. También puede crear conectores personalizados para los datos almacenados en otras ubicaciones.

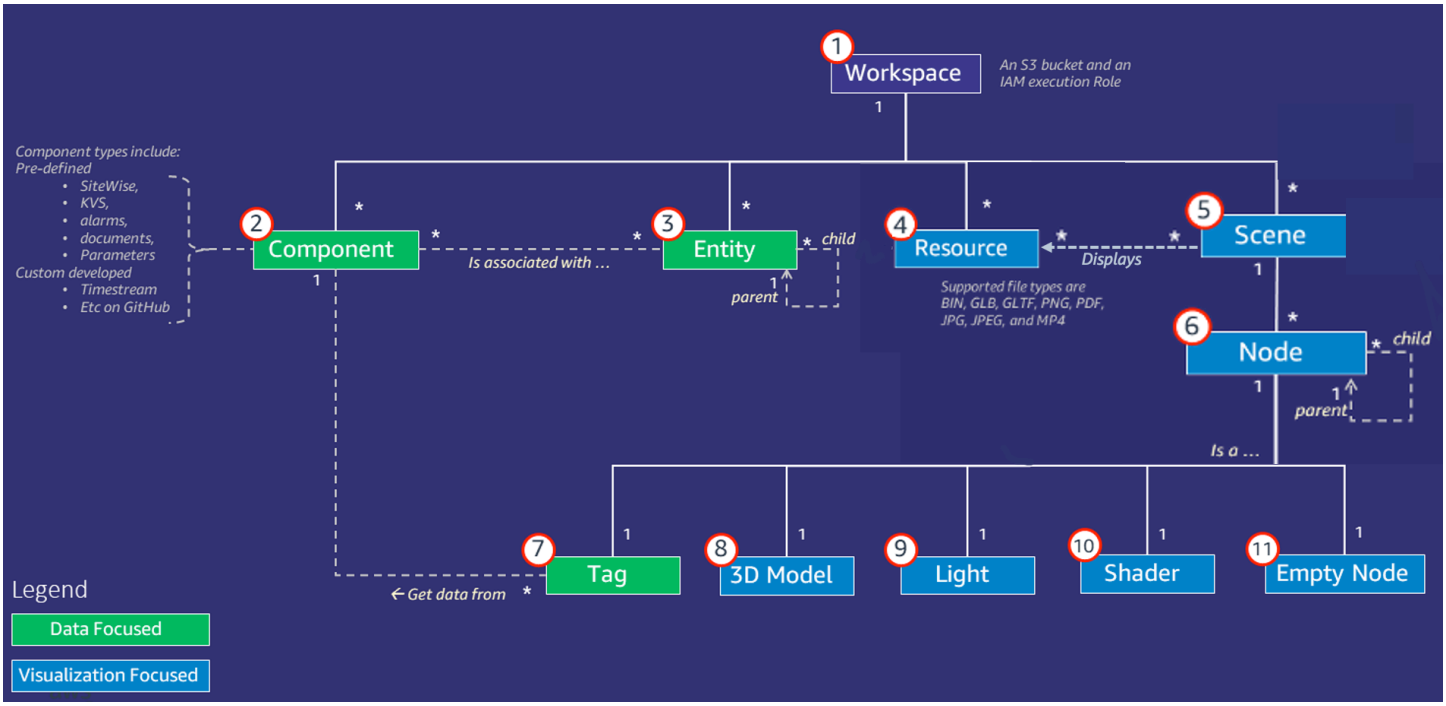
El gráfico de conocimiento y los conectores juntos proporcionan una interfaz única para consultar datos en ubicaciones dispares.

- Compositor de escenas: la AWS IoT TwinMaker consola proporciona una herramienta de composición de escenas para crear escenas en 3D. Puede cargar sus modelos 3D/CAD creados anteriormente, optimizados para su visualización en la web y convertidos al formato .gltf o .glb. A continuación, utilice el compositor de escenas para colocar varios modelos en una sola escena y así crear representaciones visuales de sus operaciones.

  También puede superponer datos en la escena. Por ejemplo, puede crear una etiqueta en una ubicación de escena que se conecte a los datos de temperatura de un sensor. Esto asocia los datos con la ubicación.
- Aplicaciones: AWS IoT TwinMaker proporciona un complemento para Grafana y Amazon Managed Grafana que puede utilizar para crear aplicaciones de paneles de control para los usuarios finales.
- Herramientas de terceros: Mendix se asocia AWS IoT TwinMaker para ofrecer soluciones completas para el IoT industrial. Consulte el taller Lean Daily Management Application con Mendix y comience AWS IoT TwinMaker a utilizar la plataforma de desarrollo de aplicaciones de bajo código (LCAP) de Mendix con AWS servicios como AWS IoT TwinMaker Kinesis Video Streams y. AWS IoT SiteWise

Componentes y conceptos clave

El siguiente diagrama ilustra cómo se combinan los conceptos clave de. AWS IoT TwinMaker



> **Note**
>
> Los asteriscos (*) del diagrama indican relaciones. one-to-many Para conocer las cuotas de cada una de estas relaciones, consulte puntos de conexión y cuotas de AWS IoT TwinMaker.

En las secciones siguientes se describen los conceptos ilustrados en el diagrama.

Espacio de trabajo

Un espacio de trabajo es un contenedor de nivel superior para su aplicación gemela digital. En este espacio de trabajo, se crean un conjunto lógico de entidades, componentes, activos de escena y otros recursos para su gemelo digital. También sirve como límite de seguridad para administrar el acceso a la aplicación gemela digital y a los recursos que contiene. Cada espacio de trabajo está vinculado al bucket de Amazon S3 donde se almacenan los datos de su espacio de trabajo. Se utilizan los roles de IAM para restringir el acceso a su espacio de trabajo.

Un espacio de trabajo puede contener varios componentes, entidades, escenas y recursos. Un tipo de componente, entidad, escena o recurso solo existe dentro de un espacio de trabajo.

Modelo de componente de una entidad

AWS IoT TwinMaker proporciona herramientas que se utilizan para modelar el sistema mediante un gráfico de entity-component-based conocimiento. Puede utilizar la arquitectura de componentes de entidad para crear una representación de su sistema físico. Este modelo de componentes de entidades consta de entidades, componentes y relaciones. Para obtener más información sobre los sistemas de componentes de entidades, consulte Sistema de componentes de entidades.

Entidad

Las entidades son representaciones digitales de los elementos de un gemelo digital que capturan las capacidades de ese elemento. Este elemento puede ser una parte del equipo físico, de un concepto o de un proceso. Las entidades tienen componentes asociados a ellas. Estos componentes proporcionan datos y contexto para la entidad asociada.

Con él AWS IoT TwinMaker, puede organizar las entidades en jerarquías personalizadas para una administración más eficiente. La vista por defecto del sistema de entidades y componentes es jerárquica.

Componente

Los componentes proporcionan contexto y datos a las entidades de una escena. Usted agrega los componentes a las entidades. La vida útil de un componente está vinculada al ciclo de vida de una entidad.

Los componentes pueden agregar datos estáticos, como una lista de documentos o las coordenadas de una ubicación geográfica. También pueden tener funciones que se conecten a otros sistemas, incluidos los sistemas que contienen datos de series temporales AWS IoT SiteWise y otros historiadores de nubes de series temporales.

Los componentes se definen mediante documentos JSON que describen la conexión entre un origen de datos e AWS IoT TwinMaker. Los componentes pueden describir fuentes de datos externas o fuentes de datos integradas en ellas. AWS IoT TwinMaker Un componente accede a un origen de datos externo mediante una función de Lambda que se especifica en el documento JSON. Un espacio de trabajo puede contener muchos componentes. Los componentes proporcionan datos a las etiquetas a través de las entidades asociadas.

AWS IoT TwinMaker proporciona varios componentes integrados que se pueden añadir desde la consola. También puede crear sus propios componentes personalizados para conectarse a orígenes de datos, como la telemetría de Timestream y las coordenadas geoespaciales. Algunos ejemplos son

la TimeStream telemetría, los componentes geoespaciales y los conectores a fuentes de datos de terceros, como Snowflake.

AWS IoT TwinMaker proporciona los siguientes tipos de componentes integrados para casos de uso comunes:

- Documento, como manuales de usuario o imágenes ubicadas en direcciones URL específicas.
- Series temporales, como los datos de los sensores procedentes de AWS IoT SiteWise.
- Alarmas, como las alarmas de series temporales procedentes de orígenes de datos externos.
- Video, de cámaras IP conectadas a Kinesis Video Streams.
- Componentes personalizados para conectarse a orígenes de datos adicionales. Por ejemplo, puede crear un conector personalizado para conectar sus entidades de AWS IoT TwinMaker a datos de series temporales almacenados externamente.

Origen de datos

Una fuente de datos es la ubicación de los datos de origen de su gemelo digital. AWS IoT TwinMaker admite dos tipos de fuentes de datos:

- Conectores jerárquicos, que permiten sincronizar continuamente un modelo externo con AWS IoT TwinMaker.
- Conectores de series temporales, que permiten conectarse a bases de datos de series temporales como AWS IoT SiteWise.

Propiedad

Las propiedades son los valores, tanto estáticos como respaldados por series temporales, contenidos en los componentes. Cuando se agregan componentes a las entidades, las propiedades del componente describen los detalles sobre el estado actual de la entidad.

AWS IoT TwinMaker admite tres tipos de propiedades:

- non-time-seriesPropiedades de valor único: estas propiedades suelen ser pares clave-valor estáticos y se almacenan directamente AWS IoT TwinMaker con los metadatos de la entidad asociada.
- Propiedades de series temporales: AWS IoT TwinMaker almacena una referencia al almacén de series temporales de estas propiedades. Por defecto, este es el valor más reciente.

- Propiedades de relación: estas propiedades almacenan una referencia a otra entidad o componente. Por ejemplo, `seen_by` es un componente de relación que podría relacionar una entidad de la cámara con otra entidad visualizada directamente por esa cámara.

Puede consultar los valores de las propiedades en orígenes de datos heterogéneos mediante la interfaz de consulta de datos unificada.

Visualización

Se utiliza AWS IoT TwinMaker para aumentar una representación tridimensional de su gemelo digital y, a continuación, verla en Grafana. Para crear escenas, use archivos CAD existentes u otros tipos de archivos 3D. A continuación, utilice superposiciones de datos para agregar datos relevantes para su gemelo digital.

Escenas

Las escenas son representaciones tridimensionales que proporcionan un contexto visual a los datos a los que se conectan. AWS IoT TwinMaker Las escenas se pueden crear utilizando un único modelo 3D gltf (formato de transmisión GL) o glb para todo el entorno, o mediante una composición de varios modelos. Las escenas también incluyen etiquetas para indicar los puntos de interés de la escena.

Las escenas son los contenedores de nivel superior para las visualizaciones. Una escena consta de uno o varios nodos.

Un espacio de trabajo puede contener varias escenas. Por ejemplo, un espacio de trabajo puede contener una escena para cada piso de una instalación.

Recursos

Las escenas muestran los recursos, que se muestran como nodos en la AWS IoT TwinMaker consola. Una escena puede contener muchos recursos.

Los recursos son imágenes y modelos tridimensionales basados en el `glTF` que se utilizan para crear una escena. Un recurso puede representar un solo equipo o un sitio completo.

Para colocar los recursos en una escena, debe cargar un archivo .gltf o .glb en la biblioteca de recursos de su espacio de trabajo y, a continuación, agregarlos a la escena.

Interfaz de usuario aumentada

Con AWS IoT TwinMaker ellas, puede ampliar sus escenas con superposiciones de datos que añaden contexto e información importantes, como los datos de los sensores, a las ubicaciones de la escena.

Nodos: los nodos son instancias de etiquetas, luces y modelos tridimensionales. También pueden estar vacíos para agregar estructura a la jerarquía de las escenas. Por ejemplo, puede agrupar varios nodos en un único nodo vacío.

Etiquetas: una etiqueta es un tipo de nodo que representa los datos de un componente (a través de una entidad). Una etiqueta se puede asociar con un solo componente. Una etiqueta es una anotación que se agrega a una posición de coordenadas `x,y,z` específica de una escena. La etiqueta conecta esta parte de la escena con el gráfico de conocimiento mediante una propiedad de la entidad. Puede usar una etiqueta para configurar el comportamiento o la apariencia visual de un elemento de la escena, como una alarma.

Luces: puedes agregar luces a una escena para enfocar determinados objetos o proyectar sombras sobre los objetos para indicar su ubicación física.

Modelos tridimensionales: un modelo tridimensional es una representación visual de un archivo .gltf o .glb importado como un recurso.

> **Note**
>
> AWS IoT TwinMaker no está diseñado para usarse en entornos peligrosos o sistemas críticos, ni en asociación con ellos, que puedan provocar lesiones corporales graves o la muerte o causar daños ambientales o materiales.
> La precisión de los datos recopilados mediante el uso de AWS IoT TwinMaker debe evaluarse según corresponda para su caso de uso. AWS IoT TwinMaker no debe utilizarse como sustituto de la supervisión humana de los sistemas físicos con el fin de evaluar si dichos sistemas funcionan de forma segura.

Empezando con AWS IoT TwinMaker

Los siguientes temas describen cómo hacerlo.

- Cree y configure un espacio de trabajo nuevo.
- Para crear una entidad y añadir componentes

Requisitos previos:

Para crear tu primer espacio de trabajo y escena, necesitas los siguientes AWS recursos.

- Una cuenta de AWS.
- Un rol de servicio de IAM para AWS IoT TwinMaker. Este rol se genera automáticamente de forma predeterminada al crear un nuevo AWS IoT TwinMaker espacio de trabajo en la AWS IoT TwinMaker consola.

  Si opta por no permitir que AWS IoT TwinMaker se cree automáticamente un nuevo rol de servicio de IAM, debe especificar uno que ya haya creado.

  Para obtener instrucciones sobre cómo crear o administrar este rol de servicio, consulte ???.

  Para obtener más información acerca de los roles de servicio de IAM, consulte Creación de un rol para delegar permisos a un Servicio de AWS.

  > **Important**
  >
  > Este rol de servicio debe tener una política adjunta que otorgue permiso al servicio para leer y escribir en un bucket de Amazon S3. AWS IoT TwinMaker utiliza este rol para acceder a otros servicios en su nombre. También tendrá que asignar una relación de confianza entre esta función y AWS IoT TwinMaker para que el servicio pueda asumirla. Si tu gemelo interactúa con otros AWS servicios, añade también los permisos necesarios para esos servicios.

Temas

- Crear y administrar un rol de servicio para AWS IoT TwinMaker
- Creación de un espacio de trabajo

- Crear su primera entidad
- Configuración de una AWS cuenta

Crear y administrar un rol de servicio para AWS IoT TwinMaker

AWS IoT TwinMaker requiere que utilice un rol de servicio para permitir que obtenga acceso a los recursos de otros servicios en su nombre. Este rol debe tener una relación de confianza con AWS IoT TwinMaker. Al crear un espacio de trabajo, debe asignar este rol al espacio de trabajo. Este tema contiene ejemplos de políticas que le muestran cómo configurar permisos para escenarios comunes.

Asignar confianza

La siguiente política establece una relación de confianza entre su función y AWS IoT TwinMaker. Asigna esta relación de confianza al rol que utiliza para su espacio de trabajo.

JSON

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

Permisos de Amazon S3

La siguiente política permite a su rol leer, eliminar y escribir en un bucket de Amazon S3. Los espacios de trabajo almacenan recursos en Amazon S3, por lo que los permisos de Amazon S3 son necesarios para todos los espacios de trabajo.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetBucket*",
        "s3:GetObject",
        "s3:ListBucket",
        "s3:PutObject"
      ],
      "Resource": [
        "arn:aws:s3:::*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:DeleteObject"
      ],
      "Resource": [
        "arn:aws:s3:::*/DO_NOT_DELETE_WORKSPACE_*"
      ]
    }
  ]
}
```

> **Note**
>
> Cuando creas un espacio de trabajo, AWS IoT TwinMaker crea un archivo en tu bucket de Amazon S3 que indica que lo está utilizando un espacio de trabajo. Esta política da AWS IoT TwinMaker permiso para eliminar ese archivo al eliminar el espacio de trabajo.
> AWS IoT TwinMaker coloca otros objetos relacionados con su espacio de trabajo. Es su responsabilidad eliminar estos objetos al eliminar un espacio de trabajo.

Asignar permisos a un bucket de Amazon S3 específico

Al crear un espacio de trabajo en la AWS IoT TwinMaker consola, puede optar por AWS IoT TwinMaker crear un bucket de Amazon S3 para usted. Puede encontrar información sobre este depósito mediante el siguiente AWS CLI comando.

```
aws iottwinmaker get-workspace --workspace-id workspace name
```

El siguiente ejemplo muestra el formato de los resultados de este comando.

```
{
    "arn": "arn:aws:iottwinmaker:region:account Id:workspace/workspace name",
    "creationDateTime": "2021-11-30T11:30:00.000000-08:00",
    "description": "",
    "role": "arn:aws:iam::account Id:role/service role name",
    "s3Location": "arn:aws:s3:::bucket name",
    "updateDateTime": "2021-11-30T11:30:00.000000-08:00",
    "workspaceId": "workspace name"
}
```

Para actualizar su política de modo que asigne permisos a un bucket de Amazon S3 específico, utilice el valor de*bucket name*.

La siguiente política permite a su rol leer, eliminar y escribir en un bucket de Amazon S3.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetBucket*",
        "s3:GetObject",
        "s3:ListBucket",
```

```
        "s3:PutObject"
      ],
      "Resource": [
        "arn:aws:s3:::bucket name",
        "arn:aws:s3:::bucket name/*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:DeleteObject"
      ],
      "Resource": [
        "arn:aws:s3:::iottwinmakerbucket/DO_NOT_DELETE_WORKSPACE_*"
      ]
    }
  ]
}
```

Permisos para conectores integrados

Si tu espacio de trabajo interactúa con otros AWS servicios mediante conectores integrados, debes incluir los permisos para esos servicios en esta política. Si utiliza el tipo de componente com.amazon.iotsitewise.connector, debe incluir los permisos para AWS IoT SiteWise. Para obtener más información acerca de los tipos de componentes, consulte ???.

> **Note**
>
> Si interactúa con otros AWS servicios mediante un tipo de componente personalizado, debe conceder al rol permiso para ejecutar la función Lambda que implementa la función en su tipo de componente. Para obtener más información, consulte ???.

En el siguiente ejemplo, se muestra cómo incluirlo AWS IoT SiteWise en la política.

JSON

```
{
```

```
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetBucket*",
        "s3:GetObject",
        "s3:ListBucket",
        "s3:PutObject"
      ],
      "Resource": [
        "arn:aws:s3:::bucket name",
        "arn:aws:s3:::bucket name/*"
      ]
    },
    {
        "Effect": "Allow",
        "Action": [
            "iotsitewise:DescribeAsset"
        ],
        "Resource": "asset ARN"
        },
    {
        "Effect": "Allow",
        "Action": [
            "iotsitewise:DescribeAssetModel"
        ],
        "Resource": "asset model ARN"
        },
    {
      "Effect": "Allow",
      "Action": [
        "s3:DeleteObject"
      ],
      "Resource": [
        "arn:aws:s3:::*/DO_NOT_DELETE_WORKSPACE_*"
      ]
    }
  ]
}
```

Si utilizas el tipo de componente com.amazon.iotsitewise.connector y necesitas leer los datos de las propiedades AWS IoT SiteWise, debes incluir el siguiente permiso en tu política.

```
...
{
    "Action": [
        "iotsitewise:GetPropertyValueHistory",
    ],
    "Resource": [
        "AWS IoT SiteWise asset resource ARN"
    ],
    "Effect": "Allow"
},
...
```

Si utilizas el tipo de componente com.amazon.iotsitewise.connector y necesitas escribir datos de propiedad en él, debes incluir el siguiente permiso en tu política. AWS IoT SiteWise

```
...
{
    "Action": [
        "iotsitewise:BatchPutPropertyValues",
    ],
    "Resource": [
        "AWS IoT SiteWise asset resource ARN"
    ],
    "Effect": "Allow"
},
...
```

Si utiliza el tipo de componente com.amazon.iotsitewise.connector.edgevideo, debe incluir los permisos para Kinesis Video Streams. AWS IoT SiteWise El siguiente ejemplo de política muestra cómo incluir los permisos AWS IoT SiteWise de Kinesis Video Streams en su política.

```
...
{
```

```
    "Action": [
        "iotsitewise:DescribeAsset",
        "iotsitewise:GetAssetPropertyValue"
    ],
    "Resource": [
        "AWS IoT SiteWise asset resource ARN for the Edge Connector for Kinesis Video
 Streams"
    ],
    "Effect": "Allow"
},
{
    "Action": [
        "iotsitewise:DescribeAssetModel"
    ],
    "Resource": [
        "AWS IoT SiteWise model resource ARN for the Edge Connector for Kinesis Video
 Streams"
    ],
    "Effect": "Allow"
},
{
    "Action": [
        "kinesisvideo:DescribeStream"
    ],
    "Resource": [
        "Kinesis Video Streams stream ARN"
    ],
    "Effect": "Allow"
},
...
```

Permisos para un conector a un origen de datos externo

Si crea un tipo de componente que utiliza una función que se conecta a un origen de datos externo, debe conceder permiso a su rol de servicio para usar la función de Lambda que implementa la función. Para obtener más información sobre la creación de tipos de componentes y funciones, consulte ???.

El siguiente ejemplo da permiso a su rol de servicio para usar una función de Lambda.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetBucket*",
        "s3:GetObject",
        "s3:ListBucket",
        "s3:PutObject"
      ],
      "Resource": [
        "arn:aws:s3:::bucket name",
        "arn:aws:s3:::bucket name/*"
      ]
    },
    {
        "Action": [
            "lambda:invokeFunction"
        ],
        "Resource": [
            "Lambda function ARN"
        ],
        "Effect": "Allow"
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:DeleteObject"
      ],
      "Resource": [
        "arn:aws:s3:::*/DO_NOT_DELETE_WORKSPACE_*"
      ]
    }
  ]
}
```

Para obtener más información sobre cómo crear roles y asignarles políticas y relaciones de confianza mediante la consola de IAM, la API de IAM y la AWS CLI API de IAM, consulte Crear un rol para delegar permisos a un. Servicio de AWS

Modifique el rol de IAM de su espacio de trabajo para utilizar el conector de datos Athena

Para utilizar el conector de datos tabulares AWS IoT TwinMaker Athena, debe actualizar la función de IAM de su AWS IoT TwinMaker espacio de trabajo. Añada los siguientes permisos al rol de IAM de su espacio de trabajo:

> **Note**
>
> Este cambio de IAM solo funciona para los datos tabulares de Athena almacenados en AWS Glue Amazon S3. Para usar Athena con otros orígenes de datos, debe configurar un rol de IAM para Athena, consulte Administración de identidades y acceso en Athena.

```
{
    "Effect": "Allow",
    "Action": [
        "athena:GetQueryExecution",
        "athena:GetQueryResults",
        "athena:GetTableMetadata",
        "athena:GetWorkGroup",
        "athena:StartQueryExecution",
        "athena:StopQueryExecution"
    ],
    "Resource": [
        "athena resouces arn"
    ]
},// Athena permission
{
    "Effect": "Allow",
    "Action": [
        "glue:GetTable",
        "glue:GetTables",
        "glue:GetDatabase",
        "glue:GetDatabases"
    ],
```

```
    "Resource": [
        "glue resouces arn"
    ]
},// This is an example for accessing aws glue
{
    "Effect": "Allow",
    "Action": [
        "s3:ListBucket",
        "s3:GetObject"
    ],
    "Resource": [
        "Amazon S3 data source bucket resources arn"
    ]
}, // S3 bucket for storing the tabular data.
{
    "Effect": "Allow",
    "Action": [
        "s3:GetBucketLocation",
        "s3:GetObject",
        "s3:ListBucket",
        "s3:ListBucketMultipartUploads",
        "s3:ListMultipartUploadParts",
        "s3:AbortMultipartUpload",
        "s3:CreateBucket",
        "s3:PutObject",
        "s3:PutBucketPublicAccessBlock"
    ],
    "Resource": [
        "S3 query result bucket resources arn"
    ]
} // Storing the query results
```

Lea la sección Administración de identidades y acceso en Athena para obtener más información sobre la configuración de IAM de Athena.

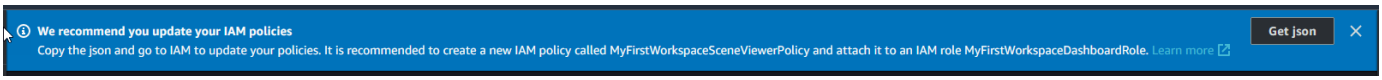
Creación de un espacio de trabajo

Para crear y configurar su primer espacio de trabajo, siga estos pasos.

> **Note**
>
> Este tema muestra cómo crear un espacio de trabajo sencillo con un único recurso. Para un espacio de trabajo completo con múltiples recursos, prueba la configuración de muestra en el repositorio de AWS IoT TwinMaker muestras de Github.

1. En la página de inicio de la consola de AWS IoT TwinMaker, seleccione Espacios de trabajo en el panel de navegación izquierdo.
2. En el panel Espacios de trabajo, elija Crear espacio de trabajo.
3. En la página Crear un espacio de trabajo, escriba un nombre para su espacio de trabajo.
4. (Opcional) Añada una descripción para su espacio de trabajo.
5. En Recurso de S3, seleccione Crear un bucket de S3. Esta opción crea un bucket de Amazon S3 en el que se AWS IoT TwinMaker almacena la información y los recursos relacionados con el espacio de trabajo. Cada espacio de trabajo requiere su propio bucket.
6. En Rol de ejecución, seleccione Generar automáticamente un nuevo rol o el rol de IAM personalizado que creó para este espacio de trabajo.

   Si elige Generar automáticamente un nuevo rol, AWS IoT TwinMaker adjunta una política al rol que otorga permiso al nuevo rol de servicio para acceder a otros AWS servicios, incluido el permiso para leer y escribir en el bucket de Amazon S3 que especificó en el paso anterior. Para obtener más información sobre cómo asignar permisos a este rol, consulte ???.
7. Seleccione Crear espacio de trabajo. El siguiente banner aparece en la parte superior de la página de espacios de trabajo.

8. Seleccione Obtener json. Le recomendamos que añada la política de IAM que ve al rol de IAM que se AWS IoT TwinMaker creó para los usuarios y las cuentas que ven el panel de control de Grafana. El nombre de este rol sigue este patrón: *`workspace-name`* DashboardRole Para obtener instrucciones sobre cómo crear una política y asociarla a un rol, consulte Modificación de la política de permisos de un rol (consola).

   El siguiente ejemplo contiene la política que se debe añadir al rol del panel de control.

JSON

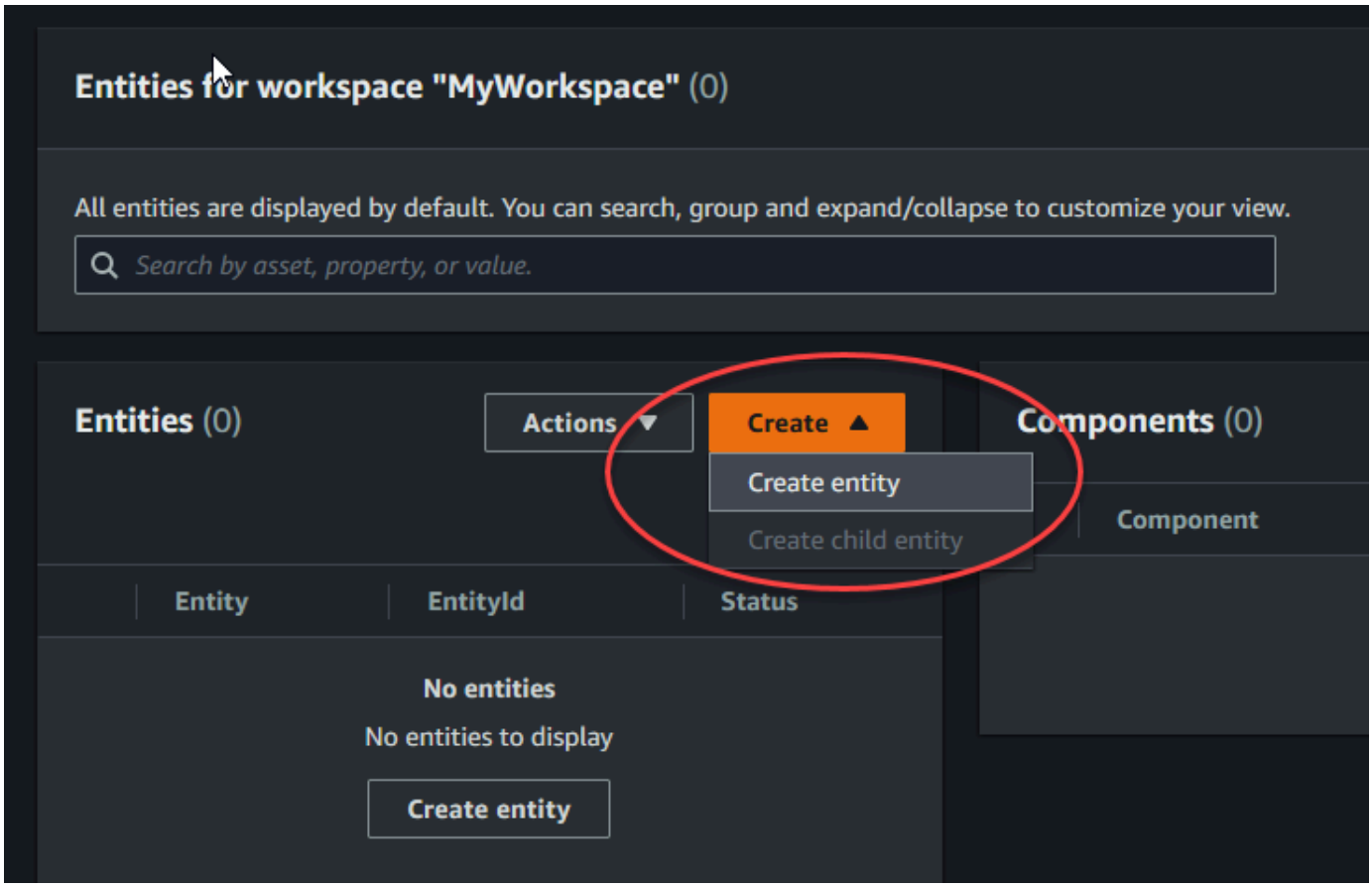
```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject"
      ],
      "Resource": [
        "arn:aws:s3:::iottwinmaker-workspace-workspace-name-lower-
case-account-id",
        "arn:aws:s3:::iottwinmaker-workspace-workspace-name-lower-
case-account-id/*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "iottwinmaker:Get*",
        "iottwinmaker:List*"
      ],
      "Resource": [
        "arn:aws:iottwinmaker:us-east-1:account-id:workspace/workspace-name",
        "arn:aws:iottwinmaker:us-east-1:account-id:workspace/workspace-name/
*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": "iottwinmaker:ListWorkspaces",
      "Resource": "*"
    }
  ]
}
```

Ya está listo para empezar a crear un modelo de datos para su espacio de trabajo con su primera entidad. Para obtener instrucciones al respecto, consulte Crear su primera entidad.

Crear su primera entidad

Para crear su primera entidad, siga los pasos que se describen a continuación.

1. En la página Espacios de trabajo, elija su espacio de trabajo y, a continuación, en el panel izquierdo, elija Entidades.
2. En la página Entidades, elija Crear y, a continuación, elija Crear entidad.



3. En la ventana Crear una entidad, introduzca un nombre para la entidad. En este ejemplo se usa una entidad **`CookieMixer`**.
4. (Opcional) Escriba una descripción para la entidad.
5. Seleccione Crear entidad.

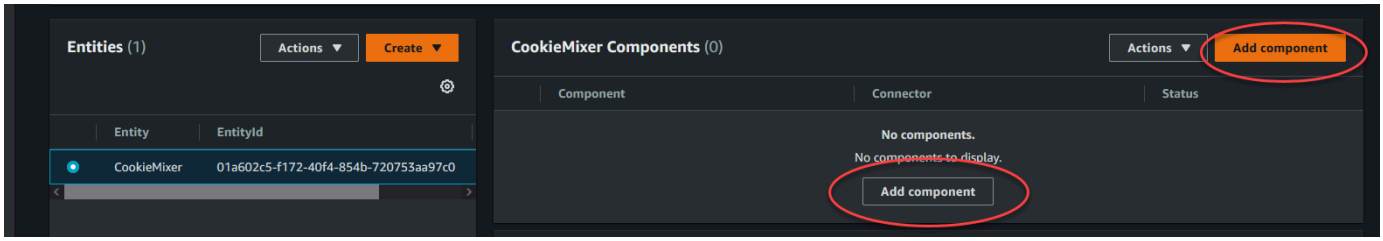
Las entidades contienen datos sobre cada elemento del espacio de trabajo. Los datos se colocan en las entidades añadiendo componentes. AWS IoT TwinMaker proporciona los siguientes tipos de componentes integrados.

- Parámetros: añade un conjunto de propiedades clave-valor.
- Documento: agrega un nombre y una URL a un documento que contiene información sobre la entidad.
- Alarmas: se conecta a una fuente de datos de series temporales de alarmas.
- SiteWise conector: extrae las propiedades de las series temporales definidas en un AWS IoT SiteWise activo.
- Conector Edge para Kinesis Video AWS IoT Greengrass Streams: extrae datos de vídeo del conector Edge para KVS. AWS IoT Greengrass Para obtener más información, consulte AWS IoT TwinMaker integración de vídeo.

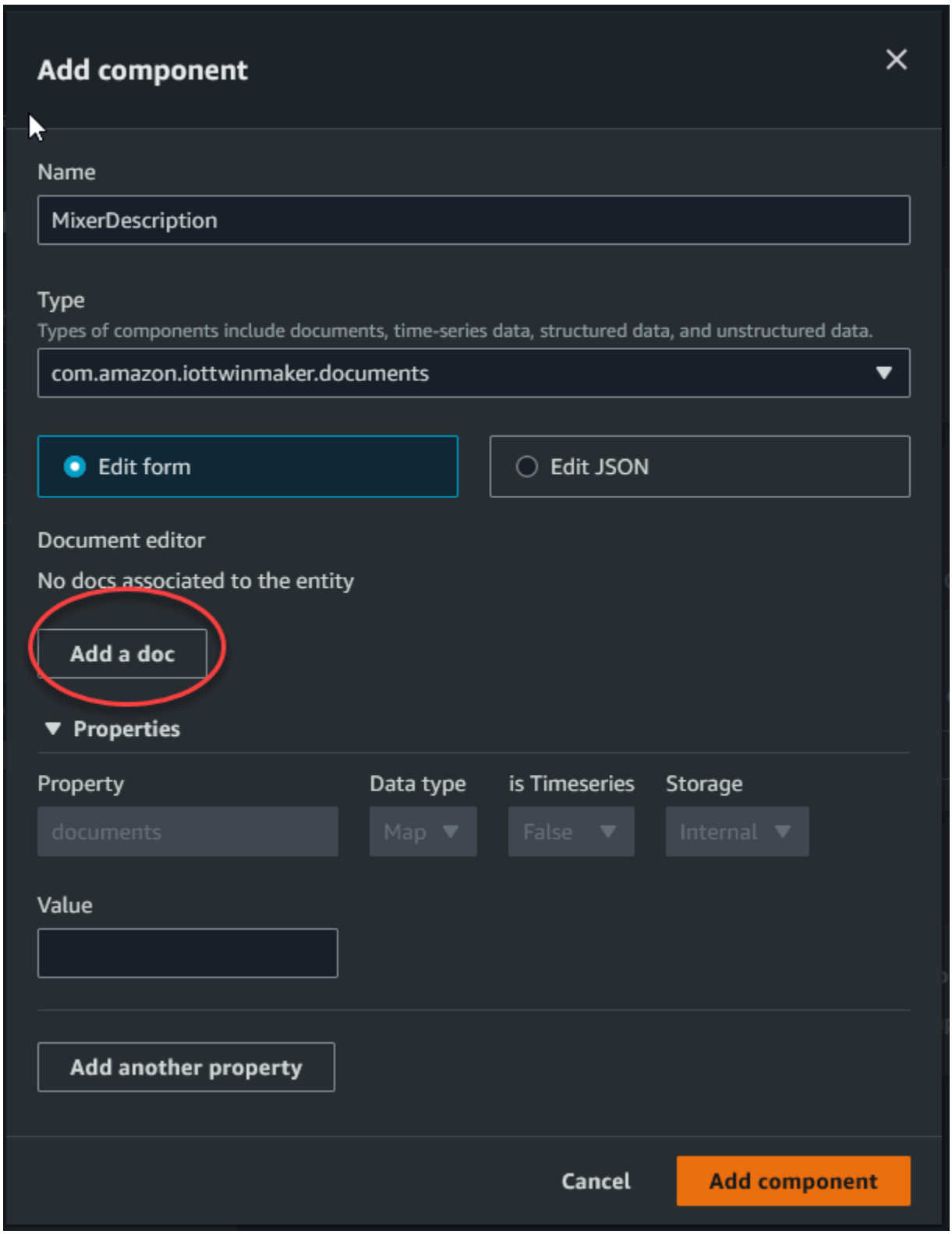
Puede ver estos tipos de componentes y sus definiciones seleccionando Tipos de componentes en el panel izquierdo. También puede crear un nuevo tipo de componente en la página de tipos de componentes. Para obtener más información acerca de la creación de tipos de componentes, consulte Uso y creación de tipos de componentes.

En este ejemplo, creamos un componente de documento simple que añade información descriptiva sobre su entidad.

1. En la página Entidades, elija la entidad y, a continuación, elija añadir componente.



2. En la ventana Añadir componente, introduzca un nombre para el componente. Como en este ejemplo se utiliza una entidad mezcladora de cookies, introducimos **`MixerDescription`** en el campo Nombre.



3. Selecciona Añadir un documento y, a continuación, introduce los valores para el nombre del documento y la URL externa. Con el componente de documentos, puede almacenar una lista de documentos externos URLs que contienen información importante sobre la entidad.

4. Elija Añadir componente.

Ahora tiene todo listo para crear su primera escena. Para obtener instrucciones al respecto, consulte Creación y edición de AWS IoT TwinMaker escenas.

Configuración de una AWS cuenta

Si no tiene una Cuenta de AWS, complete los siguientes pasos para crearla.

Para suscribirse a una Cuenta de AWS

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

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

   Cuando te registras en un Cuenta de AWS, Usuario raíz de la cuenta de AWSse crea un. 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.

Inscríbase en un Cuenta de AWS

Si no tiene uno Cuenta de AWS, complete los siguientes pasos para crearlo.

Para suscribirse a una Cuenta de AWS

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

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

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

AWS te envía un correo electrónico de confirmación una vez finalizado el proceso de registro. En cualquier momento, puede ver la actividad de su cuenta actual y administrarla accediendo a https://aws.amazon.com/y seleccionando Mi cuenta.

Creación de un usuario con acceso administrativo

Después de crear un usuario administrativo Cuenta de AWS, asegúrelo Usuario raíz de la cuenta de AWS AWS IAM Identity Center, habilite y cree un usuario administrativo para no usar el usuario root en las tareas diarias.

Proteja su Usuario raíz de la cuenta de AWS

1. Inicie sesión AWS Management Consolecomo propietario de la cuenta seleccionando el usuario root e introduciendo su dirección de Cuenta de AWS correo electrónico. En la siguiente página, escriba su contraseña.

   Para obtener ayuda para iniciar sesión con el usuario raíz, consulte Iniciar sesión como usuario raíz en la Guía del usuario de AWS Sign-In .
2. Active la autenticación multifactor (MFA) para el usuario raíz.

   Para obtener instrucciones, consulte Habilitar un dispositivo MFA virtual para el usuario Cuenta de AWS raíz (consola) en la Guía del usuario de IAM.

Creación de un usuario con acceso administrativo

1. Activar IAM Identity Center.

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

   Para ver un tutorial sobre su uso Directorio de IAM Identity Center como fuente de identidad, consulte Configurar el acceso de los usuarios con la configuración predeterminada Directorio de IAM Identity Center en la Guía del AWS IAM Identity Center usuario.

Inicio de sesión como usuario con acceso de administrador

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

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

Concesión de acceso a usuarios adicionales

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

   Para conocer las instrucciones, consulte Create a permission set en la Guía del usuario de AWS IAM Identity Center .
2. Asigne usuarios a un grupo y, a continuación, asigne el acceso de inicio de sesión único al grupo.

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

Uso y creación de tipos de componentes

En este tema se explican los valores y estructuras que se utilizan para crear un tipo de AWS IoT TwinMaker componente. En él se muestra cómo crear un objeto de solicitud que se puede pasar a la CreateComponentTypeAPI o mediante el editor de tipos de componentes de la AWS IoT TwinMaker consola.

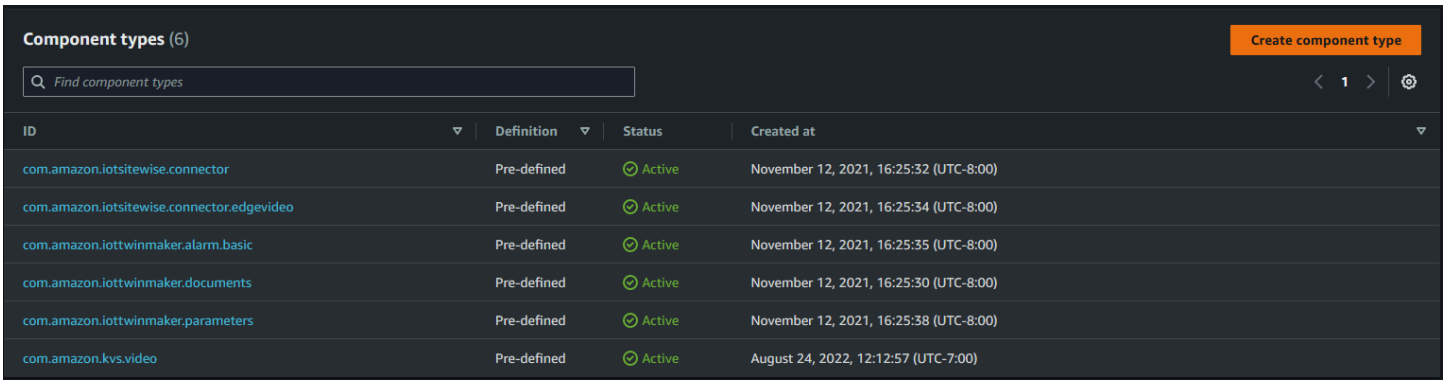
Los componentes proporcionan contexto para las propiedades y los datos de sus entidades asociadas.

Tipos de componentes integrados

En la AWS IoT TwinMaker consola, al elegir un espacio de trabajo y, a continuación, elegir Tipos de componentes en el panel izquierdo, aparecen los siguientes tipos de componentes.

- com.amazon.iotsitewise.resourcesync: tipo de componente que sincroniza automáticamente los activos y modelos de AWS IoT SiteWise activos y los convierte en componentes y tipos de componentes de entidades. AWS IoT TwinMaker Para obtener más información sobre el uso de la sincronización de activos, consulta Sincronización de activos con. AWS IoT SiteWiseAWS IoT SiteWise
- com.amazon.iottwinmaker.alarm.basic: es un componente de alarma básico que extrae datos de la alarma de un origen externo a una entidad. Este componente no contiene una función que se conecte a un origen de datos específico. Esto significa que el componente de la alarma es abstracto y lo puede heredar otro tipo de componente que especifique un origen de datos y una función que lea desde ese origen.
- com.amazon.iottwinmaker.documents: es una asignación sencilla de títulos a URL para documentos que contienen información sobre una entidad.
- com.amazon.iotsitewise.connector.edgevideo: componente que extrae vídeo de un dispositivo IoT mediante el componente Edge Connector for Kinesis Video Streams a una entidad. AWS IoT Greengrass El componente Edge Connector para Kinesis Video AWS IoT Greengrass Streams no es AWS IoT TwinMaker un componente, sino un componente AWS IoT Greengrass prediseñado que se implementa localmente en su dispositivo de IoT.
- com.amazon.iotsitewise.connector: es un componente que introduce datos de AWS IoT SiteWise en una entidad.
- com.amazon.iottwinmaker.parameters: es un componente que agrega pares clave-valor estáticos a una entidad.

- com.amazon.kvs.video: componente que extrae vídeo de Kinesis Video Streams a una entidad. AWS IoT TwinMaker



Características principales de los tipos de AWS IoT TwinMaker componentes

En la siguiente lista se describen las principales características de los tipos de componentes.

- Definiciones de propiedades: el PropertyDefinitionRequestobjeto define una propiedad que se puede rellenar en el compositor de escenas o se puede rellenar con datos extraídos de fuentes de datos externas. Las propiedades estáticas que establezca se almacenan en AWS IoT TwinMaker. Las propiedades de las series temporales y otras propiedades que se extraen de los orígenes de datos se almacenan de forma externa.

  Las definiciones de propiedades se especifican dentro de una cadena en el mapa `PropertyDefinitionRequest`. Cada cadena de la asignación debe ser exclusiva del mapa.

- Funciones: el FunctionRequestobjeto especifica una función Lambda que lee y, potencialmente, escribe en una fuente de datos externa.

  Un tipo de componente que contiene una propiedad con un valor almacenado externamente, pero que no tiene una función correspondiente para recuperar los valores es un tipo de componente abstracto. Puede ampliar tipos de componentes concretos a partir de un tipo de componente abstracto. No puede agregar tipos de componentes abstractos a una entidad. No aparecen en el compositor de escenas.

  Las funciones se especifican dentro de una cadena a un mapa `FunctionRequest`. La cadena debe especificar uno de los siguientes tipos de función predefinidos.

- `dataReader`: es una función que extrae datos de un origen externo.
- `dataReaderByEntity`: es una función que extrae datos de un origen externo.

  Cuando se utiliza este tipo de lector de datos, la operación de la GetPropertyValueHistoryAPI solo admite consultas específicas de la entidad para las propiedades de este tipo de componente. (Solo puede solicitar el historial de valores de las propiedades para `componentName` + `entityId`).
- `dataReaderByComponentType`: es una función que extrae datos de un origen externo.

  Cuando se utiliza este tipo de lector de datos, la operación de la GetPropertyValueHistoryAPI solo admite consultas entre entidades para las propiedades de este tipo de componente. (Solo puede solicitar el historial de valores de la propiedad para `componentTypeId`).
- `dataWriter`: es una función que escribe datos en un origen externo.
- `schemaInitializer`: es una función que inicializa automáticamente los valores de las propiedades cada vez que se crea una entidad que contiene el tipo de componente.

En un tipo de componente no abstracto se requiere uno de los tres tipos de funciones de lectura de datos.

Para ver un ejemplo de una función de Lambda que implementa componentes de telemetría de Timestream, incluidas las alarmas, consulte el lector de datos en Muestras de AWS IoT TwinMaker.

> **Note**
>
> Como el conector de la alarma hereda del tipo de componente de la alarma abstracta, la función de Lambda debe devolver el valor `alarm_key`. Si no devuelve este valor, Grafana no lo reconocerá como una alarma. Esto es obligatorio para todos los componentes que devuelven alarmas.

- Herencia: los tipos de componentes promueven la reutilización del código mediante la herencia. Un tipo de componente puede heredar hasta 10 tipos de componentes principales.

  Utilice el parámetro `extendsFrom` para especificar los tipos de componentes de los que su tipo de componente hereda propiedades y funciones.
- isSingleton: algunos componentes contienen propiedades, como las coordenadas de ubicación, que no se pueden incluir más de una vez en una entidad. Establezca el valor del parámetro

`isSingleton` como `true` para indicar que su tipo de componente solo se puede incluir una vez en una entidad.

Creación de definiciones de propiedades

En la tabla siguiente se describen los parámetros de una `PropertyDefinitionRequest`.

Parámetro	Descripción
`isExternalId`	Un booleano que especifica si la propiedad es un identificador único (como un identificador de AWS IoT SiteWise activo) del valor de una propiedad que se almacena externamente. El valor predeterminado de esta propiedad es `false`.
`isStoredExternally`	Un valor booleano que especifica si la propiedad está almacenada externamente. El valor predeterminado de esta propiedad es `false`.
`isTimeSeries`	Un valor booleano que especifica si la propiedad almacena datos de serie temporal. El valor predeterminado de esta propiedad es `false`
`isRequiredInEntity`	Un booleano que especifica si la propiedad debe tener un valor en una entidad que utilice el tipo de componente.
`dataType`	DataTypeObjeto que especifica el tipo de datos (como cadena, mapa, lista y unidad de medida) de la propiedad.

Parámetro	Descripción
`defaultValue`	DataValueObjeto que especifica el valor predeterminado de la propiedad.
`configuration`	Un string-to-string mapa que especifica la información adicional que necesita para conectarse a una fuente de datos externa.

Creación de funciones

En la tabla siguiente se describen los parámetros de una `FunctionRequest`.

Parámetro	Descripción
`implementedBy`	DataConnectorObjeto que especifica la función Lambda que se conecta a la fuente de datos externa.
`requiredProperties`	Una lista de propiedades que la función necesita para leer y escribir en un origen de datos externo.
`scope`	El ámbito de una función. Utilice `Workspace` para funciones con un alcance que abarca todo un espacio de trabajo. Utilice `Entity` para funciones con un alcance limitado a la entidad que contiene el componente.

Para ver ejemplos que muestran cómo crear y extender tipos de componentes, consulte ???.

Tipos de componentes de ejemplo

Este tema contiene ejemplos que muestran cómo implementar los conceptos clave de los tipos de componentes.

Alarma (abstracta)

El siguiente ejemplo es el tipo de componente de alarma abstracto que aparece en la AWS IoT TwinMaker consola. Contiene una lista de `functions` que consta de un `dataReader` que no tiene ningún valor `implementedBy`.

```
{
  "componentTypeId": "com.example.alarm.basic:1",
  "workspaceId": "MyWorkspace",
  "description": "Abstract alarm component type",
  "functions": {
    "dataReader": {
         "isInherited": false
    }
  },
  "isSingleton": false,
  "propertyDefinitions": {
    "alarm_key": {
      "dataType": { "type": "STRING" },
      "isExternalId": true,
      "isRequiredInEntity": true,
      "isStoredExternally": false,
      "isTimeSeries": false
    },
    "alarm_status": {
      "dataType": {
        "allowedValues": [
          {
            "stringValue": "ACTIVE"
          },
          {
            "stringValue": "SNOOZE_DISABLED"
          },
          {
            "stringValue": "ACKNOWLEDGED"
          },
          {
            "stringValue": "NORMAL"
          }
        ],
        "type": "STRING"
      },
```

```
      "isRequiredInEntity": false,
      "isStoredExternally": true,
      "isTimeSeries": true
    }
  }
}
```

Notas:

Los valores para `componentTypeId` y `workspaceID` son obligatorios. El valor de `componentTypeId` debe ser exclusivo de su espacio de trabajo. El valor de `alarm_key` es un identificador único que una función puede utilizar para recuperar datos de la alarma de un origen externo. El valor de la clave es obligatorio y se almacena en AWS IoT TwinMaker. Los valores `alarm_status` de las series temporales se almacenan en el origen externo.

Hay más ejemplos disponibles en Muestras de AWS IoT TwinMaker.

Telemetría de Timestream

El siguiente ejemplo es un tipo de componente simple que recupera datos de telemetría sobre un tipo específico de componente (como una alarma o un mezclador de galletas) de una fuente externa. Especifica una función de Lambda que hereda los tipos de componentes.

```
{
    "componentTypeId": "com.example.timestream-telemetry",
    "workspaceId": "MyWorkspace",
    "functions": {
        "dataReader": {
            "implementedBy": {
                "lambda": {
                    "arn": "lambdaArn"
                }
            }
        }
    },
    "propertyDefinitions": {
        "telemetryType": {
            "dataType": { "type": "STRING" },
            "isExternalId": false,
            "isStoredExternally": false,
```

```
            "isTimeSeries": false,
            "isRequiredInEntity": true
        },
        "telemetryId": {
            "dataType": { "type": "STRING" },
            "isExternalId": false,
            "isStoredExternally": false,
            "isTimeSeries": false,
            "isRequiredInEntity": true
        }
    }
}
```

Alarma (se hereda de una alarma abstracta)

En el siguiente ejemplo se heredan los tipos de componentes de la alarma abstracta y de telemetría de Timestream. Especifica su propia función de Lambda que recupera los datos de la alarma.

```
{
    "componentTypeId": "com.example.cookiefactory.alarm",
    "workspaceId": "MyWorkspace",
    "extendsFrom": [
        "com.example.timestream-telemetry",
        "com.amazon.iottwinmaker.alarm.basic"
    ],
    "propertyDefinitions": {
        "telemetryType": {
            "defaultValue": {
                "stringValue": "Alarm"
            }
        }
    },
    "functions": {
        "dataReader": {
            "implementedBy": {
                "lambda": {
                    "arn": "lambdaArn"
                }
            }
        }
    }
```

```
}
```

> **Note**
>
> Como el conector de la alarma hereda del tipo de componente de la alarma abstracta, la función de Lambda debe devolver el valor `alarm_key`. Si no devuelve este valor, Grafana no lo reconocerá como una alarma. Esto es obligatorio para todos los componentes que devuelven alarmas.

Ejemplos de equipamiento

En los ejemplos en esta sección se muestra cómo modelar posibles equipos. Puede utilizar estos ejemplos para obtener algunas ideas sobre cómo modelar equipos en sus propios procesos.

Mezclador de cookies

En el siguiente ejemplo se hereda del tipo de componente de telemetría de Timestream. Especifica propiedades de series temporales adicionales para la velocidad de rotación y la temperatura del mezclador de cookies.

```
{
    "componentTypeId": "com.example.cookiefactory.mixer",
    "workspaceId": "MyWorkspace",
    "extendsFrom": [
        "com.example.timestream-telemetry"
    ],
    "propertyDefinitions": {
        "telemetryType": {
            "defaultValue" : { "stringValue": "Mixer" }
        },
        "RPM": {
            "dataType": { "type": "DOUBLE" },
            "isTimeSeries": true,
            "isStoredExternally": true
        },
        "Temperature": {
            "dataType": { "type": "DOUBLE" },
            "isTimeSeries": true,
```

```
            "isStoredExternally": true
        }
    }
}
```

Depósito de agua

En el siguiente ejemplo se hereda del tipo de componente de telemetría de Timestream. Especifica propiedades de series temporales adicionales para el volumen y el caudal de un depósito de agua.

```
{
    "componentTypeId": "com.example.cookiefactory.watertank",
    "workspaceId": "MyWorkspace",
    "extendsFrom": [
        "com.example.timestream-telemetry"
    ],
    "propertyDefinitions": {
        "telemetryType": {
            "defaultValue" : { "stringValue": "WaterTank" }
        },
        "tankVolume1": {
            "dataType": { "type": "DOUBLE" },
            "isTimeSeries": true,
            "isStoredExternally": true
        },
        "tankVolume2": {
            "dataType": { "type": "DOUBLE" },
            "isTimeSeries": true,
            "isStoredExternally": true
        },
        "flowRate1": {
            "dataType": { "type": "DOUBLE" },
            "isTimeSeries": true,
            "isStoredExternally": true
        },
        "flowrate2": {
            "dataType": { "type": "DOUBLE" },
            "isTimeSeries": true,
            "isStoredExternally": true
        }
    }
```

```
}
```

Ubicación del espacio

El siguiente ejemplo contiene propiedades, cuyos valores se almacenan en. AWS IoT TwinMaker Como los valores los especifican los usuarios y se almacenan internamente, no es necesaria ninguna función para recuperarlos. En el ejemplo también se usa el tipo de datos `RELATIONSHIP` para especificar una relación con otro tipo de componente.

Este componente proporciona un mecanismo ligero para agregar contexto a un gemelo digital. Puede usarlo para agregar metadatos que indiquen dónde se encuentra algo. También puede usar esta información en la lógica que se utiliza para determinar qué cámaras pueden ver un equipo o un espacio, o para saber cómo enviar a alguien a un lugar.

```
{
    "componentTypeId": "com.example.cookiefactory.space",
    "workspaceId": "MyWorkspace",
    "propertyDefinitions": {
        "position":  {"dataType": {"nestedType": {"type": "DOUBLE"},"type": "LIST"}},
        "rotation":  {"dataType": {"nestedType": {"type": "DOUBLE"},"type": "LIST"}},
        "bounds":  {"dataType": {"nestedType": {"type": "DOUBLE"},"type": "LIST"}},
        "parent_space" : { "dataType": {"type": "RELATIONSHIP"}}
    }
}
```

AWS IoT TwinMaker operaciones masivas

Utilice a metadataTransferJob para transferir y gestionar sus AWS IoT TwinMaker recursos a gran escala. A metadataTransferJob le permite realizar operaciones masivas y transferir recursos entre AWS IoT TwinMaker AWS IoT SiteWise y Amazon S3.

Puede utilizar operaciones masivas en los siguientes escenarios:

- Migración masiva de activos y datos entre cuentas, por ejemplo, la migración de una cuenta de desarrollo a una cuenta de producción.
- Gestión de activos a gran escala, como la carga y edición de AWS IoT activos a gran escala.
- Importación masiva de sus activos a AWS IoT TwinMaker y AWS IoT SiteWise.
- Importación masiva de AWS IoT TwinMaker entidades desde archivos de ontología existentes, como `BIM` archivos `revit` o.

Temas

- Conceptos y terminología clave
- Realizar operaciones de importación y exportación masivas
- AWS IoT TwinMaker esquema de trabajo de transferencia de metadatos

Conceptos y terminología clave

AWS IoT TwinMaker las operaciones masivas utilizan los siguientes conceptos y terminología:

- Importación: acción de mover recursos a un espacio de AWS IoT TwinMaker trabajo. Por ejemplo, desde un archivo local, un archivo de un bucket de Amazon S3 o desde AWS IoT SiteWise un AWS IoT TwinMaker espacio de trabajo.
- Exportación: acción de mover recursos de un AWS IoT TwinMaker espacio de trabajo a una máquina local o a un bucket de Amazon S3.
- Origen: la ubicación inicial desde la que desea mover los recursos.

  Por ejemplo, un bucket de Amazon S3 es una fuente de importación y un AWS IoT TwinMaker espacio de trabajo es una fuente de exportación.
- Destino: la ubicación deseada a la que desea mover sus recursos.

Por ejemplo, un bucket de Amazon S3 es un destino de exportación y un AWS IoT TwinMaker espacio de trabajo es un destino de importación.

- AWS IoT SiteWise Esquema: esquema que se utiliza para importar y exportar recursos desde y hacia AWS IoT SiteWise.
- AWS IoT TwinMaker Esquema: esquema que se utiliza para importar y exportar recursos hacia y desde AWS IoT TwinMaker.
- AWS IoT TwinMaker recursos de nivel superior: recursos utilizados en los existentes APIs. Específicamente, una entidad o un ComponentType.
- AWS IoT TwinMaker recursos de subnivel: tipos de recursos anidados que se utilizan en las definiciones de metadatos. En concreto, un componente.
- Metadatos: información clave necesaria para importar o exportar correctamente AWS IoT SiteWise y AWS IoT TwinMaker recursos.
- metadataTransferJob: el objeto que se crea al ejecutar`CreateMetadataTransferJob`.

AWS IoT TwinMaker metadataTransferJob funcionalidad

En este tema se explica el comportamiento que se AWS IoT TwinMaker sigue cuando se ejecuta una operación masiva: cómo metadataTransferJob se procesa a. También se explica cómo definir un esquema con los metadatos necesarios para transferir los recursos. AWS IoT TwinMaker las operaciones masivas admiten las siguientes funciones:

- Crear o reemplazar un recurso de nivel superior: AWS IoT TwinMaker creará nuevos recursos o reemplazará todos los recursos existentes que se identifiquen de forma exclusiva mediante un identificador de recurso.

  Por ejemplo, si existe una entidad en el sistema, la definición de la entidad se sustituirá por la nueva definida en la plantilla situada debajo de la `Entity` clave.
- Crear o reemplazar un subrecurso:

  Desde el EntityComponent nivel, solo puede crear o reemplazar un componente. La entidad ya debe existir; de lo contrario, la acción generará un ValidationException.

  Desde el nivel de propiedad o relación, solo puede crear o reemplazar una propiedad o relación, y el elemento que lo contiene EntityComponent debe existir ya.
- Eliminar subrecurso:

AWS IoT TwinMaker también admite la eliminación de subrecursos. Un subrecurso puede ser un componente, una propiedad o una relación.

Si desea eliminar un componente, debe hacerlo desde el nivel de entidad.

Si desea eliminar una propiedad o relación, debe hacerlo desde la entidad o el EntityComponent nivel.

Para eliminar un subrecurso, debe actualizar el recurso de nivel superior y omitir la definición del subrecurso.

- Sin eliminación de recursos de nivel superior: nunca AWS IoT TwinMaker se eliminarán los recursos de nivel superior. Un recurso de nivel superior hace referencia a una entidad o. ComponentType
- No hay definiciones de subrecursos para el mismo recurso de nivel superior en una plantilla:

  No puedes proporcionar la definición completa de entidad y la definición de subrecurso (como propiedad) de la misma entidad en la misma plantilla.

  Si se usa un ID de entidad en la entidad, no se puede usar el mismo ID en la entidad EntityComponent, la propiedad o la relación.

  Si se usa una combinación de EntityID o ComponentName EntityComponent, no se puede usar la misma combinación en EntityComponent, propiedad o relación.

  Si se usa una combinación EntityID, ComponentName o PropertyName en una propiedad o relación, no podrá usar la misma combinación en la propiedad o relación.
- ExternalId es opcional para AWS IoT TwinMaker: ExternalId Se puede utilizar para ayudarle a identificar sus recursos.

Realizar operaciones de importación y exportación masivas

En este tema se explica cómo realizar operaciones de importación y exportación masivas y cómo gestionar los errores en los trabajos de transferencia. Proporciona ejemplos de trabajos de transferencia mediante comandos CLI.

La referencia AWS IoT TwinMaker de la API contiene información sobre esta CreateMetadataTransferJoby otras acciones de la API.

Temas

- metadataTransferJob requisitos previos
- Permisos de IAM
- Ejecute una operación masiva
- Gestión de errores
- Importa plantillas de metadatos
- AWS IoT TwinMaker metadataTransferJob ejemplos

metadataTransferJob requisitos previos

Complete los siguientes requisitos previos antes de ejecutar un: metadataTransferJob

- Cree un AWS IoT TwinMaker espacio de trabajo. El espacio de trabajo puede ser el destino de importación o la fuente de exportación de un metadataTransferJob. Para obtener información sobre la creación de un espacio de trabajo, consulte,Creación de un espacio de trabajo.
- Cree un bucket de Amazon S3 para almacenar recursos. Para obtener más información sobre el uso de Amazon S3, consulte ¿Qué es Amazon S3?

Permisos de IAM

Cuando realiza operaciones masivas, debe crear una política de IAM con permisos que permitan el intercambio de AWS recursos entre Amazon S3 y su máquina local. AWS IoT TwinMaker AWS IoT SiteWise Para obtener más información sobre la creación de políticas de IAM, consulte Creación de políticas de IAM.

Las declaraciones de política de AWS IoT TwinMaker Amazon S3 AWS IoT SiteWise y Amazon S3 se enumeran aquí:

- AWS IoT TwinMaker política:

  JSON

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

```
            "s3:PutObject",
            "s3:GetObject",
            "s3:GetBucketLocation",
            "s3:ListBucket",
            "s3:AbortMultipartUpload",
            "s3:ListBucketMultipartUploads",
            "s3:ListMultipartUploadParts"
        ],
        "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "iottwinmaker:GetWorkspace",
                "iottwinmaker:CreateEntity",
                "iottwinmaker:GetEntity",
                "iottwinmaker:UpdateEntity",
                "iottwinmaker:GetComponentType",
                "iottwinmaker:CreateComponentType",
                "iottwinmaker:UpdateComponentType",
                "iottwinmaker:ListEntities",
                "iottwinmaker:ListComponentTypes",
                "iottwinmaker:ListTagsForResource",
                "iottwinmaker:TagResource",
                "iottwinmaker:UntagResource"
            ],
            "Resource": "*"
        }
    ]
}
```

- AWS IoT SiteWise política:

  JSON

```
{
    "Version": "2012-10-17",
    "Statement": [{
        "Effect": "Allow",
        "Action": [
            "s3:PutObject",
            "s3:GetObject",
            "s3:GetBucketLocation",
            "s3:ListBucket",
```

```
            "s3:AbortMultipartUpload",
            "s3:ListBucketMultipartUploads",
            "s3:ListMultipartUploadParts"
        ],
        "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "iotsitewise:CreateAsset",
                "iotsitewise:CreateAssetModel",
                "iotsitewise:UpdateAsset",
                "iotsitewise:UpdateAssetModel",
                "iotsitewise:UpdateAssetProperty",
                "iotsitewise:ListAssets",
                "iotsitewise:ListAssetModels",
                "iotsitewise:ListAssetProperties",
                "iotsitewise:ListAssetModelProperties",
                "iotsitewise:ListAssociatedAssets",
                "iotsitewise:DescribeAsset",
                "iotsitewise:DescribeAssetModel",
                "iotsitewise:DescribeAssetProperty",
                "iotsitewise:AssociateAssets",
                "iotsitewise:DisassociateAssets",
                "iotsitewise:AssociateTimeSeriesToAssetProperty",
                "iotsitewise:DisassociateTimeSeriesFromAssetProperty",
                "iotsitewise:BatchPutAssetPropertyValue",
                "iotsitewise:BatchGetAssetPropertyValue",
                "iotsitewise:TagResource",
                "iotsitewise:UntagResource",
                "iotsitewise:ListTagsForResource"
            ],
            "Resource": "*"
        }
    ]
}
```

- Política de Amazon S3:

```
{
    "Effect": "Allow",
    "Action": [
        "s3:PutObject",
        "s3:GetObject",
```

```
            "s3:GetBucketLocation",
            "s3:ListBucket",
            "s3:AbortMultipartUpload",
            "s3:ListBucketMultipartUploads",
            "s3:ListMultipartUploadParts"
        ],
        "Resource": "*"
}
```

Como alternativa, puede ajustar su política de Amazon S3 para acceder únicamente a un único bucket de Amazon S3 (consulte la política siguiente).

Política de Amazon S3 con ámbito de depósito único

```
{
    "Effect": "Allow",
    "Action": [
        "s3:PutObject",
        "s3:GetObject",
        "s3:GetBucketLocation",
        "s3:ListBucket",
        "s3:AbortMultipartUpload",
        "s3:ListBucketMultipartUploads",
        "s3:ListMultipartUploadParts"
    ],
    "Resource": [
        "arn:aws:s3:::bucket name",
        "arn:aws:s3:::bucket name/*"
    ]
}
```

Configure el control de acceso para un metadataTransferJob

Para controlar el tipo de trabajos a los que puede acceder un usuario, añada la siguiente política de IAM al rol utilizado para llamar AWS IoT TwinMaker.

> **Note**
>
> Esta política solo permite el acceso a los trabajos de AWS IoT TwinMaker importación y exportación que transfieren recursos desde y hacia Amazon S3.

```
{
    "Effect": "Allow",
    "Action": [
        "iottwinmaker:*DataTransferJob*"
    ],
    "Resource": "*",
    "Condition": {
        "StringLikeIfExists": {
            "iottwinmaker:sourceType": [
                "s3",
                "iottwinmaker"
            ],
            "iottwinmaker:destinationType": [
                "iottwinmaker",
                "s3"
            ]
        }
    }
}
```

Ejecute una operación masiva

En esta sección se explica cómo realizar operaciones de importación y exportación masivas.

Importación de datos de Amazon S3 a AWS IoT TwinMaker

1. Especifique los recursos que desea transferir mediante el AWS IoT TwinMaker metadataTransferJob esquema. Cree y almacene su archivo de esquema en su bucket de Amazon S3.

   Para ver esquemas de ejemplo, consulteImporta plantillas de metadatos.

2. Cree un cuerpo de solicitud y guárdelo como un archivo JSON. El cuerpo de la solicitud especifica el origen y el destino del trabajo de transferencia. Asegúrese de especificar su bucket de Amazon S3 como origen y su AWS IoT TwinMaker espacio de trabajo como destino.

   El siguiente es un ejemplo del cuerpo de una solicitud:

   ```
   {
       "metadataTransferJobId": "your-transfer-job-Id",
       "sources": [{
           "type": "s3",
   ```

```
        "s3Configuration": {
            "location": "arn:aws:s3:::amzn-s3-demo-bucket/your_import_data.json"
        }
    }],
    "destination": {
        "type": "iottwinmaker",
        "iotTwinMakerConfiguration": {
            "workspace": "arn:aws:iottwinmaker:us-
east-1:111122223333:workspace/your-worksapce-name"
        }
    }
}
```

Registra el nombre del archivo que le diste al cuerpo de la solicitud; lo necesitarás en el siguiente paso. En este ejemplo, se nombra el cuerpo de la solicitud`createMetadataTransferJobImport.json`.

3. Ejecute el siguiente comando CLI para invocarlo `CreateMetadataTransferJob` (sustituya el nombre del archivo input-json por el nombre que le dio al cuerpo de la solicitud):

```
aws iottwinmaker create-metadata-transfer-job --region us-east-1 \
--cli-input-json file://createMetadataTransferJobImport.json
```

Esto crea un metadataTransferJob e inicia el proceso de transferencia de los recursos seleccionados.

Exportación de datos desde AWS IoT TwinMaker Amazon S3

1. Cree un cuerpo de solicitud en formato JSON con los filtros adecuados para elegir los recursos que desee exportar. Para este ejemplo utilizamos:

```
{
    "metadataTransferJobId": "your-transfer-job-Id",
    "sources": [{
        "type": "iottwinmaker",
        "iotTwinMakerConfiguration": {
            "workspace": "arn:aws:iottwinmaker:us-
east-1:111122223333:workspace/your-workspace-name",
            "filters": [{
                "filterByEntity": {
                    "entityId": "parent"
```

```
                }},
                {
                "filterByEntity": {
                    "entityId": "child"
                }},
                {
                "filterByComponentType": {
                    "componentTypeId": "component.type.minimal"
                }}
            ]
        }
    }],
    "destination": {
        "type": "s3",
        "s3Configuration": {
            "location": "arn:aws:s3:::amzn-s3-demo-bucket"
        }
    }
}
```

La `filters` matriz le permite especificar qué recursos se van a exportar. En este ejemplo, filtramos por`entity`, y`componentType`.

Asegúrese de especificar su AWS IoT TwinMaker espacio de trabajo como origen y su bucket de Amazon S3 como destino del trabajo de transferencia de metadatos.

Guarde el cuerpo de la solicitud y registre el nombre del archivo; lo necesitará en el siguiente paso. Para este ejemplo, asignamos un nombre al cuerpo de nuestra solicitud`createMetadataTransferJobExport.json`.

2. Ejecute el siguiente comando CLI para invocarlo `CreateMetadataTransferJob` (sustituya el nombre del archivo input-json por el nombre que le dio al cuerpo de la solicitud):

```
aws iottwinmaker create-metadata-transfer-job --region us-east-1 \
--cli-input-json file://createMetadataTransferJobExport.json
```

Esto crea un metadataTransferJob e inicia el proceso de transferencia de los recursos seleccionados.

Para comprobar o actualizar el estado de un trabajo de transferencia, utilice los siguientes comandos:

- Para cancelar un trabajo, usa la acción CancelMetadataTransferJobde la API. Cuando llamas CancelMetadataTransferJob, la API solo cancela una ejecución metadataTransferJob y los recursos ya exportados o importados no se ven afectados por esta llamada a la API.
- Para recuperar información sobre un trabajo específico, usa la acción de la GetMetadataTransferJobAPI.

  O bien, puede invocar GetMetadataTransferJob un trabajo de transferencia existente con el siguiente comando CLI:

  ```
  aws iottwinmaker get-metadata-transfer-job --job-id ExistingJobId
  ```

  Si solicitas GetMetadataTransferJob un trabajo de AWS IoT TwinMaker importación o exportación que no existe, recibirás un `ResourceNotFoundException` error en la respuesta.
- Para enumerar los trabajos actuales, usa la acción de la ListMetadataTransferJobsAPI.

  A continuación, se muestra un ejemplo de CLI que llama ListMetadataTransferJobs AWS IoT TwinMaker como DestinationType y `s3` como SourceType:

  ```
  aws iottwinmaker list-metadata-transfer-jobs --destination-type iottwinmaker --
  source-type s3
  ```

  > **Note**
  >
  > Puede cambiar los valores de los parámetros SourceType y DestinationType para que coincidan con el origen y el destino del trabajo de importación o exportación.

Para ver más ejemplos de comandos CLI que invocan estas acciones de API, consulteAWS IoT TwinMaker metadataTransferJob ejemplos.

Si encuentra algún error durante el trabajo de transferencia, consulteGestión de errores.

Gestión de errores

Después de crear y ejecutar un trabajo de transferencia, puedes llamar GetMetadataTransferJob para diagnosticar cualquier error que se haya producido:

```
aws iottwinmaker get-metadata-transfer-job \
```

```
--metadata-transfer-job-id your_metadata_transfer_job_id \
--region us-east-1
```

Una vez que vea el estado del trabajo al que se dirigeCOMPLETED, podrá comprobar los resultados del trabajo. GetMetadataTransferJob devuelve un objeto denominado MetadataTransferJobProgressque contiene los siguientes campos:

- FailedCount: indica el número de recursos que fallaron durante el proceso de transferencia.
- Número omitido: indica el número de recursos que se omitieron durante el proceso de transferencia.
- SuceededCount: indica la cantidad de recursos que se realizaron correctamente durante el proceso de transferencia.
- TotalCount: indica el recuento total de los recursos involucrados en el proceso de transferencia.

Además, se devuelve un elemento ReportURL que contiene una URL prefirmada. Si tu trabajo de transferencia contiene errores que deseas investigar más a fondo, puedes descargar un informe de errores completo utilizando esta URL.

Importa plantillas de metadatos

Puede importar muchos componentes, tipos de componentes o entidades con una sola operación de importación masiva. Los ejemplos de esta sección muestran cómo hacerlo.

template: Importing entities

Utilice el siguiente formato de plantilla para un trabajo que importe entidades:

```
{
  "entities": [
    {
      "description": "string",
      "entityId": "string",
      "entityName": "string",
      "parentEntityId": "string",
      "tags": {
        "string": "string"
      },
      "components": {
        "string": {
          "componentTypeId": "string",
```

```
          "description": "string",
          "properties": {
            "string": {
              "definition": {
                "configuration": {
                  "string": "string"
                },
                "dataType": "DataType",
                "defaultValue": "DataValue",
                "displayName": "string",
                "isExternalId": "boolean",
                "isRequiredInEntity": "boolean",
                "isStoredExternally": "boolean",
                "isTimeSeries": "boolean"
              },
              "value": "DataValue"
            }
          },
          "propertyGroups": {
            "string": {
              "groupType": "string",
              "propertyNames": [
                "string"
              ]
            }
          }
        }
      }
    ]
}
```

template: Importing componentTypes

Utilice el siguiente formato de plantilla para un trabajo que importe tipos de componentes:

```
{
  "componentTypes": [
    {
      "componentTypeId": "string",
      "componentTypeName": "string",
      "description": "string",
      "extendsFrom": [
        "string"
```

```
      ],
      "functions": {
        "string": {
          "implementedBy": {
            "isNative": "boolean",
            "lambda": {
              "functionName": "Telemetry-tsDataReader",
              "arn": "Telemetry-tsDataReaderARN"
            }
          },
          "requiredProperties": [
            "string"
          ],
          "scope": "string"
        }
      },
      "isSingleton": "boolean",
      "propertyDefinitions": {
        "string": {
          "configuration": {
            "string": "string"
          },
          "dataType": "DataType",
          "defaultValue": "DataValue",
          "displayName": "string",
          "isExternalId": "boolean",
          "isRequiredInEntity": "boolean",
          "isStoredExternally": "boolean",
          "isTimeSeries": "boolean"
        }
      },
      "propertyGroups": {
        "string": {
          "groupType": "string",
          "propertyNames": [
            "string"
          ]
        }
      },
      "tags": {
        "string": "string"
      }
    }
  ]
```

```
}
```

template: Importing components

Utilice el siguiente formato de plantilla para un trabajo que importe componentes:

```
{
  "entityComponents": [
    {
      "entityId": "string",
      "componentName": "string",
      "componentTypeId": "string",
      "description": "string",
      "properties": {
        "string": {
          "definition": {
            "configuration": {
              "string": "string"
            },
            "dataType": "DataType",
            "defaultValue": "DataValue",
            "displayName": "string",
            "isExternalId": "boolean",
            "isRequiredInEntity": "boolean",
            "isStoredExternally": "boolean",
            "isTimeSeries": "boolean"
          },
          "value": "DataValue"
        }
      },
      "propertyGroups": {
        "string": {
          "groupType": "string",
          "propertyNames": [
            "string"
          ]
        }
      }
    }
  ]
}
```

AWS IoT TwinMaker metadataTransferJob ejemplos

Utilice los siguientes comandos para gestionar las transferencias de metadatos:

- CreateMetadataTransferJobAcción de la API.

  Ejemplo de comando CLI:

  ```
  aws iottwinmaker create-metadata-transfer-job --region us-east-1 \
  --cli-input-json file://yourTransferFileName.json
  ```

- Para cancelar un trabajo, utilice la acción CancelMetadataTransferJobde la API.

  Ejemplo de comando CLI:

  ```
  aws iottwinmaker cancel-metadata-transfer-job
  --region us-east-1 \
  --metadata-transfer-job-id job-to-cancel-id
  ```

  Cuando llamas CancelMetadataTransferJob, solo se cancela un trabajo de transferencia de metadatos específico y los recursos ya exportados o importados no se ven afectados.

- Para recuperar información sobre un trabajo específico, usa la acción de la GetMetadataTransferJobAPI.

  Ejemplo de comando CLI:

  ```
  aws iottwinmaker get-metadata-transfer-job \
  --metadata-transfer-job-id your_metadata_transfer_job_id \
  --region us-east-1 \
  ```

- Para enumerar los trabajos actuales, utilice la acción ListMetadataTransferJobsde la API.

  Puedes filtrar los resultados devueltos ListMetadataTransferJobs mediante un archivo JSON. Consulte el siguiente procedimiento mediante la CLI:

  1. Cree un archivo JSON de entrada CLI para especificar los filtros que quiere usar:

     ```
     {
         "sourceType": "s3",
         "destinationType": "iottwinmaker",
         "filters": [{
     ```

```
        "workspaceId": "workspaceforbulkimport"
    },
    {
        "state": "COMPLETED"
    }]
}
```

Guárdelo y registre el nombre del archivo, lo necesitará al ingresar el comando CLI.

2. Utilice el archivo JSON como argumento para el siguiente comando CLI:

```
aws iottwinmaker list-metadata-transfer-job --region us-east-1 \
--cli-input-json file://ListMetadataTransferJobsExample.json
```

AWS IoT TwinMaker esquema de trabajo de transferencia de metadatos

metadataTransferJob esquema de importación: utilice este esquema de AWS IoT TwinMaker metadatos para validar sus datos cuando los cargue en un bucket de Amazon S3:

```
{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "title": "IoTTwinMaker",
  "description": "Metadata transfer job resource schema for IoTTwinMaker",
  "definitions": {
    "ExternalId": {
      "type": "string",
      "minLength": 1,
      "maxLength": 128,
      "pattern": "[a-zA-Z0-9][a-zA-Z_\\-0-9.:]*[a-zA-Z0-9]+"
    },
    "Description": {
      "type": "string",
      "minLength": 0,
      "maxLength": 512
    },
    "DescriptionWithDefault": {
      "type": "string",
      "minLength": 0,
      "maxLength": 512,
      "default": ""
```

```
    },
    "ComponentTypeName": {
      "description": "A friendly name for the component type.",
      "type": "string",
      "pattern": ".*[^\\u0000-\\u001F\\u007F]*.*",
      "minLength": 1,
      "maxLength": 256
    },
    "ComponentTypeId": {
      "description": "The ID of the component type.",
      "type": "string",
      "pattern": "[a-zA-Z_.\\-0-9:]+",
      "minLength": 1,
      "maxLength": 256
    },
    "ComponentName": {
      "description": "The name of the component.",
      "type": "string",
      "pattern": "[a-zA-Z_\\-0-9]+",
      "minLength": 1,
      "maxLength": 256
    },
    "EntityId": {
      "description": "The ID of the entity.",
      "type": "string",
      "minLength": 1,
      "maxLength": 128,
      "pattern": "[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}|^[a-zA-
Z0-9][a-zA-Z_\\-0-9.:]*[a-zA-Z0-9]+"
    },
    "EntityName": {
      "description": "The name of the entity.",
      "type": "string",
      "minLength": 1,
      "maxLength": 256,
      "pattern": "[a-zA-Z_0-9-.][a-zA-Z_0-9-. ]*[a-zA-Z0-9]+"
    },
    "ParentEntityId": {
      "description": "The ID of the parent entity.",
      "type": "string",
      "minLength": 1,
      "maxLength": 128,
      "pattern": "\\$ROOT|^[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]
{12}|^[a-zA-Z0-9][a-zA-Z_\\-0-9.:]*[a-zA-Z0-9]+",
```

```
      "default": "$ROOT"
    },
    "DisplayName": {
      "description": "A friendly name for the property.",
      "type": "string",
      "pattern": ".*[^\\u0000-\\u001F\\u007F]*.*",
      "minLength": 0,
      "maxLength": 256
    },
    "Tags": {
      "description": "Metadata that you can use to manage the entity / componentType",
      "patternProperties": {
        "^([\\p{L}\\p{Z}\\p{N}_.:/=+\\-@]*)$": {
          "type": "string",
          "minLength": 1,
          "maxLength": 256
        }
      },
      "existingJavaType": "java.util.Map<String,String>",
      "minProperties": 0,
      "maxProperties": 50
    },
    "Relationship": {
      "description": "The type of the relationship.",
      "type": "object",
      "properties": {
        "relationshipType": {
          "description": "The type of the relationship.",
          "type": "string",
          "pattern": ".*",
          "minLength": 1,
          "maxLength": 256
        },
        "targetComponentTypeId": {
          "description": "The ID of the target component type associated with this
relationship.",
          "$ref": "#/definitions/ComponentTypeId"
        }
      },
      "additionalProperties": false
    },
    "DataValue": {
      "description": "An object that specifies a value for a property.",
      "type": "object",
```

```
      "properties": {
        "booleanValue": {
          "description": "A Boolean value.",
          "type": "boolean"
        },
        "doubleValue": {
          "description": "A double value.",
          "type": "number"
        },
        "expression": {
          "description": "An expression that produces the value.",
          "type": "string",
          "pattern": "(^\\$\\{Parameters\\.[a-zA-z]+([a-zA-z_0-9]*)}$)",
          "minLength": 1,
          "maxLength": 316
        },
        "integerValue": {
          "description": "An integer value.",
          "type": "integer"
        },
        "listValue": {
          "description": "A list of multiple values.",
          "type": "array",
          "minItems": 0,
          "maxItems": 50,
          "uniqueItems": false,
          "insertionOrder": false,
          "items": {
            "$ref": "#/definitions/DataValue"
          },
          "default": null
        },
        "longValue": {
          "description": "A long value.",
          "type": "integer",
          "existingJavaType": "java.lang.Long"
        },
        "stringValue": {
          "description": "A string value.",
          "type": "string",
          "pattern": ".*",
          "minLength": 1,
          "maxLength": 256
        },
```

```
        "mapValue": {
          "description": "An object that maps strings to multiple DataValue objects.",
          "type": "object",
          "patternProperties": {
            "[a-zA-Z_\\-0-9]+": {
              "$ref": "#/definitions/DataValue"
            }
          },
          "additionalProperties": {
            "$ref": "#/definitions/DataValue"
          }
        },
        "relationshipValue": {
          "description": "A value that relates a component to another component.",
          "type": "object",
          "properties": {
            "TargetComponentName": {
              "type": "string",
              "pattern": "[a-zA-Z_\\-0-9]+",
              "minLength": 1,
              "maxLength": 256
            },
            "TargetEntityId": {
              "type": "string",
              "pattern": "[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}|
^[a-zA-Z0-9][a-zA-Z_\\-0-9.:]*[a-zA-Z0-9]+",
              "minLength": 1,
              "maxLength": 128
            }
          },
          "additionalProperties": false
        }
      },
      "additionalProperties": false
    },
    "DataType": {
      "description": "An object that specifies the data type of a property.",
      "type": "object",
      "properties": {
        "allowedValues": {
          "description": "The allowed values for this data type.",
          "type": "array",
          "minItems": 0,
          "maxItems": 50,
```

```
          "uniqueItems": false,
          "insertionOrder": false,
          "items": {
            "$ref": "#/definitions/DataValue"
          },
          "default": null
        },
        "nestedType": {
          "description": "The nested type in the data type.",
          "$ref": "#/definitions/DataType"
        },
        "relationship": {
          "description": "A relationship that associates a component with another
component.",
          "$ref": "#/definitions/Relationship"
        },
        "type": {
          "description": "The underlying type of the data type.",
          "type": "string",
          "enum": [
            "RELATIONSHIP",
            "STRING",
            "LONG",
            "BOOLEAN",
            "INTEGER",
            "DOUBLE",
            "LIST",
            "MAP"
          ]
        },
        "unitOfMeasure": {
          "description": "The unit of measure used in this data type.",
          "type": "string",
          "pattern": ".*",
          "minLength": 1,
          "maxLength": 256
        }
      },
      "required": [
        "type"
      ],
      "additionalProperties": false
    },
    "PropertyDefinition": {
```

```
      "description": "An object that specifies information about a property.",
      "type": "object",
      "properties": {
        "configuration": {
          "description": "An object that specifies information about a property.",
          "patternProperties": {
            "[a-zA-Z_\\-0-9]+": {
              "type": "string",
              "pattern": "[a-zA-Z_\\-0-9]+",
              "minLength": 1,
              "maxLength": 256
            }
          },
          "existingJavaType": "java.util.Map<String,String>"
        },
        "dataType": {
          "description": "An object that contains information about the data type.",
          "$ref": "#/definitions/DataType"
        },
        "defaultValue": {
          "description": "An object that contains the default value.",
          "$ref": "#/definitions/DataValue"
        },
        "displayName": {
          "description": "An object that contains the default value.",
          "$ref": "#/definitions/DisplayName"
        },
        "isExternalId": {
          "description": "A Boolean value that specifies whether the property ID comes
from an external data store.",
          "type": "boolean",
          "default": null
        },
        "isRequiredInEntity": {
          "description": "A Boolean value that specifies whether the property is
required.",
          "type": "boolean",
          "default": null
        },
        "isStoredExternally": {
          "description": "A Boolean value that specifies whether the property is stored
externally.",
          "type": "boolean",
          "default": null
```

```
        },
        "isTimeSeries": {
          "description": "A Boolean value that specifies whether the property consists
of time series data.",
          "type": "boolean",
          "default": null
        }
      },
      "additionalProperties": false
    },
    "PropertyDefinitions": {
      "type": "object",
      "patternProperties": {
        "[a-zA-Z_\\-0-9]+": {
          "$ref": "#/definitions/PropertyDefinition"
        }
      },
      "additionalProperties": {
        "$ref": "#/definitions/PropertyDefinition"
      }
    },
    "Property": {
      "type": "object",
      "properties": {
        "definition": {
          "description": "The definition of the property",
          "$ref": "#/definitions/PropertyDefinition"
        },
        "value": {
          "description": "The value of the property.",
          "$ref": "#/definitions/DataValue"
        }
      },
      "additionalProperties": false
    },
    "Properties": {
      "type": "object",
      "patternProperties": {
        "[a-zA-Z_\\-0-9]+": {
          "$ref": "#/definitions/Property"
        }
      },
      "additionalProperties": {
        "$ref": "#/definitions/Property"
```

```
      }
    },
    "PropertyName": {
      "type": "string",
      "pattern": "[a-zA-Z_\\-0-9]+"
    },
    "PropertyGroup": {
      "description": "An object that specifies information about a property group.",
      "type": "object",
      "properties": {
        "groupType": {
          "description": "The type of property group.",
          "type": "string",
          "enum": [
            "TABULAR"
          ]
        },
        "propertyNames": {
          "description": "The list of property names in the property group.",
          "type": "array",
          "minItems": 1,
          "maxItems": 256,
          "uniqueItems": true,
          "insertionOrder": false,
          "items": {
            "$ref": "#/definitions/PropertyName"
          },
          "default": null
        }
      },
      "additionalProperties": false
    },
    "PropertyGroups": {
      "type": "object",
      "patternProperties": {
        "[a-zA-Z_\\-0-9]+": {
          "$ref": "#/definitions/PropertyGroup"
        }
      },
      "additionalProperties": {
        "$ref": "#/definitions/PropertyGroup"
      }
    },
    "Component": {
```

```
      "type": "object",
      "properties": {
        "componentTypeId": {
          "$ref": "#/definitions/ComponentTypeId"
        },
        "description": {
          "$ref": "#/definitions/Description"
        },
        "properties": {
          "description": "An object that maps strings to the properties to set in the
 component type. Each string in the mapping must be unique to this object.",
          "$ref": "#/definitions/Properties"
        },
        "propertyGroups": {
          "description": "An object that maps strings to the property groups to set in
 the entity component. Each string in the mapping must be unique to this object.",
          "$ref": "#/definitions/PropertyGroups"
        }
      },
      "required": [
        "componentTypeId"
      ],
      "additionalProperties": false
    },
    "RequiredProperty": {
      "type": "string",
      "pattern": "[a-zA-Z_\\-0-9]+"
    },
    "LambdaFunction": {
      "type": "object",
      "properties": {
        "arn": {
          "type": "string",
          "pattern": "arn:((aws)|(aws-cn)|(aws-us-gov)|(\\${partition})):lambda:(([a-
z0-9-]+)|(\\${region})):([0-9]{12}|(\\${accountId})):function:[/a-zA-Z0-9_-]+",
          "minLength": 1,
          "maxLength": 128
        }
      },
      "additionalProperties": false,
      "required": [
        "arn"
      ]
    },
```

```
    "DataConnector": {
      "description": "The data connector.",
      "type": "object",
      "properties": {
        "isNative": {
          "description": "A Boolean value that specifies whether the data connector is
native to IoT TwinMaker.",
          "type": "boolean"
        },
        "lambda": {
          "description": "The Lambda function associated with this data connector.",
          "$ref": "#/definitions/LambdaFunction"
        }
      },
      "additionalProperties": false
    },
    "Function": {
      "description": "The function of component type.",
      "type": "object",
      "properties": {
        "implementedBy": {
          "description": "The data connector.",
          "$ref": "#/definitions/DataConnector"
        },
        "requiredProperties": {
          "description": "The required properties of the function.",
          "type": "array",
          "minItems": 1,
          "maxItems": 256,
          "uniqueItems": true,
          "insertionOrder": false,
          "items": {
            "$ref": "#/definitions/RequiredProperty"
          },
          "default": null
        },
        "scope": {
          "description": "The scope of the function.",
          "type": "string",
          "enum": [
            "ENTITY",
            "WORKSPACE"
          ]
        }
```

```
      },
      "additionalProperties": false
    },
    "Entity": {
      "type": "object",
      "properties": {
        "description": {
          "description": "The description of the entity.",
          "$ref": "#/definitions/DescriptionWithDefault"
        },
        "entityId": {
          "$ref": "#/definitions/EntityId"
        },
        "entityExternalId": {
          "description": "The external ID of the entity.",
          "$ref": "#/definitions/ExternalId"
        },
        "entityName": {
          "$ref": "#/definitions/EntityName"
        },
        "parentEntityId": {
          "$ref": "#/definitions/ParentEntityId"
        },
        "tags": {
          "$ref": "#/definitions/Tags"
        },
        "components": {
          "description": "A map that sets information about a component.",
          "type": "object",
          "patternProperties": {
            "[a-zA-Z_\\-0-9]+": {
              "$ref": "#/definitions/Component"
            }
          },
          "additionalProperties": {
            "$ref": "#/definitions/Component"
          }
        }
      },
      "required": [
        "entityId",
        "entityName"
      ],
      "additionalProperties": false
```

```
    },
    "ComponentType": {
      "type": "object",
      "properties": {
        "description": {
          "description": "The description of the component type.",
          "$ref": "#/definitions/DescriptionWithDefault"
        },
        "componentTypeId": {
          "$ref": "#/definitions/ComponentTypeId"
        },
        "componentTypeExternalId": {
          "description": "The external ID of the component type.",
          "$ref": "#/definitions/ExternalId"
        },
        "componentTypeName": {
          "$ref": "#/definitions/ComponentTypeName"
        },
        "extendsFrom": {
          "description": "Specifies the parent component type to extend.",
          "type": "array",
          "minItems": 1,
          "maxItems": 256,
          "uniqueItems": true,
          "insertionOrder": false,
          "items": {
            "$ref": "#/definitions/ComponentTypeId"
          },
          "default": null
        },
        "functions": {
          "description": "a Map of functions in the component type. Each function's key
must be unique to this map.",
          "type": "object",
          "patternProperties": {
            "[a-zA-Z_\\-0-9]+": {
              "$ref": "#/definitions/Function"
            }
          },
          "additionalProperties": {
            "$ref": "#/definitions/Function"
          }
        },
        "isSingleton": {
```

```
          "description": "A Boolean value that specifies whether an entity can have
more than one component of this type.",
          "type": "boolean",
          "default": false
        },
        "propertyDefinitions": {
          "description": "An map of the property definitions in the component type.
Each property definition's key must be unique to this map.",
          "$ref": "#/definitions/PropertyDefinitions"
        },
        "propertyGroups": {
          "description": "An object that maps strings to the property groups to set in
the component type. Each string in the mapping must be unique to this object.",
          "$ref": "#/definitions/PropertyGroups"
        },
        "tags": {
          "$ref": "#/definitions/Tags"
        }
      },
      "required": [
        "componentTypeId"
      ],
      "additionalProperties": false
    },
    "EntityComponent": {
      "type": "object",
      "properties": {
        "entityId": {
          "$ref": "#/definitions/EntityId"
        },
        "componentName": {
          "$ref": "#/definitions/ComponentName"
        },
        "componentExternalId": {
          "description": "The external ID of the component.",
          "$ref": "#/definitions/ExternalId"
        },
        "componentTypeId": {
          "$ref": "#/definitions/ComponentTypeId"
        },
        "description": {
          "description": "The description of the component.",
          "$ref": "#/definitions/Description"
        },
```

```
        "properties": {
          "description": "An object that maps strings to the properties to set in the
component. Each string in the mapping must be unique to this object.",
          "$ref": "#/definitions/Properties"
        },
        "propertyGroups": {
          "description": "An object that maps strings to the property groups to set in
the component. Each string in the mapping must be unique to this object.",
          "$ref": "#/definitions/PropertyGroups"
        }
      },
      "required": [
        "entityId",
        "componentTypeId",
        "componentName"
      ],
      "additionalProperties": false
    }
  },
  "additionalProperties": false,
  "properties": {
    "entities": {
      "type": "array",
      "uniqueItems": false,
      "items": {
        "$ref": "#/definitions/Entity"
      }
    },
    "componentTypes": {
      "type": "array",
      "uniqueItems": false,
      "items": {
        "$ref": "#/definitions/ComponentType"
      }
    },
    "entityComponents": {
      "type": "array",
      "uniqueItems": false,
      "items": {
        "$ref": "#/definitions/EntityComponent"
      },
      "default": null
    }
  }
```

```
}
```

A continuación, se muestra un ejemplo en el que se crea un nuevo ComponentType llamado `component.type.intial` y se crea una entidad llamada`initial`:

```
{
  "componentTypes": [
    {
      "componentTypeId": "component.type.initial",
      "tags": {
        "key": "value"
      }
    }
  ],
  "entities": [
    {
      "entityName": "initial",
      "entityId": "initial"
    }
  ]
}
```

Este es un ejemplo que actualiza las entidades existentes:

```
{
  "componentTypes": [
    {
      "componentTypeId": "component.type.initial",
      "description": "updated"
    }
  ],
  "entities": [
    {
      "entityName": "parent",
      "entityId": "parent"
    },
    {
      "entityName": "child",
      "entityId": "child",
      "components": {
        "testComponent": {
          "componentTypeId": "component.type.initial",
          "properties": {
```

```
            "testProperty": {
              "definition": {
                "configuration": {
                  "alias": "property"
                },
                "dataType": {
                  "relationship": {
                    "relationshipType": "parent",
                    "targetComponentTypeId": "test"
                  },
                  "type": "STRING",
                  "unitOfMeasure": "t"
                },
                "displayName": "displayName"
              }
            }
          }
        },
        "parentEntityId": "parent"
      }
    ],
    "entityComponents": [
      {
        "entityId": "initial",
        "componentTypeId": "component.type.initial",
        "componentName": "entityComponent",
        "description": "additionalDescription",
        "properties": {
          "additionalProperty": {
            "definition": {
              "configuration": {
                "alias": "additionalProperty"
              },
              "dataType": {
                "type": "STRING"
              },
              "displayName": "additionalDisplayName"
            },
            "value": {
              "stringValue": "test"
            }
          }
        }
```

```
    }
  ]
}
```

AWS IoT TwinMaker conectores de datos

AWS IoT TwinMaker utiliza una arquitectura basada en conectores para que pueda conectar los datos de su propio almacén de datos a. AWS IoT TwinMaker Esto significa que no necesita migrar los datos antes de usarlos. AWS IoT TwinMaker Actualmente, AWS IoT TwinMaker es compatible con conectores de primera calidad para AWS IoT SiteWise. Si almacena datos de modelado y propiedades AWS IoT SiteWise, no necesitará implementar sus propios conectores. Si almacena los datos de modelado o propiedades en otros almacenes de datos, como Timestream, DynamoDB o Snowflake, debe implementar AWS Lambda conectores con la interfaz de conector de AWS IoT TwinMaker datos para poder invocar el conector cuando sea necesario. AWS IoT TwinMaker

Temas

- AWS IoT TwinMaker conectores de datos
- AWS IoT TwinMaker Conector de datos tabulares Athena
- Desarrollo de conectores de datos de series temporales AWS IoT TwinMaker

AWS IoT TwinMaker conectores de datos

Los conectores necesitan acceder al banco de datos subyacente para resolver las consultas enviadas y devolver resultados o un error.

Para obtener información sobre los conectores disponibles, sus interfaces de solicitud y sus interfaces de respuesta, consulte los siguientes temas.

Para obtener más información sobre las propiedades utilizadas en las interfaces de los conectores, consulte la acción de la GetPropertyValueHistoryAPI.

> **Note**
>
> Algunos conectores tienen dos campos de marca de tiempo en las interfaces de solicitud y respuesta para las propiedades de hora de inicio y finalización. Ambos `startDateTime` y `endDateTime` utilizan un número largo para representar el segundo de tiempo Unix, lo que ya no se admite. Para mantener la compatibilidad con versiones anteriores, seguimos enviando un valor de marca de tiempo a ese campo, pero recomendamos utilizar los campos `startTime` y `endTime` que sean coherentes con el formato de marca de tiempo de nuestra API.

Temas

- Conector de inicialización de esquemas
- DataReaderByEntity
- DataReaderByComponentType
- DataReader
- AttributePropertyValueReaderByEntity
- DataWriter
- Ejemplos

Conector de inicialización de esquemas

Puede utilizar el inicializador de esquemas en el tipo de componente o en el ciclo de vida de la entidad para obtener el tipo de componente o las propiedades del componente del origen de datos subyacente. El inicializador de esquemas importa automáticamente el tipo o las propiedades del componente sin solicitar explícitamente una acción de API para configurar `properties`.

SchemaInitializer interfaz de solicitud

```
{
  "workspaceId": "string",
  "entityId": "string",
  "componentName": "string",
  "properties": {
    // property name as key,
    // value is of type PropertyRequest
    "string": "PropertyRequest"
  }
}
```

> **Note**
>
> El mapa de propiedades de esta interfaz de solicitud es un `PropertyRequest`. Para obtener más información, consulte PropertyRequest.

SchemaInitializer interfaz de respuesta

```
{
  "properties": {
    // property name as key,
    // value is of type PropertyResponse
    "string": "PropertyResponse"
  }
}
```

> **Note**
>
> El mapa de propiedades de esta interfaz de solicitud es un `PropertyResponse`. Para obtener más información, consulte PropertyResponse.

DataReaderByEntity

DataReaderByEntity es un conector de plano de datos que se utiliza para obtener los valores de series temporales de las propiedades de un único componente.

Para obtener información sobre los tipos de propiedades, la sintaxis y el formato de este conector, consulte la acción de la GetPropertyValueHistoryAPI.

DataReaderByEntityinterfaz de solicitud

```
{
  "startDateTime": long, // In epoch sec, deprecated
  "startTime": "string", // ISO-8601 timestamp format
  "endDateTime": long, // In epoch sec, deprecated
  "endTime": "string", // ISO-8601 timestamp format
  "properties": {
    // A map of properties as in the get-entity API response
    // property name as key,
    // value is of type PropertyResponse
    "string": "PropertyResponse"
   },
  "workspaceId": "string",
  "selectedProperties": List:"string",
  "propertyFilters": List:PropertyFilter,
```

```
  "entityId": "string",
  "componentName": "string",
  "componentTypeId": "string",
  "interpolation": InterpolationParameters,
  "nextToken": "string",
  "maxResults": int,
  "orderByTime": "string"
  }
```

DataReaderByEntityinterfaz de respuesta

```
{
  "propertyValues": [
    {
      "entityPropertyReference": EntityPropertyReference, // The same
 as EntityPropertyReference
      "values": [
        {
        "timestamp": long, // Epoch sec, deprecated
        "time": "string", // ISO-8601 timestamp format
        "value": DataValue // The same as DataValue
        }
      ]
    }
  ],
  "nextToken": "string"
}
```

DataReaderByComponentType

Para obtener los valores de series temporales de las propiedades comunes que provienen del mismo tipo de componente, utilice el conector DataReaderByEntity del plano de datos. Por ejemplo, si define propiedades de series temporales en el tipo de componente y tiene varios componentes que utilizan ese tipo de componente, puede consultar esas propiedades en todos los componentes de un intervalo de tiempo determinado. Un caso de uso habitual es cuando se quiere consultar el estado de alarma de varios componentes para obtener una vista global de las entidades.

Para obtener información sobre los tipos de propiedades, la sintaxis y el formato de este conector, consulte la acción de la GetPropertyValueHistoryAPI.

DataReaderByComponentType interfaz de solicitud

```
{
  "startDateTime": long, // In epoch sec, deprecated
  "startTime": "string", // ISO-8601 timestamp format
  "endDateTime": long, // In epoch sec, deprecated
  "endTime": "string", // ISO-8601 timestamp format
  "properties": { // A map of properties as in the get-entity API response
    // property name as key,
    // value is of type PropertyResponse
    "string": "PropertyResponse"
   },
  "workspaceId": "string",
  "selectedProperties": List:"string",
  "propertyFilters": List:PropertyFilter,
  "componentTypeId": "string",
  "interpolation": InterpolationParameters,
  "nextToken": "string",
  "maxResults": int,
  "orderByTime": "string"
}
```

DataReaderByComponentType interfaz de respuesta

```
{
  "propertyValues": [
    {
      "entityPropertyReference": EntityPropertyReference, // The same
 as EntityPropertyReference
      "entityId": "string",
      "componentName": "string",
      "values": [
        {
        "timestamp": long, // Epoch sec, deprecated
        "time": "string", // ISO-8601 timestamp format
        "value": DataValue // The same as DataValue
        }
      ]
    }
  ],
  "nextToken": "string"
}
```

DataReader

DataReader es un conector de plano de datos que puede gestionar tanto el caso DataReaderByEntity de DataReaderByComponentType como

Para obtener información sobre los tipos de propiedades, la sintaxis y el formato de este conector, consulte la acción de la GetPropertyValueHistoryAPI.

DataReader interfaz de solicitud

Los parámetros `EntityId` y `componentName` son opcionales.

```
{
  "startDateTime": long, // In epoch sec, deprecated
  "startTime": "string", // ISO-8601 timestamp format
  "endDateTime": long, // In epoch sec, deprecated
  "endTime": "string", // ISO-8601 timestamp format
  "properties": { // A map of properties as in the get-entity API response
    // property name as key,
    // value is of type PropertyRequest
    "string": "PropertyRequest"
  },

  "workspaceId": "string",
  "selectedProperties": List:"string",
  "propertyFilters": List:PropertyFilter,
  "entityId": "string",
  "componentName": "string",
  "componentTypeId": "string",
  "interpolation": InterpolationParameters,
  "nextToken": "string",
  "maxResults": int,
  "orderByTime": "string"
}
```

DataReader interfaz de respuesta

```
{
  "propertyValues": [
    {
      "entityPropertyReference": EntityPropertyReference, // The same
 as EntityPropertyReference
```

```
      "values": [
        {
        "timestamp": long, // Epoch sec, deprecated
        "time": "string", // ISO-8601 timestamp format
        "value": DataValue // The same as DataValue
        }
      ]
    }
  ],
  "nextToken": "string"
}
```

AttributePropertyValueReaderByEntity

AttributePropertyValueReaderByEntity es un conector de plano de datos que se puede utilizar para obtener el valor de las propiedades estáticas de una sola entidad.

Para obtener información sobre los tipos de propiedades, la sintaxis y el formato de este conector, consulte la acción de la GetPropertyValueAPI.

AttributePropertyValueReaderByEntity interfaz de solicitud

```
{
  "properties": {
    // property name as key,
    // value is of type PropertyResponse
    "string": "PropertyResponse"
  }

  "workspaceId": "string",
  "entityId": "string",
  "componentName": "string",
  "selectedProperties": List:"string",
}
```

AttributePropertyValueReaderByEntity interfaz de respuesta

```
{
  "propertyValues": {
    "string": { // property name as key
        "propertyReference": EntityPropertyReference, // The same
 as EntityPropertyReference
```

```
        "propertyValue": DataValue // The same as DataValue
    }
}
```

DataWriter

DataWriter es un conector de plano de datos que se puede utilizar para volver a escribir puntos de datos de series temporales en el banco de datos subyacente para las propiedades de un solo componente.

Para obtener información sobre los tipos de propiedades, la sintaxis y el formato de este conector, consulte la acción de la BatchPutPropertyValuesAPI.

DataWriter interfaz de solicitud

```
{
  "workspaceId": "string",
  "properties": {
    // entity id as key
    "String": {
      // property name as key,
      // value is of type PropertyResponse
      "string": PropertyResponse
    }
  },
  "entries": [
    {
      "entryId": "string",
      "entityPropertyReference": EntityPropertyReference, // The same
 as EntityPropertyReference
      "propertyValues": [
        {
        "timestamp": long, // Epoch sec, deprecated
        "time": "string", // ISO-8601 timestamp format
        "value": DataValue // The same as DataValue
        }
      ]
    }
  ]
}
```

DataWriter interfaz de respuesta

```
{
  "errorEntries": [
    {
      "errors": List:BatchPutPropertyError // The value is a list of
 type BatchPutPropertyError
    }
  ]
}
```

Ejemplos

Los siguientes ejemplos de JSON son ejemplos de la sintaxis de respuesta y solicitud para varios conectores.

- SchemaInitializer:

  Los siguientes ejemplos muestran el inicializador de esquemas en el ciclo de vida de un tipo de componente.

  Solicitud:

```
{
  "workspaceId": "myWorkspace",
  "properties": {
    "modelId": {
      "definition": {
          "dataType": { "type": "STRING" },
          "isExternalId": true,
          "isFinal": true,
          "isImported": false,
          "isInherited": false,
          "isRequiredInEntity": true,
          "isStoredExternally": false,
          "isTimeSeries": false,
          "defaultValue": {
              "stringValue": "myModelId"
          }
      },
      "value": {
          "stringValue": "myModelId"
```

```
      }
    },
    "tableName": {
      "definition": {
          "dataType": { "type": "STRING" },
          "isExternalId": false,
          "isFinal": false,
          "isImported": false,
          "isInherited": false,
          "isRequiredInEntity": false,
          "isStoredExternally": false,
          "isTimeSeries": false,
          "defaultValue": {
              "stringValue": "myTableName"
          }
      },
      "value": {
          "stringValue": "myTableName"
      }
    }
  }
}
```

Respuesta:

```
{
  "properties": {
    "myProperty1": {
      "definition": {
        "dataType": {
          "type": "DOUBLE",
          "unitOfMeasure": "%"
        },
        "configuration": {
          "myProperty1Id": "idValue"
        },
        "isTimeSeries": true
      }
    },
    "myProperty2": {
      "definition": {
        "dataType": { "type": "STRING" },
        "isTimeSeries": false,
```

```
        "defaultValue": {
          "stringValue": "property2Value"
        }
      }
    }
  }
}
```

- Inicializador de esquemas en el ciclo de vida de la entidad:

  Solicitud:

```
{
  "workspaceId": "myWorkspace",
  "entityId": "myEntity",
  "componentName": "myComponent",
  "properties": {
    "assetId": {
      "definition": {
          "dataType": { "type": "STRING" },
          "isExternalId": true,
          "isFinal": true,
          "isImported": false,
          "isInherited": false,
          "isRequiredInEntity": true,
          "isStoredExternally": false,
          "isTimeSeries": false
      },
      "value": {
          "stringValue": "myAssetId"
      }
    },
    "tableName": {
      "definition": {
          "dataType": { "type": "STRING" },
          "isExternalId": false,
          "isFinal": false,
          "isImported": false,
          "isInherited": false,
          "isRequiredInEntity": false,
          "isStoredExternally": false,
          "isTimeSeries": false
      },
      "value": {
```

```
          "stringValue": "myTableName"
      }
    }
  }
}
```

Respuesta:

```
{
  "properties": {
    "myProperty1": {
      "definition": {
        "dataType": {
          "type": "DOUBLE",
          "unitOfMeasure": "%"
        },
        "configuration": {
          "myProperty1Id": "idValue"
        },
        "isTimeSeries": true
      }
    },
    "myProperty2": {
      "definition": {
        "dataType": { "type": "STRING" },
        "isTimeSeries": false
      },
      "value": {
        "stringValue": "property2Value"
      }
    }
  }
}
```

- DataReaderByEntity y DataReader:

  Solicitud:

```
{
  "workspaceId": "myWorkspace",
  "entityId": "myEntity",
  "componentName": "myComponent",
  "selectedProperties": [
```

```
    "Temperature",
    "Pressure"
  ],
  "startTime": "2022-04-07T04:04:42Z",
  "endTime": "2022-04-07T04:04:45Z",
  "maxResults": 4,
  "orderByTime": "ASCENDING",
  "properties": {
      "assetId": {
          "definition": {
              "dataType": { "type": "STRING" },
              "isExternalId": true,
              "isFinal": true,
              "isImported": false,
              "isInherited": false,
              "isRequiredInEntity": true,
              "isStoredExternally": false,
              "isTimeSeries": false
          },
          "value": {
              "stringValue": "myAssetId"
          }
      },
      "Temperature": {
          "definition": {
              "configuration": {
                  "temperatureId": "xyz123"
              },
              "dataType": {
                  "type": "DOUBLE",
                  "unitOfMeasure": "DEGC"
              },
              "isExternalId": false,
              "isFinal": false,
              "isImported": true,
              "isInherited": false,
              "isRequiredInEntity": false,
              "isStoredExternally": false,
              "isTimeSeries": true
          }
      },
      "Pressure": {
          "definition": {
              "configuration": {
```

```
                "pressureId": "xyz456"
            },
            "dataType": {
                "type": "DOUBLE",
                "unitOfMeasure": "MPA"
            },
            "isExternalId": false,
            "isFinal": false,
            "isImported": true,
            "isInherited": false,
            "isRequiredInEntity": false,
            "isStoredExternally": false,
            "isTimeSeries": true
        }
    }
  }
}
```

Respuesta:

```
{
  "propertyValues": [
    {
      "entityPropertyReference": {
        "entityId": "myEntity",
        "componentName": "myComponent",
        "propertyName": "Temperature"
      },
      "values": [
        {
          "time": "2022-04-07T04:04:42Z",
          "value": {
            "doubleValue": 588.168
          }
        },
        {
          "time": "2022-04-07T04:04:43Z",
          "value": {
            "doubleValue": 592.4224
          }
        }
      ]
    }
```

```
  ],
  "nextToken": "qwertyuiop"
}
```

- AttributePropertyValueReaderByEntity:

  Solicitud:

```
{
  "workspaceId": "myWorkspace",
  "entityId": "myEntity",
  "componentName": "myComponent",
  "selectedProperties": [
    "manufacturer",
  ],
  "properties": {
    "assetId": {
      "definition": {
        "dataType": { "type": "STRING" },
        "isExternalId": true,
        "isFinal": true,
        "isImported": false,
        "isInherited": false,
        "isRequiredInEntity": true,
        "isStoredExternally": false,
        "isTimeSeries": false
      },
      "value": {
          "stringValue": "myAssetId"
      }
    },
    "manufacturer": {
      "definition": {
        "dataType": { "type": "STRING" },
        "configuration": {
            "manufacturerPropId": "M001"
        },
        "isExternalId": false,
        "isFinal": false,
        "isImported": false,
        "isInherited": false,
        "isRequiredInEntity": false,
        "isStoredExternally": true,
        "isTimeSeries": false
```

```
      }
    }
  }
}
```

Respuesta:

```
{
  "propertyValues": {
    "manufacturer": {
      "propertyReference": {
        "propertyName": "manufacturer",
        "entityId": "myEntity",
        "componentName": "myComponent"
      },
      "propertyValue": {
        "stringValue": "Amazon"
      }
    }
  }
}
```

- DataWriter:

  Solicitud:

```
{
  "workspaceId": "myWorkspaceId",
  "properties": {
    "myEntity": {
      "Temperature": {
          "definition": {
              "configuration": {
                  "temperatureId": "xyz123"
              },
              "dataType": {
                  "type": "DOUBLE",
                  "unitOfMeasure": "DEGC"
              },
              "isExternalId": false,
              "isFinal": false,
              "isImported": true,
```

```
              "isInherited": false,
              "isRequiredInEntity": false,
              "isStoredExternally": false,
              "isTimeSeries": true
          }
      }
    }
  },
  "entries": [
    {
      "entryId": "myEntity",
      "entityPropertyReference": {
        "entityId": "myEntity",
        "componentName": "myComponent",
        "propertyName": "Temperature"
      },
      "propertyValues": [
        {
          "timestamp": 1626201120,
          "value": {
            "doubleValue": 95.6958
          }
        },
        {
          "timestamp": 1626201132,
          "value": {
            "doubleValue": 80.6959
          }
        }
      ]
    }
  ]
}
```

Respuesta:

```
{
    "errorEntries": [
        {
            "errors": [
                {
                    "errorCode": "409",
```

```
                    "errorMessage": "Conflict value at same timestamp",
                    "entry": {
                        "entryId": "myEntity",
                        "entityPropertyReference": {
                            "entityId": "myEntity",
                            "componentName": "myComponent",
                            "propertyName": "Temperature"
                        },
                        "propertyValues": [
                            "time": "2022-04-07T04:04:42Z",
                            "value": {
                                "doubleValue": 95.6958
                            }
                        ]
                    }
                }
            ]
        }
    ]
}
```

AWS IoT TwinMaker Conector de datos tabulares Athena

Con el conector de datos tabulares de Athena, puede acceder a sus almacenes de datos de Athena en AWS IoT TwinMaker. Puede utilizar los datos de Athena para crear gemelos digitales sin necesidad de realizar un esfuerzo intensivo de migración de datos. Puede usar el conector prediseñado o crear un conector Athena personalizado para acceder a los datos de sus fuentes de datos de Athena.

AWS IoT TwinMaker Requisitos previos del conector de datos Athena

Antes de utilizar el conector de datos tabulares Athena, complete los siguientes requisitos previos:

- Cree tablas de Athena administradas y sus recursos de Amazon S3 asociados. Para obtener información sobre el uso de Athena, consulte la documentación de Athena.
- Cree un AWS IoT TwinMaker espacio de trabajo. Puede crear un bucket utilizando la consola de AWS IoT TwinMaker.
- Actualiza el rol de IAM de tu espacio de trabajo con los permisos de Athena. Para obtener más información, consulte Modifique el rol de IAM de su espacio de trabajo para utilizar el conector de datos Athena.

- Familiarícese con AWS IoT TwinMaker el sistema de componentes de entidades y con cómo crear entidades. Para obtener más información, consulte Crear su primera entidad.
- Familiarícese con los conectores AWS IoT TwinMaker de datos. Para obtener más información, consulte AWS IoT TwinMaker conectores de datos.

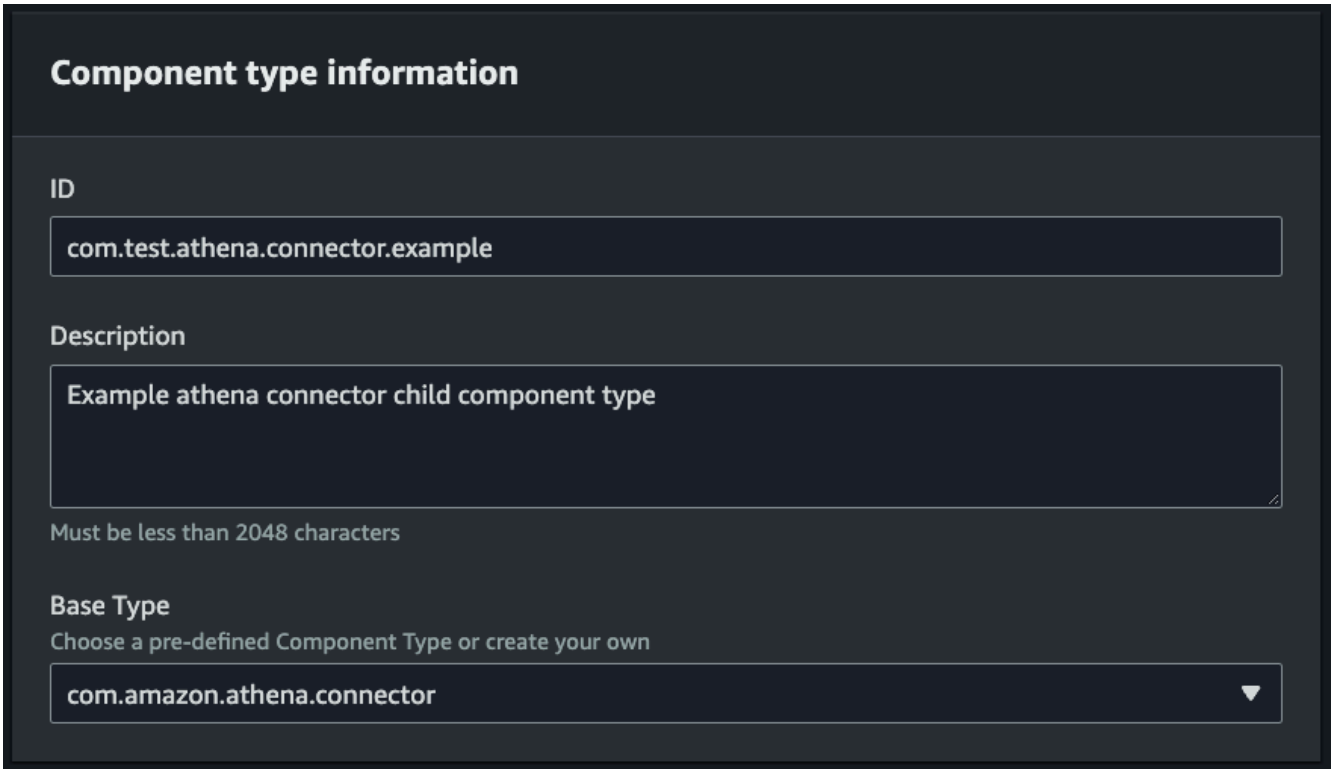
Uso del conector de datos de Athena

Para utilizar el conector de datos de Athena, debe crear un componente, utilizando el conector de Athena como tipo de componente. A continuación, debe conectar el componente a una entidad de la escena para usarlo en AWS IoT TwinMaker.

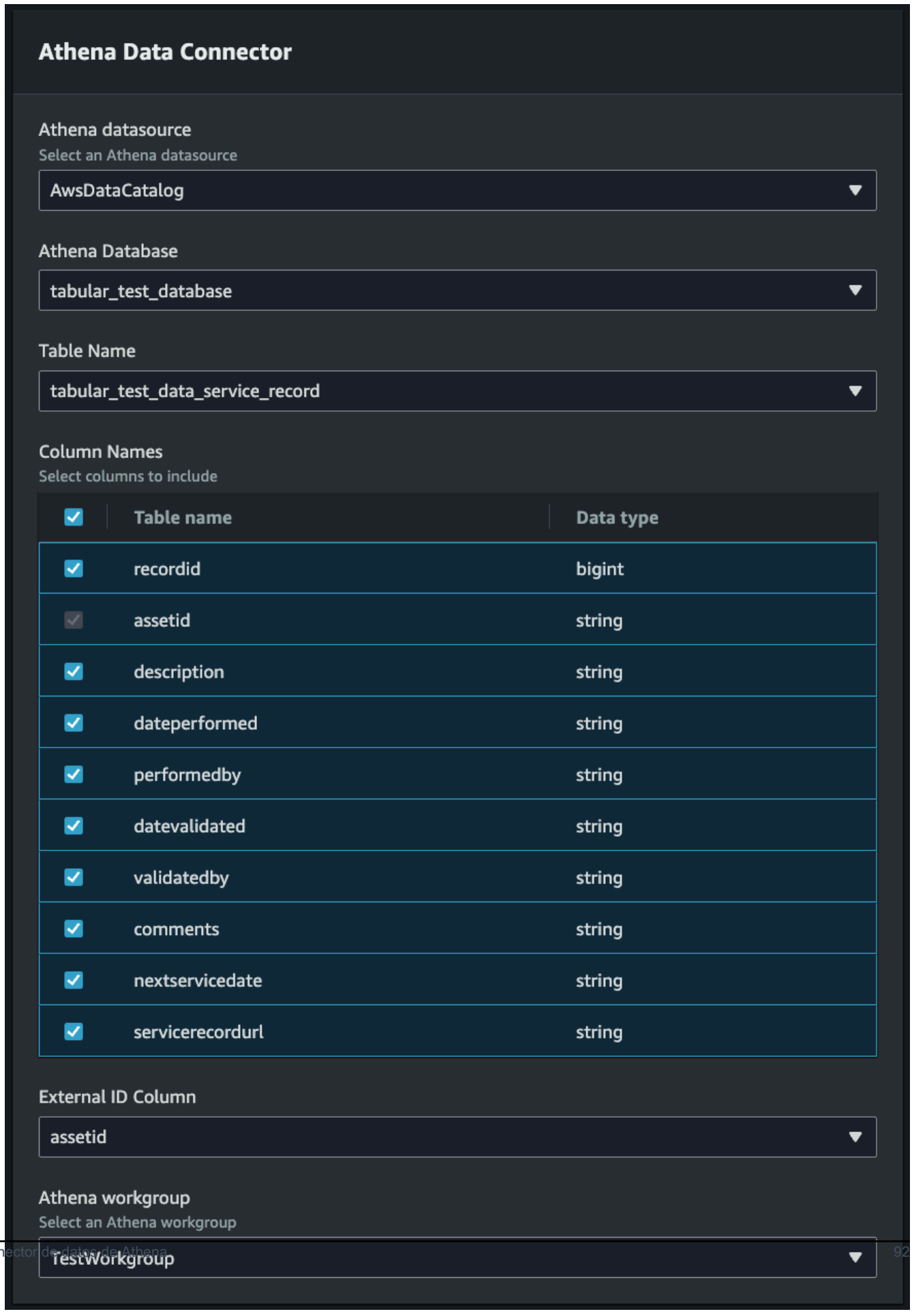
Crear un tipo de componente con el conector de datos de Athena

Utilice este procedimiento para crear un tipo de AWS IoT TwinMaker componente con el conector de datos tabulares Athena:

1. Vaya a la consola de AWS IoT TwinMaker.
2. Abra un espacio de trabajo existente o cree uno nuevo.
3. En el menú de navegación de la izquierda, elija Tipos de componentes y seleccione Crear tipo de componente para abrir la página de creación de tipos de componentes.
4. En la página Crear tipo de componente, rellene el campo ID con un ID que coincida con su caso de uso.



5. Elija el tipo Base (Básico). En la lista desplegable, seleccione el conector de datos tabulares de Athena, que está etiquetado como com.amazon.athena.connector.
6. Configure el origen de datos del tipo de componente seleccionando los recursos de Athena para los siguientes campos:
   - Elija una fuente de datos de Athena.
   - Elija una base de datos de Athena.
   - Elija un Nombre de la tabla.
   - Elija un grupo de trabajo de Athena.
7. Una vez que haya seleccionado los recursos de Athena que quiere usar como origen de datos, elija qué columnas de la tabla quiere incluir.
8. Seleccione un nombre de columna de ID externo. Seleccione una columna del paso anterior para que sirva como columna de ID externa. El identificador externo es el identificador que se utiliza para representar un activo de Athena y asignarlo a una AWS IoT TwinMaker entidad.


Athena Data Connector
Athena datasource
Select an Athena datasource
AwsDataCatalog
Athena Database
tabular_test_database
Table Name
tabular_test_data_service_record
Column Names
Select columns to include
Table name
Data type
recordid
bigint
assetid
string
description
string
dateperformed
string
performedby
string
datevalidated
string
validatedby
string
comments
string
nextservicedate
string
serviceRecordUrl
string
External ID Column
assetid
Athena workgroup
Select an Athena workgroup
TestWorkgroup

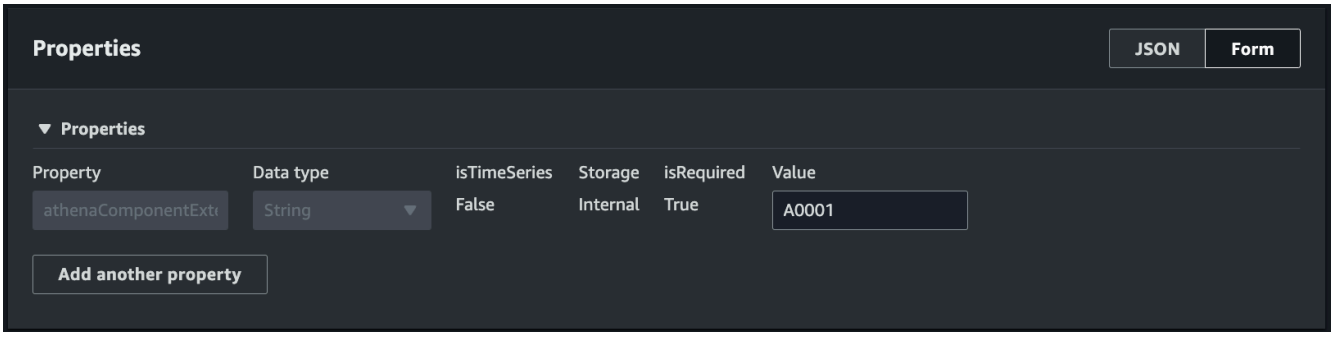
9. (Opcional) Agregue AWS etiquetas a estos recursos para poder agruparlos y organizarlos.
10. Seleccione Crear tipo de componente para terminar de crear el tipo de componente.

Crear un componente con el tipo de conector de datos de Athena y adjuntarlo a una entidad

Utilice este procedimiento para crear un AWS IoT TwinMaker componente con el conector de datos tabulares Athena y adjuntarlo a una entidad:

> **Note**
>
> Debe tener un tipo de componente existente que utilice el conector de datos tabulares de Athena como origen de datos para poder completar este procedimiento. Consulte el procedimiento anterior Crear un tipo de componente con el conector de datos de Athena antes de iniciar este tutorial.

1. Vaya a la consola de AWS IoT TwinMaker.
2. Abra un espacio de trabajo existente o cree uno nuevo.
3. En el menú de navegación de la izquierda, elija Entidades y seleccione la entidad a la que desee añadir el componente o crear una nueva entidad.
4. Cree una entidad nueva.
5. A continuación, seleccione Añadir componente, rellene el campo Nombre del componente con un nombre que coincida con su caso de uso.
6. En el menú desplegable del tipo de componente, seleccione el identificador del tipo de componente que creó en el procedimiento anterior.
7. Introduzca la información del componente, un nombre de componente y seleccione el elemento secundario ComponentType creado anteriormente. Este es el ComponentType que ha creado con el conector de datos Athena.
8. En la sección Propiedades, introduzca el athenaComponentExternalID del componente.



9. Seleccione Añadir componente para terminar de crear el componente.

Ahora ha creado correctamente un componente con el conector de datos de Athena como tipo de componente y lo ha conectado a una entidad.

Uso de la referencia JSON del conector de datos tabulares de Athena

El siguiente ejemplo es la referencia JSON completa del conector de datos tabulares de Athena. Úselo como recurso para crear conectores de datos y tipos de componentes personalizados.

```
{
    "componentTypeId": "com.amazon.athena.connector",
    "description": "Athena connector for syncing tabular data",
    "workspaceId":"AmazonOwnedTypesWorkspace",
    "propertyGroups": {
        "tabularPropertyGroup": {
            "groupType": "TABULAR",
            "propertyNames": []
        }
    },
    "propertyDefinitions": {
        "athenaDataSource": {
            "dataType": { "type": "STRING" },
            "isRequiredInEntity": true
        },
        "athenaDatabase": {
            "dataType": { "type": "STRING" },
            "isRequiredInEntity": true
        },
        "athenaTable": {
            "dataType": { "type": "STRING" },
            "isRequiredInEntity": true
        },
```

```
        "athenaWorkgroup": {
            "dataType": { "type": "STRING" },
            "isRequiredInEntity": true
        },
        "athenaExternalIdColumnName": {
            "dataType": { "type": "STRING" },
            "isRequiredInEntity": true,
            "isExternalId": false
        },
        "athenaComponentExternalId": {
            "dataType": { "type": "STRING" },
            "isStoredExternally": false,
            "isRequiredInEntity": true,
            "isExternalId": true
        }
    },
    "functions": {
        "tabularDataReaderByEntity": {
            "implementedBy": {
                "isNative": true
            }
        }
    }
}
```

Uso del conector de datos de Athena

Puede mostrar tus entidades que utilizan las tablas de Athena en Grafana. Para obtener más información, consulte Integración del panel de control de AWS IoT TwinMaker Grafana.

Lea la documentación de Athena para obtener información sobre la creación y el uso de tablas de Athena para almacenar datos.

Solución de problemas del conector de datos de Athena

En este tema se describen los problemas más comunes que pueden surgir al configurar el conector de datos Athena.

Ubicación del grupo de trabajo de Athena:

> Al crear el ComponentType del conector de Athena, un grupo de trabajo de Athena debe tener configurada la ubicación de salida. Consulte Cómo funcionan los grupos de trabajo.

Faltan permisos de roles de IAM:

Es posible que al rol AWS IoT TwinMaker; workspace le falte el permiso de acceso a la API de Athena al crear un ComponentType, añadir un componente de Ca a una entidad o ejecutar la API. GetPropertyValue Para actualizar los permisos de IAM, consulte Crear y administrar un rol de servicio para. AWS IoT TwinMaker

Visualice los datos tabulares de Athena en Grafana

También hay disponible un complemento de Grafana para visualizar sus datos tabulares en Grafana, un panel de control con funciones adicionales, como ordenar y filtrar según las propiedades seleccionadas sin realizar llamadas a la API AWS IoT TwinMaker ni interactuar con Athena. En este tema se muestra cómo configurar Grafana para visualizar los datos tabulares de Athena.

Requisitos previos

Antes de configurar un panel de Grafana para visualizar los datos tabulares de Athena, revise los siguientes requisitos previos:

- Ha configurado un entorno de Grafana. Para obtener más información, consulte Integración de AWS IoT TwinMaker Grafana.
- Puede configurar un origen de datos de Grafana. Para obtener más información, consulte Grafana AWS IoT TwinMaker.
- Está familiarizado con la creación de un nuevo panel de control y con la adición de uno nuevo.

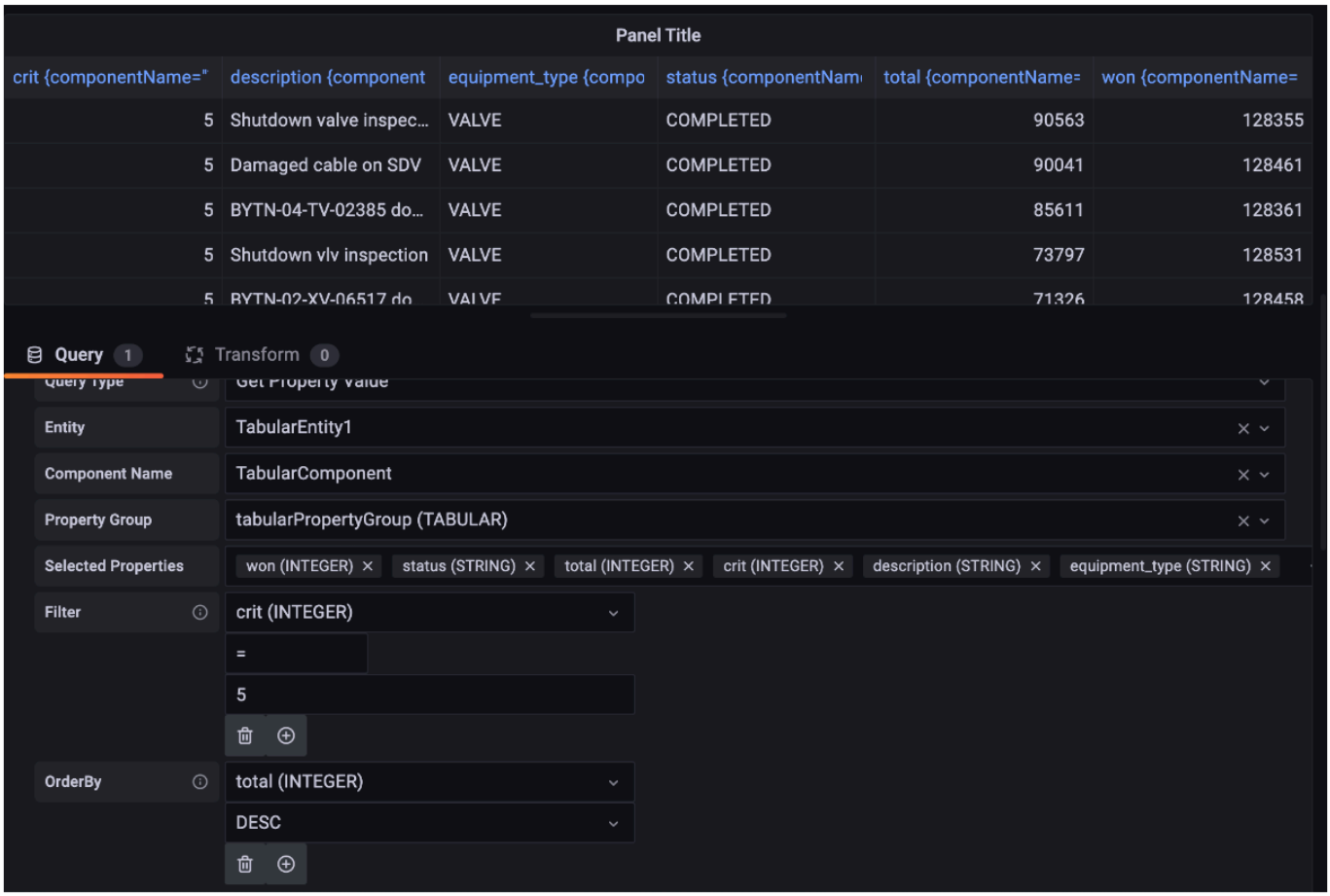
Visualice los datos tabulares de Athena en Grafana

Este procedimiento le muestra cómo configurar un panel de Grafana para visualizar los datos tabulares de Athena.

1. Abre el panel de control de AWS IoT TwinMaker Grafana.
2. Seleccione el panel Tabla en la configuración del panel.
3. Seleccione su origen de datos en la configuración de la consulta.
4. Seleccione la consulta Obtener el valor de la propiedad.
5. Seleccione una entidad.

6. Seleccione un componente que tenga un ComponentType que amplíe el tipo de componente base de Athena.
7. Seleccione el grupo de propiedades de su tabla de Athena.
8. Seleccione cualquier número de propiedades del grupo de propiedades.
9. Configure las condiciones tabulares mediante una lista de filtros y órdenes de propiedades. Con las siguientes opciones:

   - Filter: defina una expresión para el valor de una propiedad para filtrar sus datos.
   - OrderBy: especifique si los datos de una propiedad deben devolverse en orden ascendente o descendente.



Desarrollo de conectores de datos de series temporales AWS IoT TwinMaker

En esta sección se explica cómo desarrollar un conector de datos de series temporales en un step-by-step proceso. Además, presentamos un ejemplo de conector de datos de series temporales basado en toda la muestra de una fábrica de galletas, que incluye modelos 3D, entidades, componentes, alarmas y conectores. La fuente de muestras de Cookie Factory está disponible en el GitHub repositorio de AWS IoT TwinMaker muestras.

Temas

- AWS IoT TwinMaker requisitos previos del conector de datos de series temporales
- Antecedentes del conector de datos de series temporales
- Desarrollo de un conector de datos de series temporales
- Mejora de su conector de datos
- Prueba del conector
- Seguridad
- Creación de AWS IoT TwinMaker recursos
- Siguientes pasos
- Conector de series temporales de ejemplo de fábrica de galletas de AWS IoT TwinMaker

AWS IoT TwinMaker requisitos previos del conector de datos de series temporales

Antes de desarrollar el conector de datos de series temporales, le recomendamos que realice las siguientes tareas:

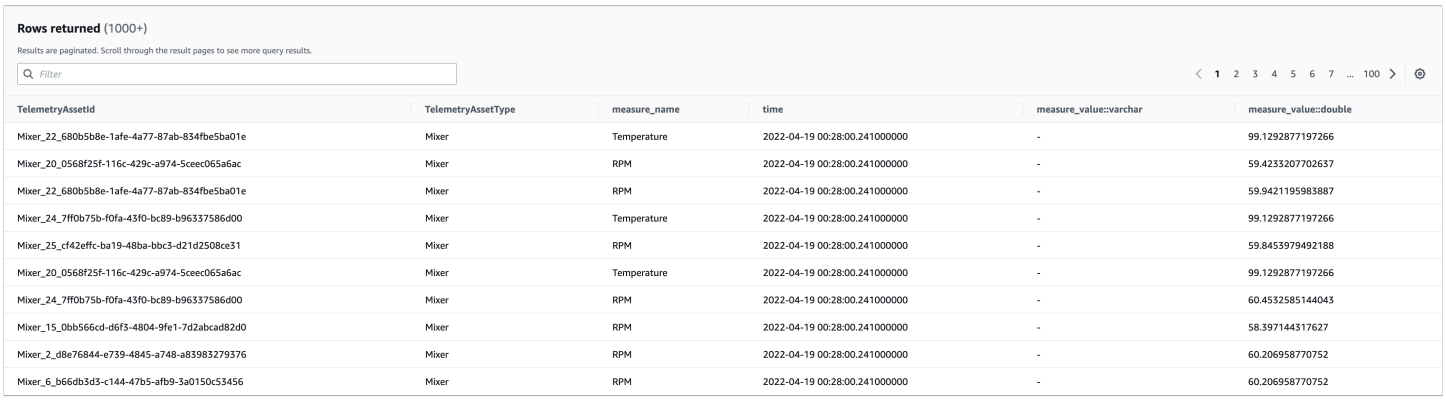
- Cree un espacio de trabajo de AWS IoT TwinMaker.
- Cree tipos de componentes de AWS IoT TwinMaker.
- Cree entidades de AWS IoT TwinMaker.
- (Opcional) Lea Uso y creación de tipos de componentes.
- (Opcional) Lea Interfaz del conector de datos de AWS IoT TwinMaker para obtener una comprensión general de los conectores de datos de AWS IoT TwinMaker .

> **Note**
>
> Para ver un ejemplo de un conector completamente implementado, consulte nuestro ejemplo de implementación de una fábrica de cookies.

Antecedentes del conector de datos de series temporales

Imagine que trabaja con una fábrica que tiene un conjunto de mezcladoras de galletas y un depósito de agua. Le gustaría crear gemelos AWS IoT TwinMaker digitales de estas entidades físicas para poder monitorear sus estados operativos comprobando varias métricas de series temporales.

Ha configurado los sensores in situ y ya está transmitiendo los datos de medición a una base de datos de Timestream. Desea poder ver y organizar los datos de medición de AWS IoT TwinMaker con una sobrecarga mínima. Puede realizar esta tarea mediante un conector de datos de series temporales. La siguiente imagen muestra un ejemplo de tabla de telemetría, que se completa mediante el uso de un conector de series temporales.



Rows returned (1000+)

Results are paginated. Scroll through the result pages to see more query results.

Filter

1 2 3 4 5 6 7 ... 100

TelemetryAssetId	TelemetryAssetType	measure_name	time	measure_value::varchar	measure_value::double
Mixer_22_680b5b8e-1afe-4a77-87ab-834fbe5ba01e	Mixer	Temperature	2022-04-19 00:28:00.241000000	-	99.1292877197266
Mixer_20_0568f25f-116c-429c-a974-5ceec065a6ac	Mixer	RPM	2022-04-19 00:28:00.241000000	-	59.4233207702637
Mixer_22_680b5b8e-1afe-4a77-87ab-834fbe5ba01e	Mixer	RPM	2022-04-19 00:28:00.241000000	-	59.9421195983887
Mixer_24_7ff0b75b-f0fa-43f0-bc89-b96337586d00	Mixer	Temperature	2022-04-19 00:28:00.241000000	-	99.1292877197266
Mixer_25_cf42effc-ba19-48ba-bbc3-d21d2508ce31	Mixer	RPM	2022-04-19 00:28:00.241000000	-	59.8453979492188
Mixer_20_0568f25f-116c-429c-a974-5ceec065a6ac	Mixer	Temperature	2022-04-19 00:28:00.241000000	-	99.1292877197266
Mixer_24_7ff0b75b-f0fa-43f0-bc89-b96337586d00	Mixer	RPM	2022-04-19 00:28:00.241000000	-	60.4532585144043
Mixer_15_0bb566cd-d6f3-4804-9fe1-7d2abcad82d0	Mixer	RPM	2022-04-19 00:28:00.241000000	-	58.397144317627
Mixer_2_d8e76844-e739-4845-a748-a83983279376	Mixer	RPM	2022-04-19 00:28:00.241000000	-	60.206958770752
Mixer_6_b66db3d3-c144-47b5-afb9-3a0150c53456	Mixer	RPM	2022-04-19 00:28:00.241000000	-	60.206958770752

Los conjuntos de datos y la tabla Timestream utilizados en esta captura de pantalla están disponibles en el repositorio de muestras.AWS IoT TwinMaker GitHub Consulte también el ejemplo de conector de fábrica de cookies para la implementación, que produce el resultado que se muestra en la captura de pantalla anterior.

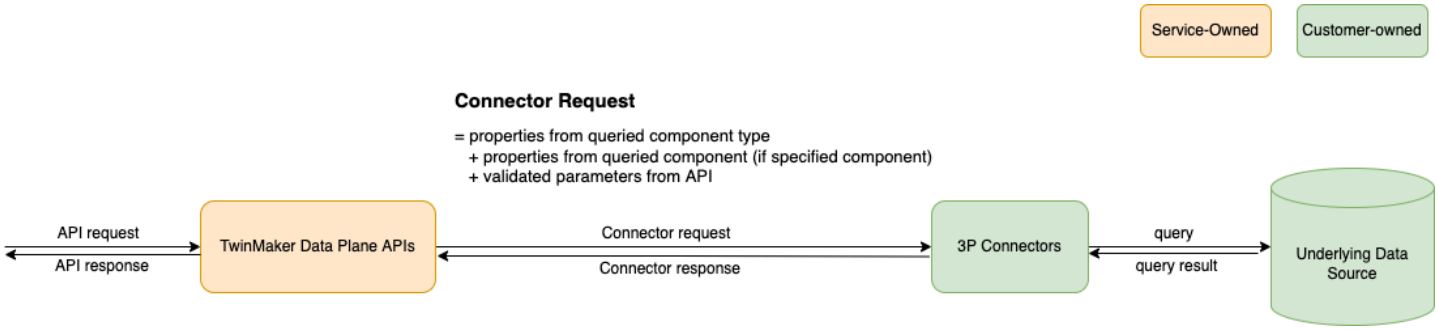
Flujo de datos de conector de datos de series temporales

Para las consultas del plano de datos, AWS IoT TwinMaker busca las propiedades correspondientes de los componentes y los tipos de componentes a partir de las definiciones de componentes y tipos de componentes. AWS IoT TwinMaker reenvía las propiedades de las AWS Lambda funciones junto con cualquier parámetro de consulta de API de la consulta.

AWS IoT TwinMaker usa funciones Lambda para acceder y resolver consultas de fuentes de datos y devolver los resultados de esas consultas. Las funciones de Lambda utilizan las propiedades del componente y del tipo de componente del plano de datos para resolver la solicitud inicial.

Los resultados de la consulta de Lambda se asignan a una respuesta de API y se le devuelven.

AWS IoT TwinMaker define la interfaz del conector de datos y la utiliza para interactuar con las funciones de Lambda. Con los conectores de datos, puede consultar su fuente de datos desde la API de AWS IoT TwinMaker sin necesidad de realizar ninguna migración de datos. La siguiente imagen describe el flujo de datos básico descrito en los párrafos anteriores.



Desarrollo de un conector de datos de series temporales

El siguiente procedimiento describe un modelo de desarrollo que se convierte gradualmente en un conector de datos de series temporales funcional. Los pasos básicos son los siguientes:

1. Cree un tipo de componente básico válido

   En un tipo de componente, se definen propiedades comunes que se comparten entre los componentes. Para obtener más información sobre la definición de tipos de componentes, consulte Uso y creación de tipos de componentes.

   AWS IoT TwinMaker utiliza un patrón de modelado de componentes de entidad para que cada componente esté asociado a una entidad. Se recomienda modelar cada elemento físico como una entidad y modelar diferentes fuentes de datos con sus propios tipos de componentes.

   El siguiente ejemplo muestra un tipo de componente de plantilla Timestream con una propiedad:

```
{"componentTypeId": "com.example.timestream-telemetry",
    "workspaceId": "MyWorkspace",
    "functions": {
        "dataReader": {
            "implementedBy": {
```

```
                "lambda": {
                    "arn": "lambdaArn"
                }
            }
        }
    },
    "propertyDefinitions": {
        "telemetryType": {
            "dataType": { "type": "STRING" },
            "isExternalId": false,
            "isStoredExternally": false,
            "isTimeSeries": false,
            "isRequiredInEntity": true
        },
        "telemetryId": {
            "dataType": { "type": "STRING" },
            "isExternalId": true,
            "isStoredExternally": false,
            "isTimeSeries": false,
            "isRequiredInEntity": true
        },
        "Temperature": {
            "dataType": { "type": "DOUBLE" },
            "isExternalId": false,
            "isTimeSeries": true,
            "isStoredExternally": true,
            "isRequiredInEntity": false
        }
    }
}
```

Los elementos clave del tipo de componente son los siguientes:

- La `telemetryId` propiedad identifica la clave única del elemento físico en la fuente de datos correspondiente. El conector de datos utiliza esta propiedad como condición de filtro para consultar únicamente los valores asociados al elemento en cuestión. Además, si incluye el valor de la propiedad `telemetryId` en la respuesta de la API del plano de datos, el cliente toma el ID y puede realizar una búsqueda inversa si es necesario.
- El campo `lambdaArn` identifica la función de Lambda con la que interactúa el tipo de componente.

- El indicador `isRequiredInEntity` impone la creación de la ID. Esta marca es necesaria para que, cuando se cree el componente, también se cree una instancia del ID del elemento.
- `TelemetryId`Se agrega al tipo de componente como un identificador externo para que el elemento pueda identificarse en la tabla Timestream.

2. Cree un componente con el tipo de componente

   Para usar el tipo de componente que creó, debe crear un componente y adjuntarlo a la entidad de la que desea recuperar los datos. Los siguientes pasos detallan el proceso de creación de ese componente:

   a. Vaya a la consola de AWS IoT TwinMaker.

   b. Seleccione y abra el mismo espacio de trabajo en el que creó los tipos de componentes.

   c. Vaya a la página de la entidad.

   d. Cree una entidad nueva o seleccione una entidad existente de la tabla.

   e. Una vez que haya seleccionado la entidad que desea usar, elija Agregar componente para abrir la página Agregar componente.

   f. Asigne un nombre al componente y, para el Tipo, elija el tipo de componente que creó con la plantilla en el punto 1. Cree un tipo de componente básico válido

3. Haga que su tipo de componente se llame conector Lambda

   El conector Lambda debe acceder al origen de datos y generar la declaración de consulta en función de la entrada y reenviarla al origen de datos. El siguiente ejemplo muestra una plantilla de solicitud JSON que hace esto.

```
{
  "workspaceId": "MyWorkspace",
  "entityId": "MyEntity",
  "componentName": "TelemetryData",
  "selectedProperties": ["Temperature"],
  "startTime": "2022-08-25T00:00:00Z",
  "endTime": "2022-08-25T00:00:05Z",
  "maxResults": 3,
  "orderByTime": "ASCENDING",
  "properties": {
      "telemetryType": {
          "definition": {
              "dataType": { "type": "STRING" },
              "isExternalId": false,
```

```
                "isFinal": false,
                "isImported": false,
                "isInherited": false,
                "isRequiredInEntity": false,
                "isStoredExternally": false,
                "isTimeSeries": false
            },
            "value": {
                "stringValue": "Mixer"
            }
        },
        "telemetryId": {
            "definition": {
                "dataType": { "type": "STRING" },
                "isExternalId": true,
                "isFinal": true,
                "isImported": false,
                "isInherited": false,
                "isRequiredInEntity": true,
                "isStoredExternally": false,
                "isTimeSeries": false
            },
            "value": {
                "stringValue": "item_A001"
            }
        },
        "Temperature": {
            "definition": {
                "dataType": { "type": "DOUBLE", },
                "isExternalId": false,
                "isFinal": false,
                "isImported": true,
                "isInherited": false,
                "isRequiredInEntity": false,
                "isStoredExternally": false,
                "isTimeSeries": true
            }
        }
    }
}
```

Los elementos clave de la solicitud:

- `selectedProperties` es una lista que debe rellenar con las propiedades para las que desea obtener mediciones de Timestream.
- Los campos `startDateTime`, `startTime`, `EndDateTime`, y `endTime` especifican un intervalo de tiempo para la solicitud. Esto determina el rango de muestra para las mediciones devueltas.
- `entityId` es el nombre de la entidad desde la que se consultan los datos.
- `componentName` es el nombre del componente desde el que se consultan los datos.
- Utilice el campo `orderByTime` para organizar el orden en el que se muestran los resultados.

En la solicitud de ejemplo anterior, esperaríamos obtener una serie de muestras de las propiedades seleccionadas durante el intervalo de tiempo determinado para el elemento en cuestión, con el orden temporal seleccionado. La declaración de respuesta se puede resumir de la siguiente manera:

```
{
  "propertyValues": [
    {
      "entityPropertyReference": {
        "entityId": "MyEntity",
        "componentName": "TelemetryData",
        "propertyName": "Temperature"
      },
      "values": [
        {
          "time": "2022-08-25T00:00:00Z",
          "value": {
            "doubleValue": 588.168
          }
        },
        {
          "time": "2022-08-25T00:00:01Z",
          "value": {
            "doubleValue": 592.4224
          }
        },
        {
          "time": "2022-08-25T00:00:02Z",
          "value": {
            "doubleValue": 594.9383
```

```
          }
        }
      ]
    }
  ],
  "nextToken": "..."
}
```

4. Actualice el tipo de componente para que tenga dos propiedades

   La siguiente plantilla JSON muestra un tipo de componente válido con dos propiedades:

```
{
    "componentTypeId": "com.example.timestream-telemetry",
    "workspaceId": "MyWorkspace",
    "functions": {
        "dataReader": {
            "implementedBy": {
                "lambda": {
                    "arn": "lambdaArn"
                }
            }
        }
    },
    "propertyDefinitions": {
        "telemetryType": {
            "dataType": { "type": "STRING" },
            "isExternalId": false,
            "isStoredExternally": false,
            "isTimeSeries": false,
            "isRequiredInEntity": true
        },
        "telemetryId": {
            "dataType": { "type": "STRING" },
            "isExternalId": true,
            "isStoredExternally": false,
            "isTimeSeries": false,
            "isRequiredInEntity": true
        },
        "Temperature": {
            "dataType": { "type": "DOUBLE" },
            "isExternalId": false,
            "isTimeSeries": true,
            "isStoredExternally": true,
```

```
            "isRequiredInEntity": false
        },
        "RPM": {
            "dataType": { "type": "DOUBLE" },
            "isExternalId": false,
            "isTimeSeries": true,
            "isStoredExternally": true,
            "isRequiredInEntity": false
        }
    }
}
```

5. Actualice el conector Lambda para gestionar la segunda propiedad

   La API del plano de AWS IoT TwinMaker datos permite consultar varias propiedades en una sola solicitud y, al AWS IoT TwinMaker enviar una lista de`selectedProperties`, sigue una única solicitud a un conector.

   La siguiente solicitud de JSON muestra una plantilla modificada que ahora admite una solicitud de dos propiedades.

```
{
  "workspaceId": "MyWorkspace",
  "entityId": "MyEntity",
  "componentName": "TelemetryData",
  "selectedProperties": ["Temperature", "RPM"],
  "startTime": "2022-08-25T00:00:00Z",
  "endTime": "2022-08-25T00:00:05Z",
  "maxResults": 3,
  "orderByTime": "ASCENDING",
  "properties": {
      "telemetryType": {
          "definition": {
              "dataType": { "type": "STRING" },
              "isExternalId": false,
              "isFinal": false,
              "isImported": false,
              "isInherited": false,
              "isRequiredInEntity": false,
              "isStoredExternally": false,
              "isTimeSeries": false
          },
          "value": {
```

```
            "stringValue": "Mixer"
        }
    },
    "telemetryId": {
        "definition": {
            "dataType": { "type": "STRING" },
            "isExternalId": true,
            "isFinal": true,
            "isImported": false,
            "isInherited": false,
            "isRequiredInEntity": true,
            "isStoredExternally": false,
            "isTimeSeries": false
        },
        "value": {
            "stringValue": "item_A001"
        }
    },
    "Temperature": {
        "definition": {
            "dataType": { "type": "DOUBLE" },
            "isExternalId": false,
            "isFinal": false,
            "isImported": true,
            "isInherited": false,
            "isRequiredInEntity": false,
            "isStoredExternally": false,
            "isTimeSeries": true
        }
    },
    "RPM": {
        "definition": {
            "dataType": { "type": "DOUBLE" },
            "isExternalId": false,
            "isFinal": false,
            "isImported": true,
            "isInherited": false,
            "isRequiredInEntity": false,
            "isStoredExternally": false,
            "isTimeSeries": true
        }
    }
}
```

```
}
```

Del mismo modo, la respuesta correspondiente también se actualiza, como se muestra en el siguiente ejemplo:

```
{
  "propertyValues": [
    {
      "entityPropertyReference": {
        "entityId": "MyEntity",
        "componentName": "TelemetryData",
        "propertyName": "Temperature"
      },
      "values": [
        {
          "time": "2022-08-25T00:00:00Z",
          "value": {
            "doubleValue": 588.168
          }
        },
        {
          "time": "2022-08-25T00:00:01Z",
          "value": {
            "doubleValue": 592.4224
          }
        },
        {
          "time": "2022-08-25T00:00:02Z",
          "value": {
            "doubleValue": 594.9383
          }
        }
      ]
    },
    {
      "entityPropertyReference": {
        "entityId": "MyEntity",
        "componentName": "TelemetryData",
        "propertyName": "RPM"
      },
      "values": [
        {
          "time": "2022-08-25T00:00:00Z",
```

```
          "value": {
            "doubleValue": 59
          }
        },
        {
          "time": "2022-08-25T00:00:01Z",
          "value": {
            "doubleValue": 60
          }
        },
        {
          "time": "2022-08-25T00:00:02Z",
          "value": {
            "doubleValue": 60
          }
        }
      ]
    }
  ],
  "nextToken": "..."
}
```

> **Note**
>
> En cuanto a la paginación, en este caso, el tamaño de página de la solicitud se aplica a todas las propiedades. Esto significa que con cinco propiedades en la consulta y un tamaño de página de 100, si hay suficientes puntos de datos en la fuente, debería esperar ver 100 puntos de datos por propiedad, es decir, 500 puntos de datos en total.

Para ver un ejemplo de implementación, consulte el ejemplo del conector Snowflake en. GitHub

Mejora de su conector de datos

Tratamiento de excepciones

Es seguro que el conector Lambda genere excepciones. En la llamada a la API del plano de datos, el AWS IoT TwinMaker servicio espera a que la función Lambda devuelva una respuesta. Si la implementación del conector genera una excepción, AWS IoT TwinMaker traduce el tipo de

excepción como talConnectorFailure, lo que hace que el cliente de la API sepa que se ha producido un problema dentro del conector.

Gestión de la paginación

En el ejemplo, Timestream proporciona una función de utilidad que puede ayudar a admitir la paginación de forma nativa. Sin embargo, para algunas otras interfaces de consulta, como SQL, puede que sea necesario un esfuerzo adicional implementar un algoritmo de paginación eficiente. Hay un ejemplo de conector Snowflake que gestiona la paginación en una interfaz SQL.

Cuando se devuelve el nuevo token a AWS IoT TwinMaker través de la interfaz de respuesta del conector, el token se cifra antes de devolverlo al cliente de la API. Cuando el token se incluye en otra solicitud, lo AWS IoT TwinMaker descifra antes de reenviarlo al conector Lambda. Se recomienda evitar añadir información confidencial al token.

Prueba del conector

Aunque todavía puede actualizar la implementación después de vincular el conector al tipo de componente, le recomendamos encarecidamente que compruebe el conector Lambda antes de integrarlo con AWS IoT TwinMaker.

Existen varias formas de probar el conector Lambda: puede probar el conector Lambda en la consola Lambda o localmente en la AWS CDK.

Para obtener más información sobre las pruebas de las funciones Lambda, consulte Probar funciones Lambda y Probar aplicaciones localmente. AWS CDK

Seguridad

Para obtener documentación sobre las mejores prácticas de seguridad con Timestream, consulte Seguridad en Timestream.

Para ver un ejemplo de prevención de inyecciones de SQL, consulte el siguiente script de Python en el GitHub repositorio de AWS IoT TwinMaker muestras.

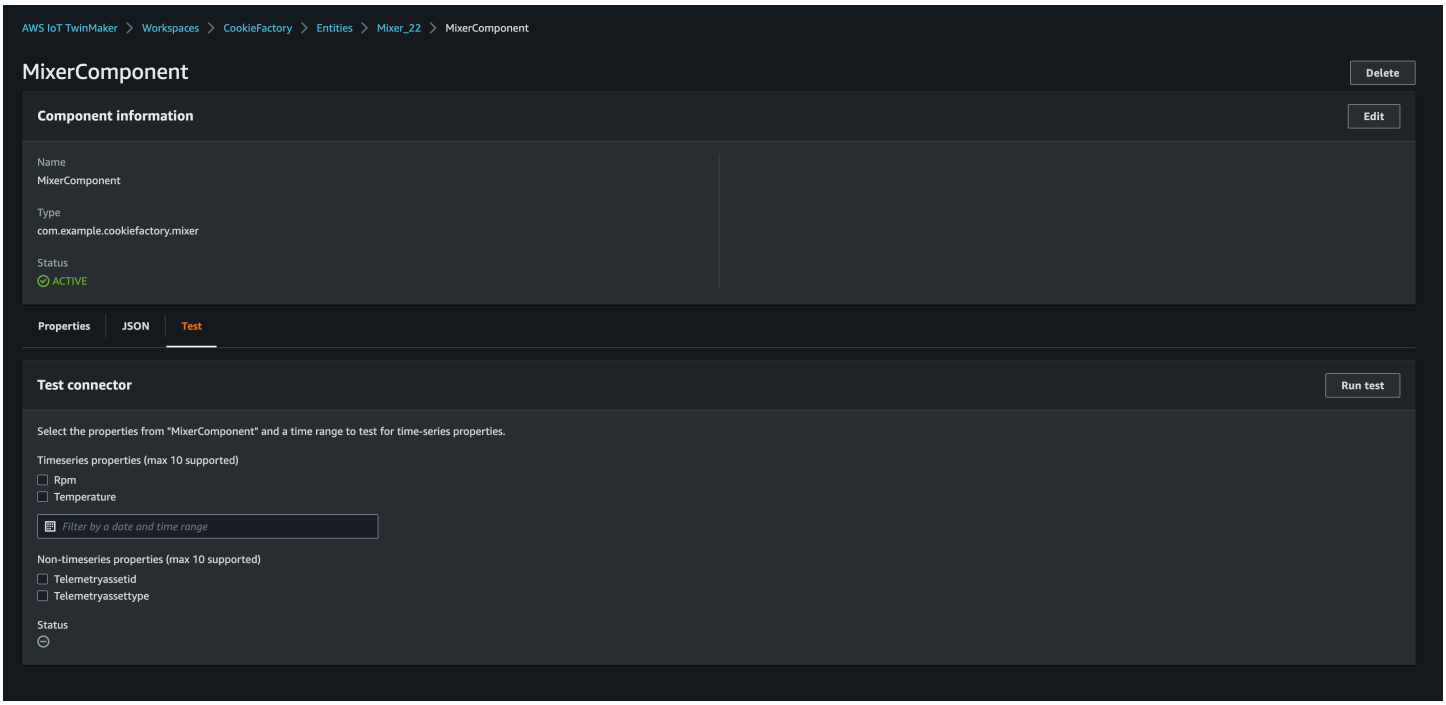
Creación de AWS IoT TwinMaker recursos

Una vez implementada la función Lambda, puede crear AWS IoT TwinMaker recursos como tipos de componentes, entidades y componentes a través de la AWS IoT TwinMaker consola o la API.

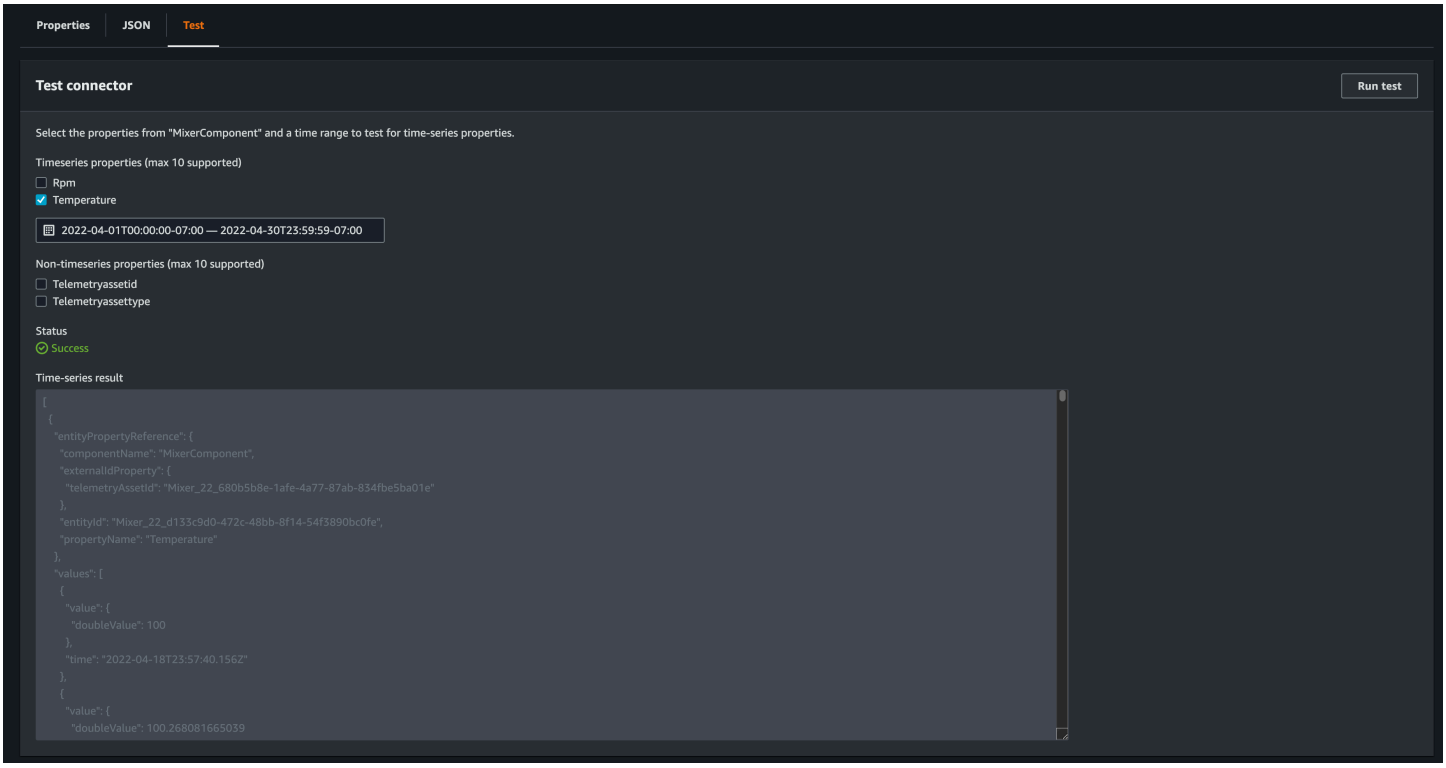
> **Note**
>
> Si sigue las instrucciones de configuración del GitHub ejemplo, todos los AWS IoT TwinMaker recursos estarán disponibles automáticamente. Puede comprobar las definiciones de los tipos de componentes en el AWS IoT TwinMaker GitHub ejemplo. Una vez que algún componente utiliza el tipo de componente, las definiciones de propiedades y las funciones del tipo de componente no se pueden actualizar.

Prueba de integración

Recomendamos realizar una prueba integrada AWS IoT TwinMaker para comprobar que la consulta del plano de datos funciona end-to-end. Puede realizarla a través de la GetPropertyValueHistoryAPI o fácilmente en AWS IoT TwinMaker la consola.



En la AWS IoT TwinMaker consola, vaya a los detalles del componente y, a continuación, en la sección Prueba, verá que todas las propiedades del componente se muestran allí. El área de pruebas de la consola permite probar propiedades y propiedades de series temporales. non-time-series Para las propiedades de series temporales, también puede utilizar la GetPropertyValueHistoryAPI y, para las non-time-series propiedades, utilizar GetPropertyValuela API. Si su conector Lambda admite consultas de propiedades múltiples, puede elegir más de una propiedad.



Siguientes pasos

Ahora puede configurar un panel de AWS IoT TwinMaker Grafana para visualizar las métricas. También puede explorar otros ejemplos de conectores de datos en el GitHub repositorio de AWS IoT TwinMaker muestras para ver si se ajustan a su caso de uso.

Conector de series temporales de ejemplo de fábrica de galletas de AWS IoT TwinMaker

El código completo de la función Lambda de la fábrica de cookies está disponible en. GitHub Aunque todavía puede actualizar la implementación después de vincular el conector al tipo de componente, le recomendamos encarecidamente que compruebe el conector Lambda antes de integrarlo con AWS IoT TwinMaker. También puede crear la función de Lambda con la consola de Lambda o localmente en AWS CDK. Para obtener más información sobre las pruebas de las funciones de Lambda, consulte Prueba de funciones Lambda y Prueba local de aplicaciones. AWS CDK

Ejemplos de tipos de componentes de fábrica de galletas

En un tipo de componente, se definen propiedades comunes que se comparten entre los componentes. Para el ejemplo de la fábrica de cookies, los componentes físicos del mismo tipo

comparten las mismas medidas, por lo que podemos definir el esquema de medidas en el tipo de componente. Como ejemplo, el tipo de mezclador se define en el siguiente ejemplo.

```
{
    "componentTypeId": "com.example.cookiefactory.mixer"
    "propertyDefinitions": {
        "RPM": {
            "dataType": { "type": "DOUBLE" },
            "isTimeSeries": true,
            "isRequiredInEntity": false,
            "isExternalId": false,
            "isStoredExternally": true
        },
        "Temperature": {
            "dataType": { "type": "DOUBLE" },
            "isTimeSeries": true,
            "isRequiredInEntity": false,
            "isExternalId": false,
            "isStoredExternally": true
        }
    }
}
```

Por ejemplo, un componente físico puede tener mediciones en una base de datos de Timestream, registros de mantenimiento en una base de datos SQL o datos de alarma en sistemas de alarma. Al crear varios componentes y asociarlos a una entidad, se vinculan diferentes orígenes de datos a la entidad y se completa el gráfico entidad-componente. En este contexto, cada componente necesita una `telemetryId` propiedad para identificar la clave única del componente en la fuente de datos correspondiente. Especificar la `telemetryId` propiedad tiene dos ventajas: la propiedad se puede utilizar en el conector de datos como condición de filtro para consultar únicamente los valores del componente en cuestión y, si se incluye el valor de la `telemetryId` propiedad en la respuesta de la API del plano de datos, el cliente toma el ID y puede realizar una búsqueda inversa si es necesario.

Si se añade `TelemetryId` al tipo de componente como identificador externo, se identifica el componente en la `TimeStream` tabla.

```
{
    "componentTypeId": "com.example.cookiefactory.mixer"
    "propertyDefinitions": {
        "telemetryId": {
            "dataType": { "type": "STRING" },
```

```
            "isTimeSeries": false,
            "isRequiredInEntity": true,
            "isExternalId": true,
            "isStoredExternally": false
        },
        "RPM": {
            "dataType": { "type": "DOUBLE" },
            "isTimeSeries": true,
            "isRequiredInEntity": false,
            "isExternalId": false,
            "isStoredExternally": true
        },
        "Temperature": {
            "dataType": { "type": "DOUBLE" },
            "isTimeSeries": true,
            "isRequiredInEntity": false,
            "isExternalId": false,
            "isStoredExternally": true
        }
    }
}
```

Del mismo modo, tenemos el tipo de componente para el `WaterTank`, como se muestra en el siguiente ejemplo JSON.

```
{
  "componentTypeId": "com.example.cookiefactory.watertank",
  "propertyDefinitions": {
    "flowRate1": {
      "dataType": { "type": "DOUBLE" },
      "isTimeSeries": true,
      "isRequiredInEntity": false,
      "isExternalId": false,
      "isStoredExternally": true
    },
    "flowrate2": {
      "dataType": { "type": "DOUBLE" },
      "isTimeSeries": true,
      "isRequiredInEntity": false,
      "isExternalId": false,
      "isStoredExternally": true
    },
    "tankVolume1": {
```

```
      "dataType": { "type": "DOUBLE" },
      "isTimeSeries": true,
      "isRequiredInEntity": false,
      "isExternalId": false,
      "isStoredExternally": true
    },
    "tankVolume2": {
      "dataType": { "type": "DOUBLE" },
      "isTimeSeries": true,
      "isRequiredInEntity": false,
      "isExternalId": false,
      "isStoredExternally": true
    },
    "telemetryId": {
      "dataType": { "type": "STRING" },
      "isTimeSeries": false,
      "isRequiredInEntity": true,
      "isExternalId": true,
      "isStoredExternally": false
    }
  }
}
```

`TelemetryType` es una propiedad opcional en el tipo de componente si su objetivo es consultar los valores de las propiedades en el ámbito de la entidad. Para ver un ejemplo, consulte los tipos de componentes definidos en el GitHub repositorio de AWS IoT TwinMaker muestras. También hay tipos de alarma integrados en la misma tabla, por lo que se define `TelemetryType` y se extraen propiedades comunes, como `TelemetryId` y `TelemetryType` a un tipo de componente principal para que otros tipos secundarios las compartan.

Ejemplo de Lambda

El conector Lambda debe acceder al origen de datos y generar la declaración de consulta en función de la entrada y reenviarla al origen de datos. En el siguiente ejemplo de JSON se muestra un ejemplo de solicitud enviada a la .

```
{
    'workspaceId': 'CookieFactory',
    'selectedProperties': ['Temperature'],
    'startDateTime': 1648796400,
    'startTime': '2022-04-01T07:00:00.000Z',
    'endDateTime': 1650610799,
```

```
    'endTime': '2022-04-22T06:59:59.000Z',
    'properties': {
        'telemetryId': {
            'definition': {
                'dataType': { 'type': 'STRING' },
                'isTimeSeries': False,
                'isRequiredInEntity': True,
                'isExternalId': True,
                'isStoredExternally': False,
                'isImported': False,
                'isFinal': False,
                'isInherited': True,
            },
            'value': {
                'stringValue': 'Mixer_22_680b5b8e-1afe-4a77-87ab-834fbe5ba01e'
            }
        }
        'Temperature': {
            'definition': {
                'dataType': { 'type': 'DOUBLE' },
                'isTimeSeries': True,
                'isRequiredInEntity': False,
                'isExternalId': False,
                'isStoredExternally': True,
                'isImported': False,
                'isFinal': False,
                'isInherited': False
            }
        }
        'RPM': {
            'definition': {
                'dataType': { 'type': 'DOUBLE' },
                'isTimeSeries': True,
                'isRequiredInEntity': False,
                'isExternalId': False,
                'isStoredExternally': True,
                'isImported': False,
                'isFinal':False,
                'isInherited': False
            }
        },
    'entityId': 'Mixer_22_d133c9d0-472c-48bb-8f14-54f3890bc0fe',
    'componentName': 'MixerComponent',
    'maxResults': 100,
```

```
    'orderByTime': 'ASCENDING'
}
```

El objetivo de la función Lambda es consultar los datos de medición históricos de una entidad determinada. AWS IoT TwinMaker proporciona un mapa de propiedades de los componentes y debe especificar un valor instanciado para el ID del componente. Por ejemplo, para gestionar la consulta a nivel de tipo de componente (que es común en los casos de uso de alarmas) y devolver el estado de alarma de todos los componentes del espacio de trabajo, el mapa de propiedades incluye definiciones de propiedades de tipo de componente.

En el caso más sencillo, como en la solicitud anterior, queremos una serie de muestras de temperatura durante un intervalo de tiempo determinado para el componente en cuestión, en orden temporal ascendente. La declaración de consulta se puede resumir de la siguiente manera:

```
...
SELECT measure_name, time, measure_value::double
    FROM {database_name}.{table_name}
    WHERE time < from_iso8601_timestamp('{request.start_time}')
    AND time >= from_iso8601_timestamp('{request.end_time}')
    AND TelemetryId = '{telemetry_id}'
    AND measure_name = '{selected_property}'
    ORDER BY time {request.orderByTime}
...
```

Creación y edición de AWS IoT TwinMaker escenas

Las escenas son visualizaciones tridimensionales de su gemelo digital. Son la forma principal para editar su gemelo digital. Aprenda a agregar alarmas, datos de series temporales, superposiciones de colores, etiquetas y reglas visuales a su escena para alinear las visualizaciones de sus gemelos digitales con su caso de uso real.

Esta sección abarca los siguientes temas:

- Antes de crear su primera escena
- Cargue los recursos a la biblioteca de AWS IoT TwinMaker recursos
- Cree sus escenas
- Agregue cámaras fijas a las entidades
- Edición de escenas mejoradas
- Edite sus escenas
- Formato de modelo de teselas 3D
- Escenas dinámicas

Antes de crear su primera escena

Las escenas dependen de los recursos para representar a su gemelo digital. Estos recursos se componen de modelos 3D, datos o archivos de textura. El tamaño y la complejidad de los recursos, los elementos de la escena, como la iluminación, y el hardware del ordenador, influyen en el rendimiento de las escenas de AWS IoT TwinMaker . Utilice la información de este tema para reducir el retraso, los tiempos de carga y mejorar la velocidad de fotogramas de las escenas.

Optimice sus recursos antes de importarlos a AWS IoT TwinMaker

Puedes usarlo AWS IoT TwinMaker para interactuar con tu gemelo digital en tiempo real. Para disfrutar de la mejor experiencia con sus escenas, le recomendamos que optimice sus recursos para usarlos en un entorno en tiempo real.

Sus modelos 3D pueden tener un impacto significativo en el rendimiento. La geometría y las mallas complejas del modelo pueden reducir el rendimiento. Por ejemplo, los modelos CAD industriales tienen un alto nivel de detalle. Recomendamos comprimir las mallas de estos modelos y reducir su número de polígonos antes de utilizarlas en AWS IoT TwinMaker las escenas. Si va a crear

nuevos modelos 3D para AWS IoT TwinMaker ellos, debe establecer un nivel de detalle y mantenerlo en todos sus modelos. Elimine los detalles de los modelos que no afecten a la visualización o la interpretación de su caso de uso.

Para comprimir los modelos y reducir el tamaño del archivo, utilice herramientas de compresión de malla de código abierto, como la compresión de datos 3D DRACO.

Las texturas no optimizadas también pueden afectar al rendimiento. Si no necesita ninguna transparencia en sus texturas, considere la posibilidad de elegir el formato de imagen PEG en lugar del formato PNG. Puede comprimir los archivos de textura mediante herramientas de compresión de texturas de código abierto, como la compresión de texturas Basis Universal.

Mejores prácticas de rendimiento en AWS IoT TwinMaker

Para obtener el mejor rendimiento AWS IoT TwinMaker, tenga en cuenta las siguientes limitaciones y prácticas recomendadas.

- AWS IoT TwinMaker el rendimiento del renderizado de escenas depende del hardware. El rendimiento varía según las distintas configuraciones de hardware del ordenador.
- Recomendamos un recuento total de polígonos inferior a 1 millón en todos los objetos de su AWS IoT TwinMaker.
- Recomendamos un total de 200 objetos por escena. Si se aumenta el número de objetos en una escena por encima de 200, se puede reducir la velocidad de fotogramas de la escena.
- Recomendamos que el tamaño total de todos los recursos 3D únicos de la escena no supere los 100 megabytes. De lo contrario, es posible que los tiempos de carga sean lentos o que el rendimiento se degrade en función del navegador y del hardware.
- Las escenas tienen iluminación ambiental de forma predeterminada. Puede añadir luces adicionales a una escena para enfocar determinados objetos o proyectar sombras sobre los objetos. Recomendamos usar una luz por escena. Usa luces donde sea necesario y evita replicar luces del mundo real dentro de una escena.

Más información

Utilice estos recursos para obtener más información sobre cómo utilizar técnicas de optimización para mejorar el rendimiento de sus escenas.

- Cómo convertir y comprimir modelos OBJ a GLTF para usarlos con AWS IoT TwinMaker

- Optimice sus modelos 3D para contenido web
- Optimización de escenas para un mejor rendimiento de WebGL

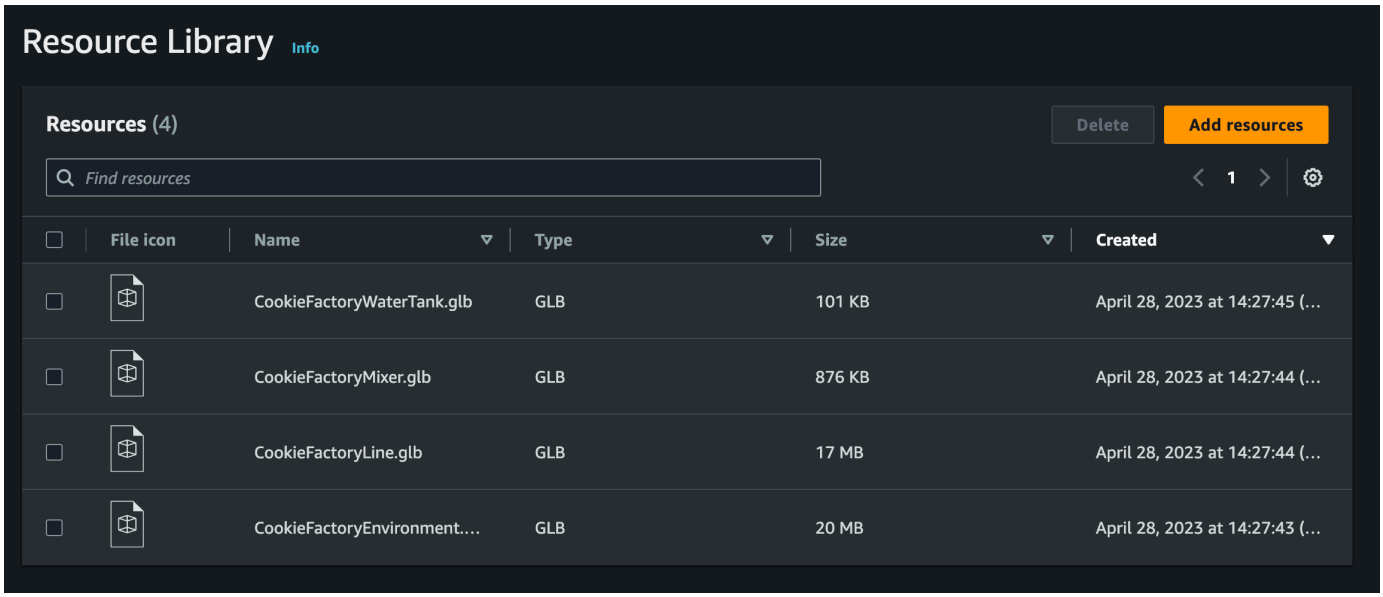
Cargue los recursos a la biblioteca de AWS IoT TwinMaker recursos

Puede usar la biblioteca de recursos para controlar y administrar cualquier recurso que desee colocar en las escenas de su aplicación gemela digital. Para AWS IoT TwinMaker conocer los recursos, cárguelos en la página de la consola de la biblioteca de recursos.

Cargue los archivos a la biblioteca de recursos mediante la consola

Siga estos pasos para añadir archivos a la biblioteca de recursos mediante la AWS IoT TwinMaker consola.

1. En el menú de navegación de la izquierda, en Espacios de trabajo, selecciona Biblioteca de recursos.
2. Selecciona Añadir recursos y elige los archivos que quieres cargar.



Cree sus escenas

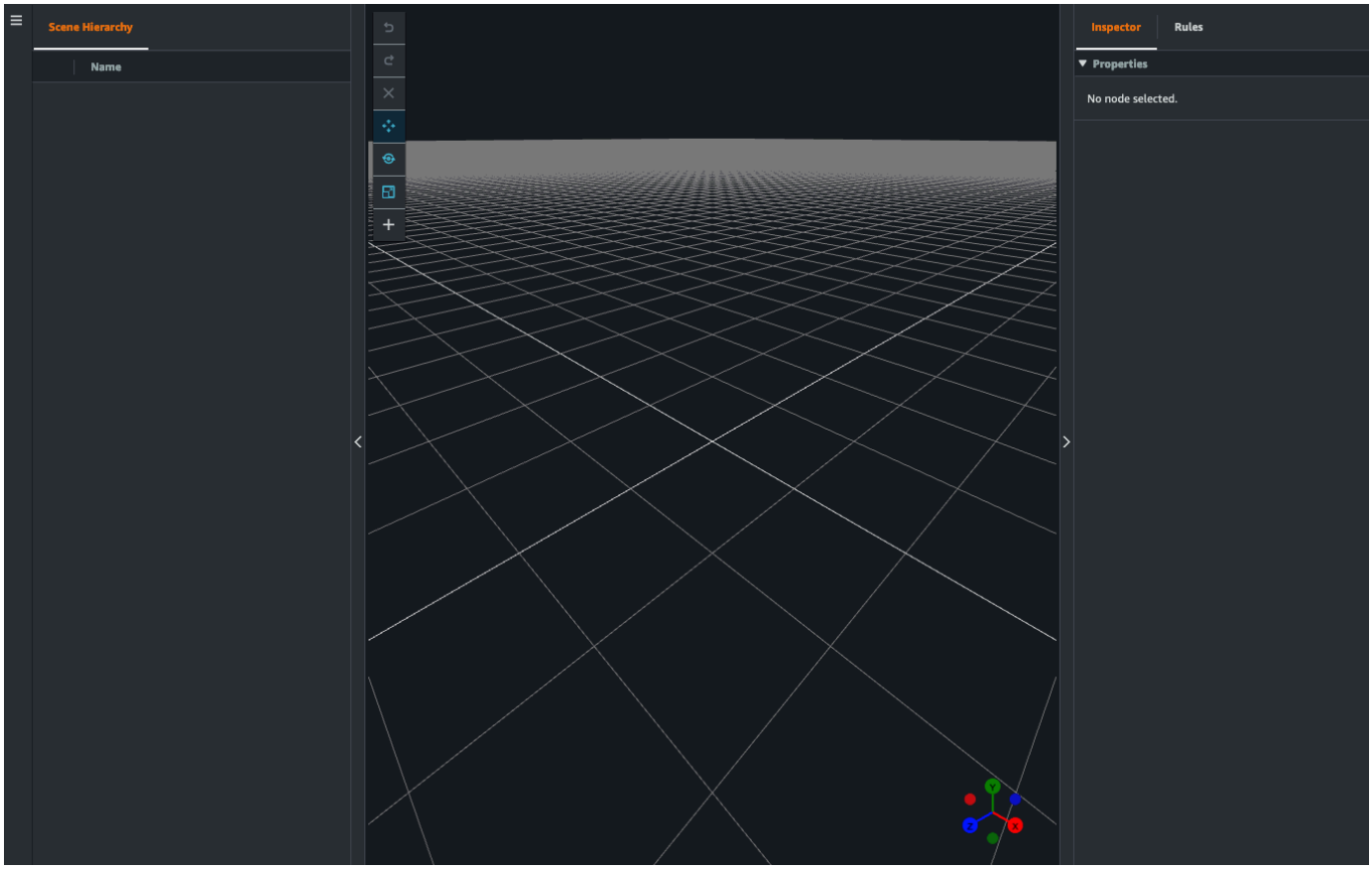
En esta sección, configurará una escena para poder editar su gemelo digital. Puedes importar un modelo 3D que hayas subido a la biblioteca de recursos y, a continuación, añadir widgets y vincular

los datos de las propiedades a los objetos para completar tu gemelo digital. Los objetos de la escena pueden incluir un edificio o espacio completo, o piezas individuales del equipo colocadas en su ubicación física.

> **Note**
>
> Antes de crear una escena, debe crear un espacio de trabajo.

Utilice el siguiente procedimiento para crear la escena en AWS IoT TwinMaker.

1. Para abrir el panel de escenas, en la barra de navegación izquierda de su espacio de trabajo, elija Escenas.
2. Elija Crear escena. Se abre el nuevo panel de creación de escenas.
3. En el panel de creación de escenas, introduzca un nombre y una descripción para la nueva escena. Si tienes un plan de precios estándar o por paquetes escalonados, puedes seleccionar el tipo de escena. Se recomienda utilizar una escena dinámica.
4. Cuando esté listo para crear la escena, elija Crear escena. La nueva escena se abre y está lista para que pueda trabajar con ella.

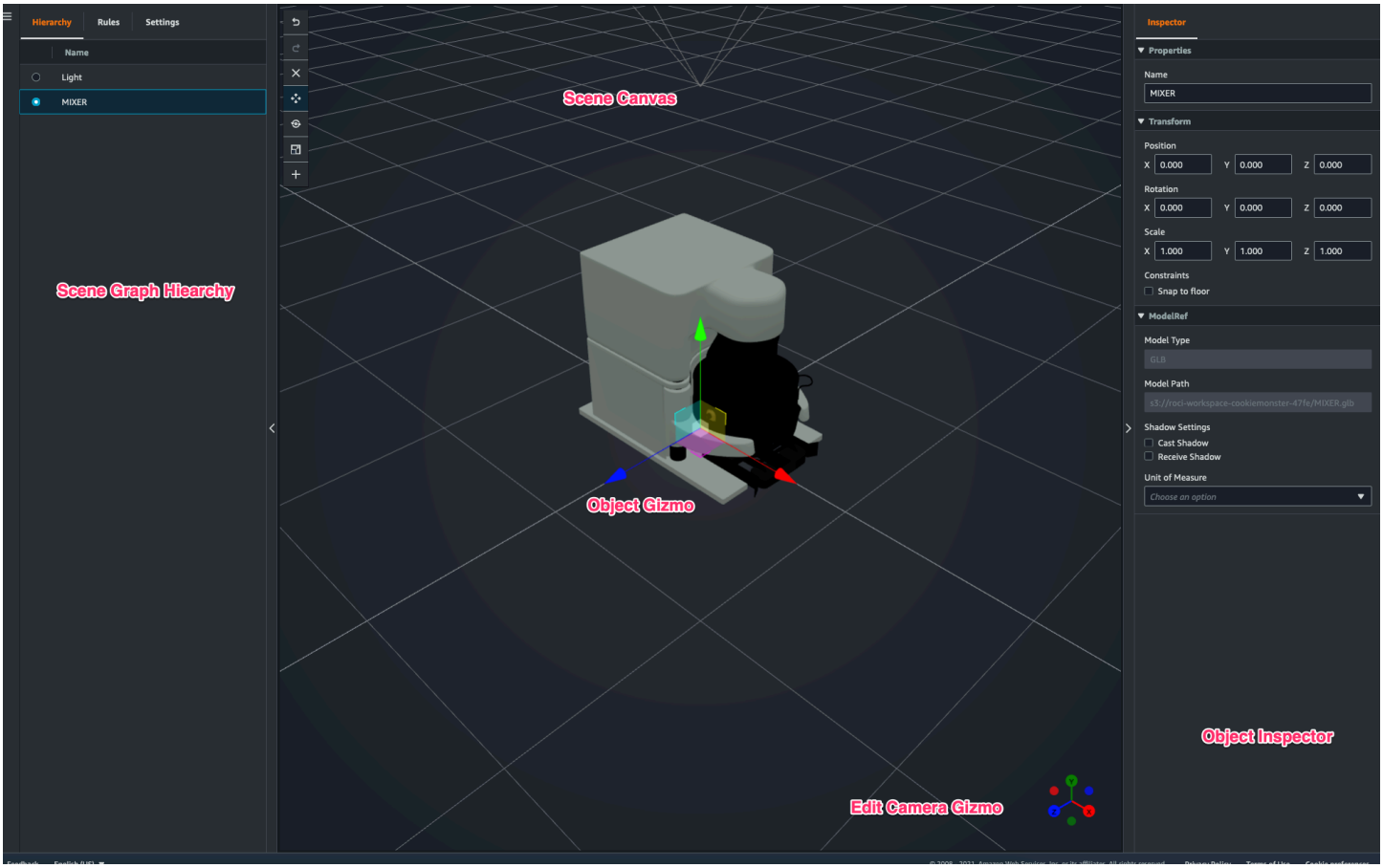


Usa la navegación 3D AWS IoT TwinMaker en tus escenas

La AWS IoT TwinMaker escena tiene un conjunto de controles de navegación que puede utilizar para navegar de forma eficiente por el espacio 3D de la escena. Para interactuar con el espacio 3D y los objetos representados por la escena, utilice los siguientes widgets y opciones del menú.

- Inspector: utilice la ventana Inspector para ver y editar las propiedades y los ajustes de una entidad o un componente seleccionados en la jerarquía.
- Lienzo de la escena: el lienzo de la escena es el espacio 3D en el que puede posicionar y orientar cualquier recurso 3D que desee utilizar.
- Jerarquía de gráficos de escenas: puede usar este panel para ver todas las entidades presentes en su escena. Aparece en el lado izquierdo de la ventana.
- Artilugio de objetos: use este artilugio para mover objetos por el lienzo. Aparece en el centro de un objeto 3D seleccionado en el Lienzo de escenas.

- Editar artilugio de la cámara: utilice el artilugio de edición de cámara para ver rápidamente la orientación actual de la cámara de visualización de escena y modificar el ángulo de visión. Puede encontrar este artilugio en la esquina inferior derecha de la vista de la escena.
- Controles de zoom: para navegar por el Lienzo de la escena, haga clic con el botón derecho y arrastre en la dirección en la que desee moverse. Para rotar, haga clic con el botón izquierdo y arrastre para girar. Para hacer zoom, use la rueda de desplazamiento del ratón o pellizque y separe los dedos en el panel táctil de su ordenador portátil.



Los botones de escena del panel jerárquico muestran las siguientes funciones, ordenadas según el diseño de los botones:

- Deshacer: deshace el último cambio que hizo en la escena.
- Rehacer: rehace el último cambio que hizo en la escena.
- Más (+): utilice este botón para acceder a las siguientes acciones: Agregar un nodo vacío, Agregar un modelo 3D, Agregar una etiqueta, Agregar luz y Agregar sombreador de modelos.

- Cambiar el método de navegación: acceda a las opciones de navegación de la cámara de la escena: Orbital y Panorámica.
- Papelera (eliminar): utilice este botón para eliminar un objeto seleccionado de la escena.
- Herramientas de manipulación de objetos: utilice este botón para trasladar, rotar y escalar el objeto seleccionado.

Agregue cámaras fijas a las entidades

Puede adjuntar vistas de cámara fijas a las entidades que estén dentro de AWS IoT TwinMaker las escenas. Estas cámaras proporcionan una perspectiva fija en un modelo 3D, lo que permite cambiar rápida y fácilmente la perspectiva de una escena a la entidad objetivo.

1. Navegue hasta su escena en la consola de AWS IoT TwinMaker.
2. En el menú de jerarquía de escenas, seleccione la entidad a la que desea conectar la cámara.
3. Pulse el botón + y, en las opciones desplegables, seleccione Agregar cámara desde la vista actual. Para aplicar una cámara con la perspectiva actual a la entidad, haga lo siguiente:
4. En el inspector, puede configurar la cámara y ajustar las siguientes opciones:
    - El nombre de la cámara
    - La posición y la rotación de la cámara
    - La distancia focal de la cámara
    - El nivel de zoom
    - Los planos de recorte de cerca y de lejos
5. Para acceder a la cámara después colocarla, haga lo siguiente: Seleccione la entidad a la que agregó la cámara en la jerarquía. Busque el nombre de la cámara que aparece debajo de la entidad.
6. Una vez que seleccione la cámara que se colocó en su entidad, la vista de la cámara de la escena se ajustará a la perspectiva establecida de la cámara colocada.

Edición de escenas mejoradas

AWS IoT TwinMaker las escenas incluyen un conjunto de herramientas para mejorar, editar y manipular los recursos presentes en la escena.

Los siguientes temas le enseñan cómo utilizar las funciones de edición mejoradas en sus AWS IoT TwinMaker escenas.

- Colocación dirigida de los objetos de la escena
- Selección de submodelos
- Edite las entidades en la jerarquía de escenas

Colocación dirigida de los objetos de la escena

AWS IoT TwinMaker le permite colocar y añadir objetos a la escena con precisión. Esta característica de edición mejorada proporciona un mayor control sobre la colocación de las etiquetas, las entidades, las luces y los modelos en la escena.

1. Navegue hasta su escena en la consola de AWS IoT TwinMaker.
2. Pulse el botón + y seleccione una de las opciones desplegables. Esta puede ser un modelo, una luz, una etiqueta o cualquier cosa del menú +.

   Cuando mueva el cursor en el espacio 3D de la escena, debería ver un objetivo alrededor del cursor.
3. Usa el objetivo para colocar con precisión los elementos en tu escena.

Selección de submodelos

AWS IoT TwinMaker permite seleccionar submodelos de modelos 3D en las escenas y aplicarles propiedades estándar, como etiquetas, luces o reglas.

Los formatos de archivo de los modelos 3D contienen metadatos que pueden especificar subáreas del modelo como submodelos dentro del modelo más grande. Por ejemplo, un modelo podría ser un sistema de filtración; las partes individuales del sistema, como tanques, tuberías o un motor, se marcan como submodelos del modelo 3D de la filtración.

Formatos de archivo 3D admitidos en las escenas: GLB y GLTF.

1. Navegue hasta su escena en la consola de AWS IoT TwinMaker.
2. Si no tiene modelos en la escena, asegúrese de agregar uno seleccionando la opción en el menú +.

3. Seleccione el modelo que aparece en la jerarquía de escenas. Una vez seleccionado, la jerarquía debería mostrar todos los submodelos que se encuentren debajo del modelo.

   > **Note**
   >
   > Si no ve ningún submodelo en la lista, es probable que el modelo no esté configurado para tener ningún submodelo.

4. Para cambiar la visibilidad de un submodelo, pulse el icono en forma de ojo, situado a la derecha del nombre del submodelo en la jerarquía.
5. Para editar los datos del submodelo, como el nombre o la posición, el inspector de escenas se abrirá cuando se seleccione un submodelo. Utilice el menú del inspector para actualizar o cambiar los datos del submodelo.
6. Para agregar etiquetas, luces, reglas u otras propiedades a los submodelos, pulse el signo + con el submodelo seleccionado en la jerarquía.

Edite las entidades en la jerarquía de escenas

AWS IoT TwinMaker las escenas permiten editar directamente las propiedades de las entidades de la tabla jerárquica. El siguiente procedimiento muestra las acciones que puede llevar a cabo en una entidad a través del menú jerárquico.

1. Navegue hasta su escena en la consola de AWS IoT TwinMaker.
2. Abra la jerarquía de escenas y seleccione un subelemento de la entidad que desee manipular.
3. Una vez seleccionado el elemento, pulse el botón + y, en el menú desplegable, seleccione una de las siguientes opciones:

   - Agregar un nodo vacío
   - Agregar un modelo 3D
   - Agregar luz
   - Agregar cámara desde la vista actual
   - Agregar etiqueta
   - Agregar sombreador de modelos
   - Agregar indicador de movimiento

4. Tras la selección de una de las opciones del menú desplegable, esta se aplicará a la escena como la selección secundaria del elemento seleccionado en el paso 2.
5. Puede reordenar los elementos secundarios y reorganizar los elementos jerárquicamente seleccionando un elemento secundario y arrastrándolo en la jerarquía hacia un nuevo elemento principal.

Agregue anotaciones a las entidades

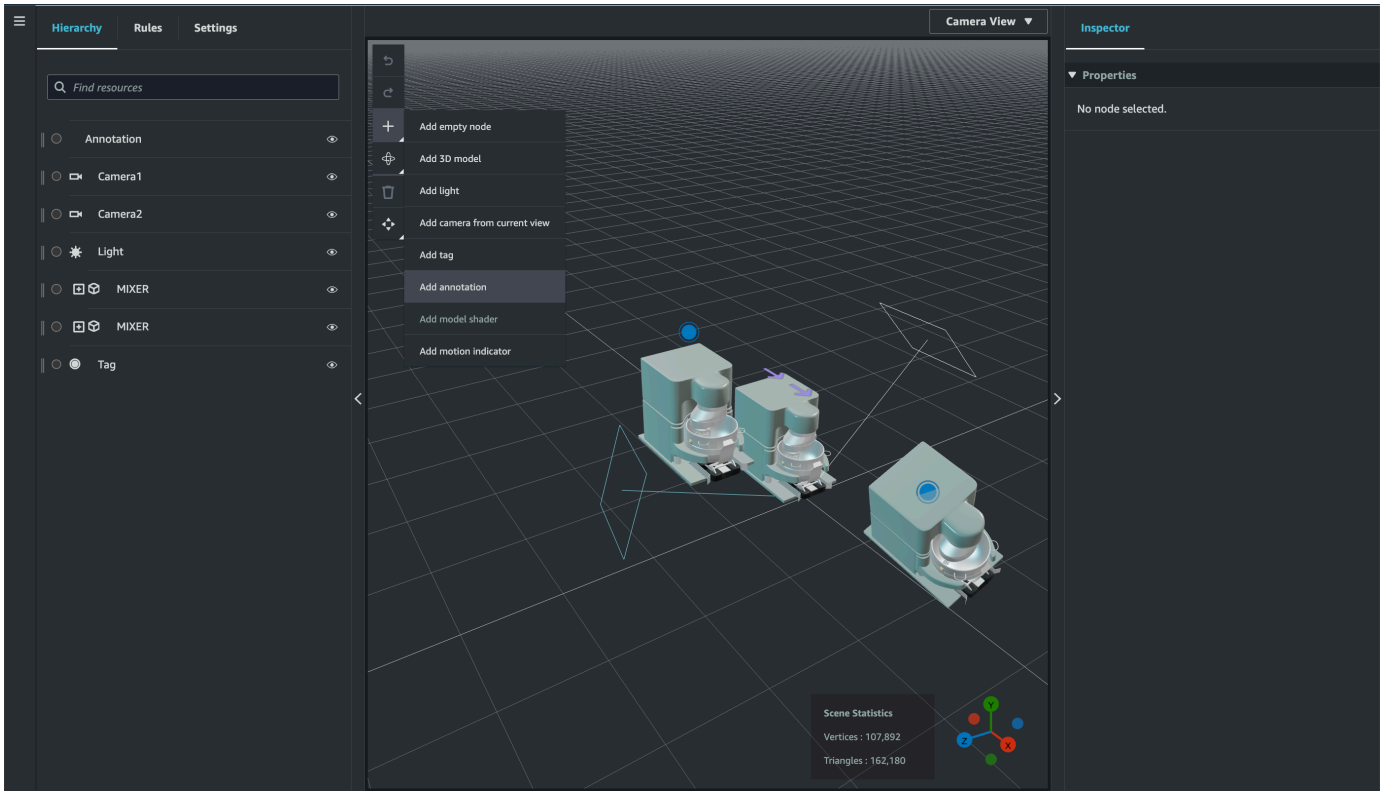
El compositor de AWS IoT TwinMaker escenas le permite anotar cualquier elemento de la jerarquía de escenas. La anotación se crea en Markdown.

Para obtener más información sobre cómo escribir en Markdown, consulte la documentación oficial sobre la sintaxis de Markdown, Syntaxis básica.

> **Note**
>
> AWS IoT TwinMaker las anotaciones y superponen únicamente la sintaxis de Markdown y no la de HTML.

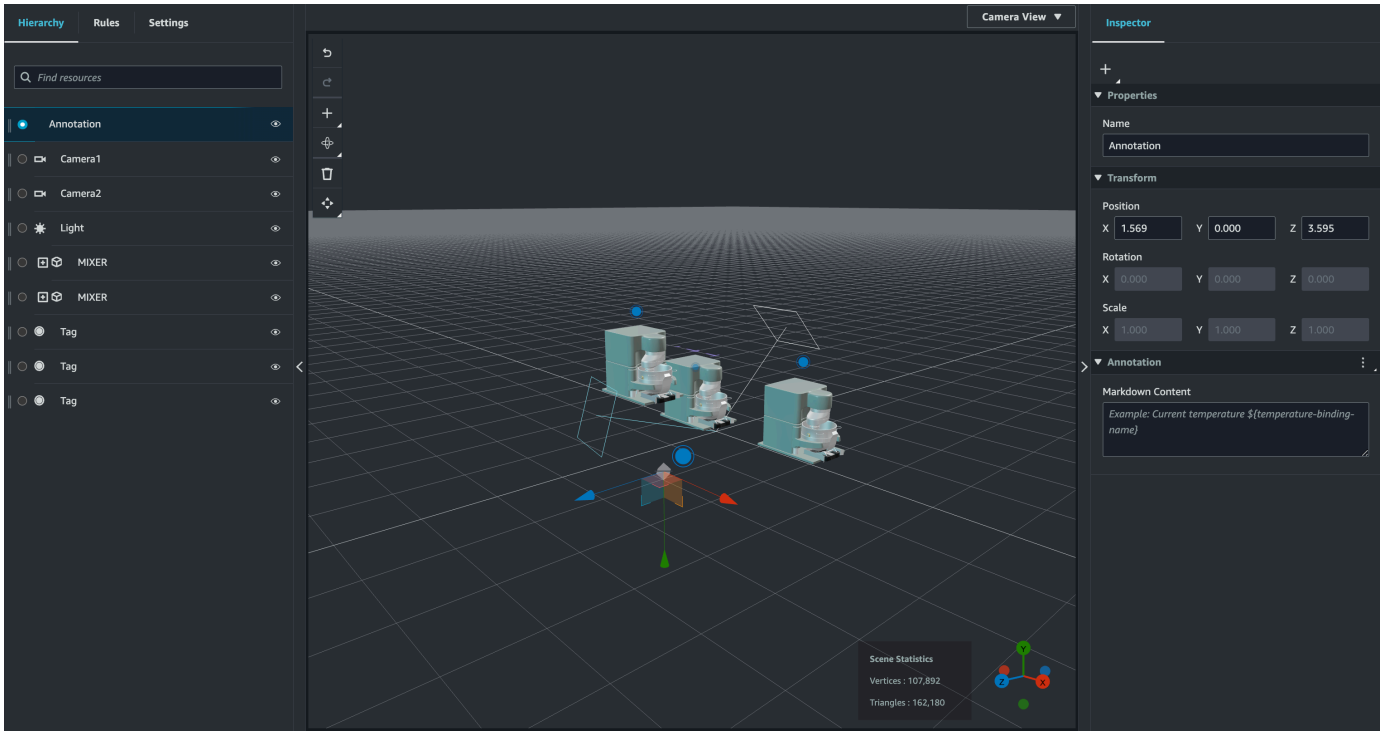
Agregar una anotación a una entidad

1. Navegue hasta su escena en la consola de AWS IoT TwinMaker.
2. Seleccione un elemento de la jerarquía de escenas que desee anotar. Si no se selecciona ningún elemento de la jerarquía, puede agregar una anotación a la raíz.
3. Pulse el botón más + y elija la opción Agregar anotación.

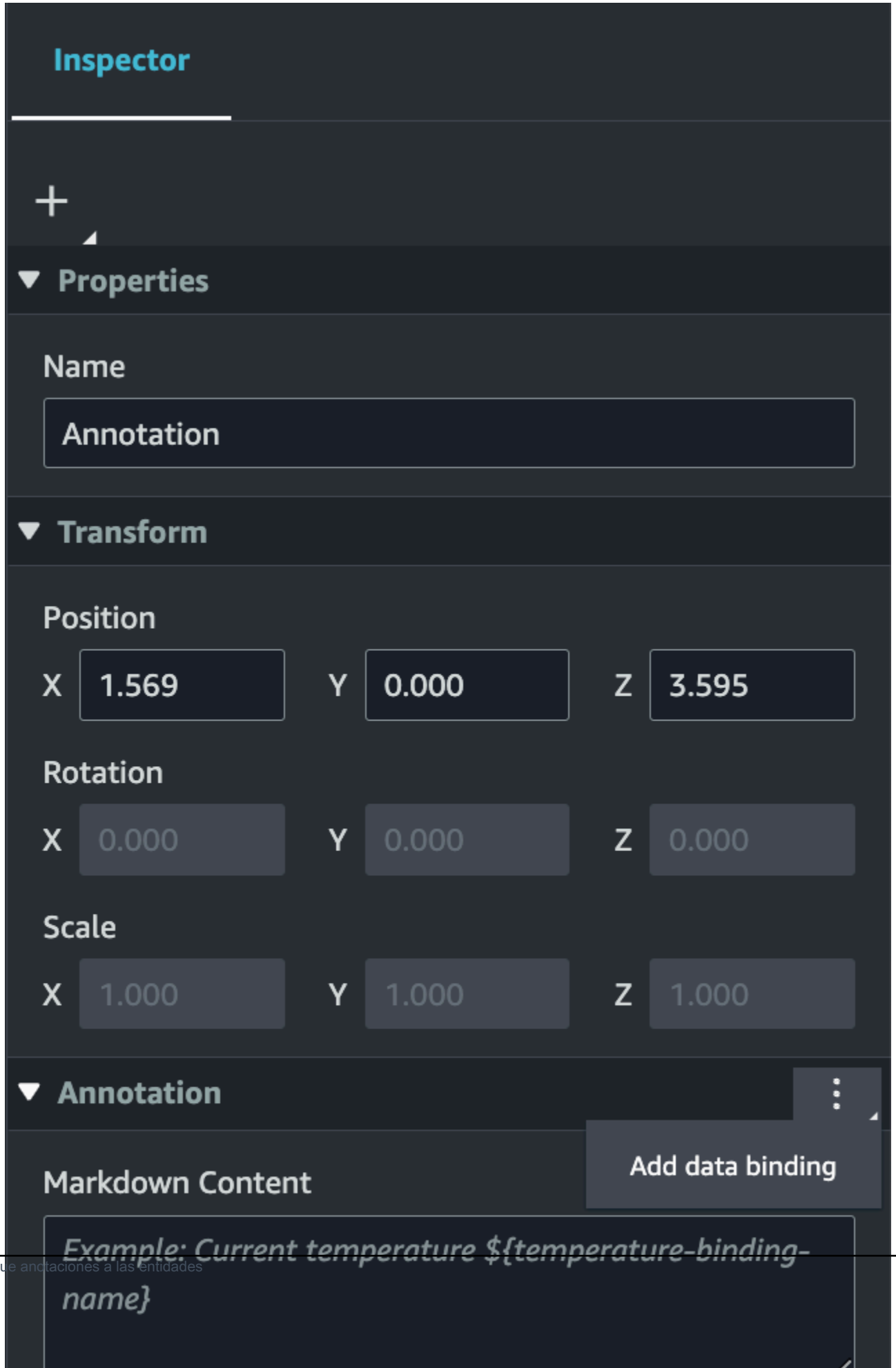


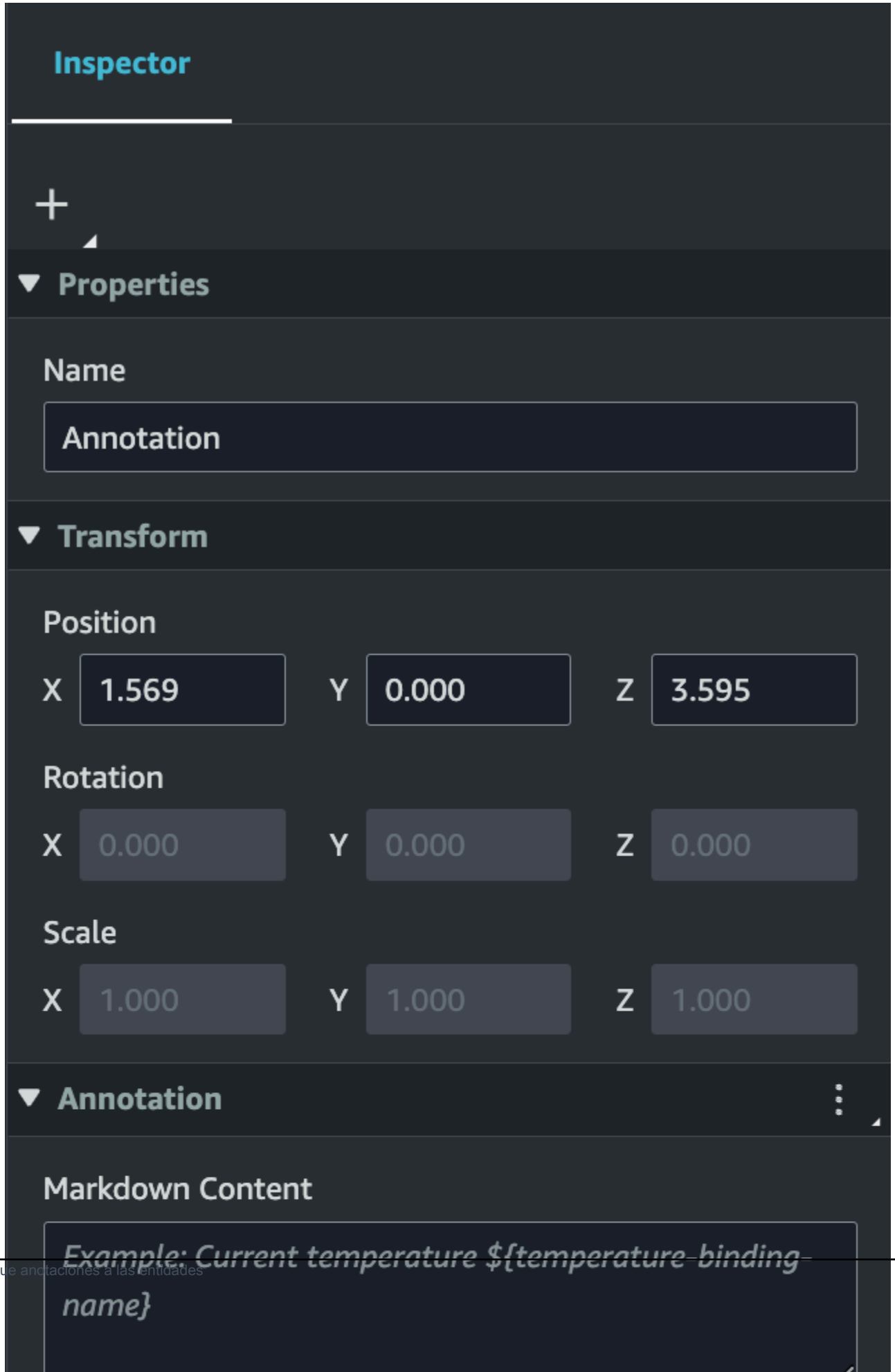
4. En la ventana del Inspector de la izquierda, desplácese hacia abajo hasta la sección de anotaciones. Con la sintaxis de Markdown, escriba el texto que desea que muestre la anotación.

   Para obtener más información sobre cómo escribir en Markdown, consulte la documentación oficial sobre la sintaxis de Markdown, Syntaxis básica.



5. Para vincular los datos de la AWS IoT TwinMaker escena a una anotación, seleccione Añadir enlace de datos, añada el ID de la entidad y, a continuación, seleccione el nombre del componente y el nombre de la propiedad de la entidad desde la que desee mostrar los datos. Puede actualizar el nombre del enlace para usarlo como una variable de Markdown y mostrar los datos en la anotación.

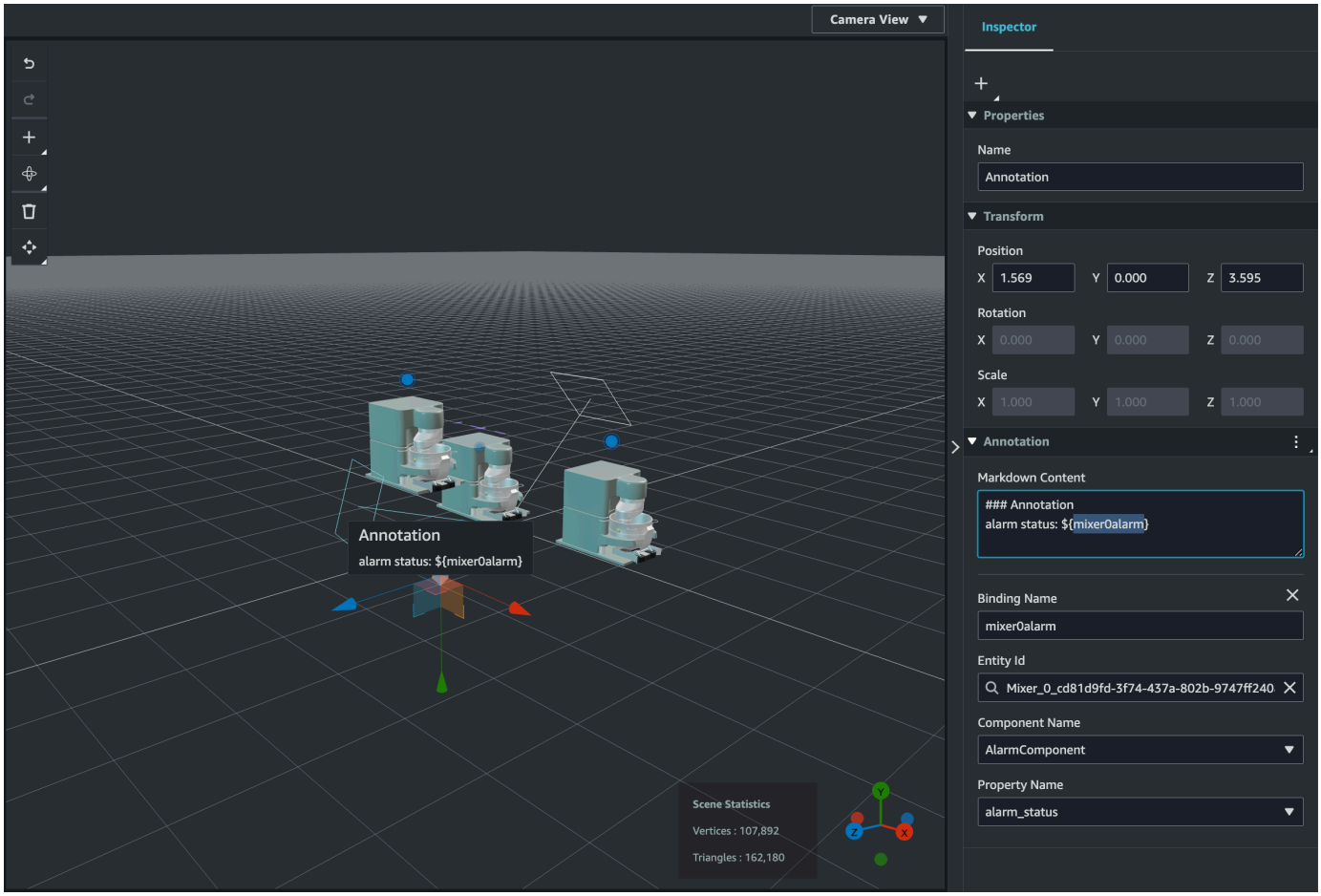

Inspector
Properties
Name
Annotation
Transform
Position
X 1.569
Y 0.000
Z 3.595
Rotation
X 0.000
Y 0.000
Z 0.000
Scale
X 1.000
Y 1.000
Z 1.000
Annotation
Add data binding
Markdown Content
Example: Current temperature ${temperature-binding-name}


Inspector
Properties
Name
Annotation
Transform
Position
X 1.569
Y 0.000
Z 3.595
Rotation
X 0.000
Y 0.000
Z 0.000
Scale
X 1.000
Y 1.000
Z 1.000
Annotation
Markdown Content
Example: Current temperature ${temperature-binding-name}

6. El nombre del enlace se utiliza para representar la variable de la anotación.

   Introduzca un nombre de enlace para mostrar el valor histórico más reciente de la serie temporal de una entidad en la anotación mediante la sintaxis AWS IoT TwinMaker de variables: `${variable-name}`

   A modo de ejemplo, esta superposición muestra el valor de `mixer0alarm`, en la anotación con la sintaxis `${mixer0alarm}`.



Agregue superposiciones a las Etiquetas

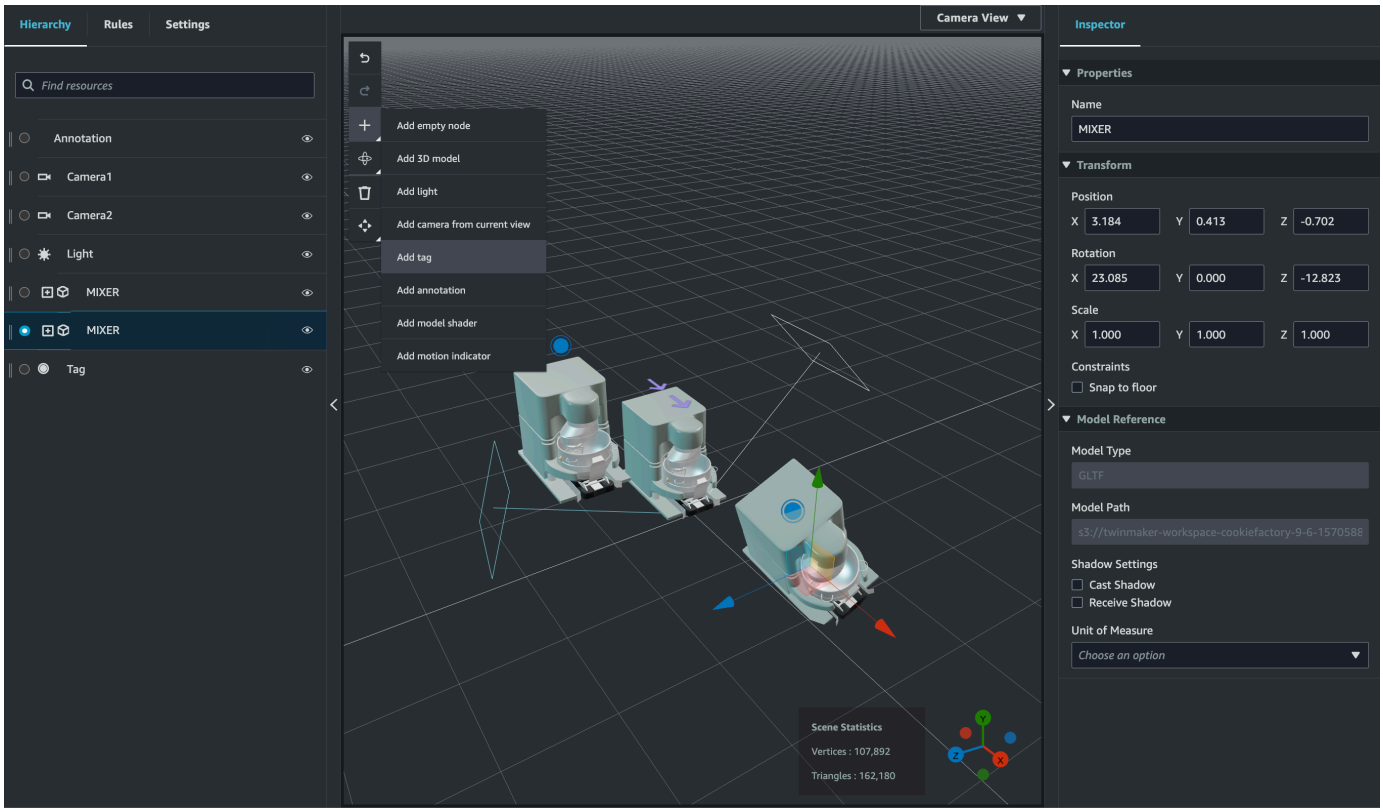
Puede crear superposiciones para sus escenas. AWS IoT TwinMaker Las superposiciones de escenas están asociadas a etiquetas y se pueden usar para mostrar datos críticos asociados a las entidades de la escena. La superposición se crea y se proporciona en Markdown.

Para obtener más información sobre cómo escribir en Markdown, consulte la documentación oficial sobre la sintaxis de Markdown, Syntaxis básica.

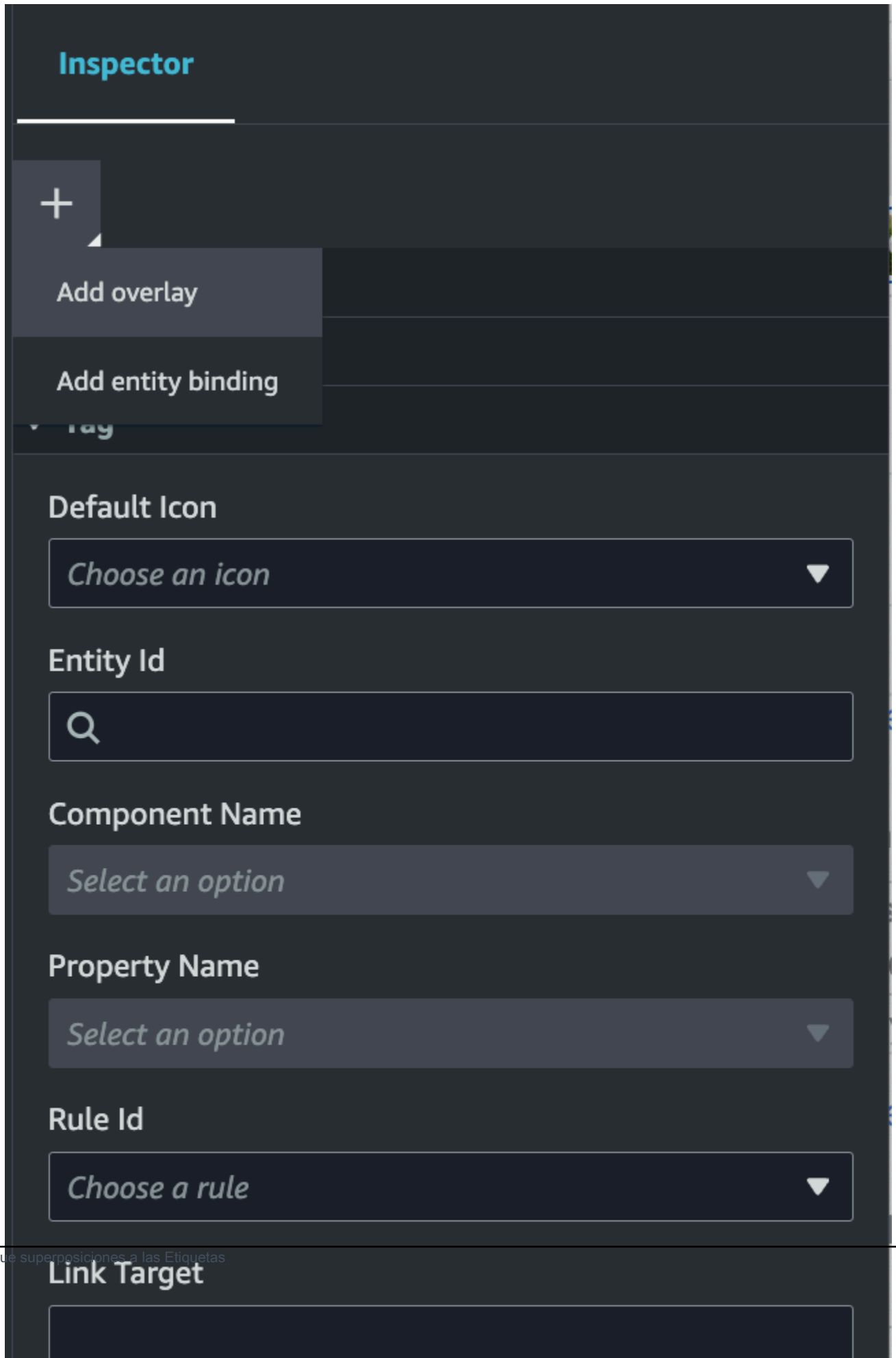
> **Note**
>
> De forma predeterminada, una superposición solo está visible en una escena cuando se selecciona la etiqueta asociada a ella. Puedes cambiar esta opción en los ajustes de la escena para que todas las superposiciones estén visibles a la vez.

1. Navegue hasta su escena en la consola de AWS IoT TwinMaker.
2. La AWS IoT TwinMaker superposición está asociada a una escena de etiquetas, puedes actualizar una etiqueta existente o añadir una nueva.

   Pulse el botón más + y elija la opción Agregar etiqueta.



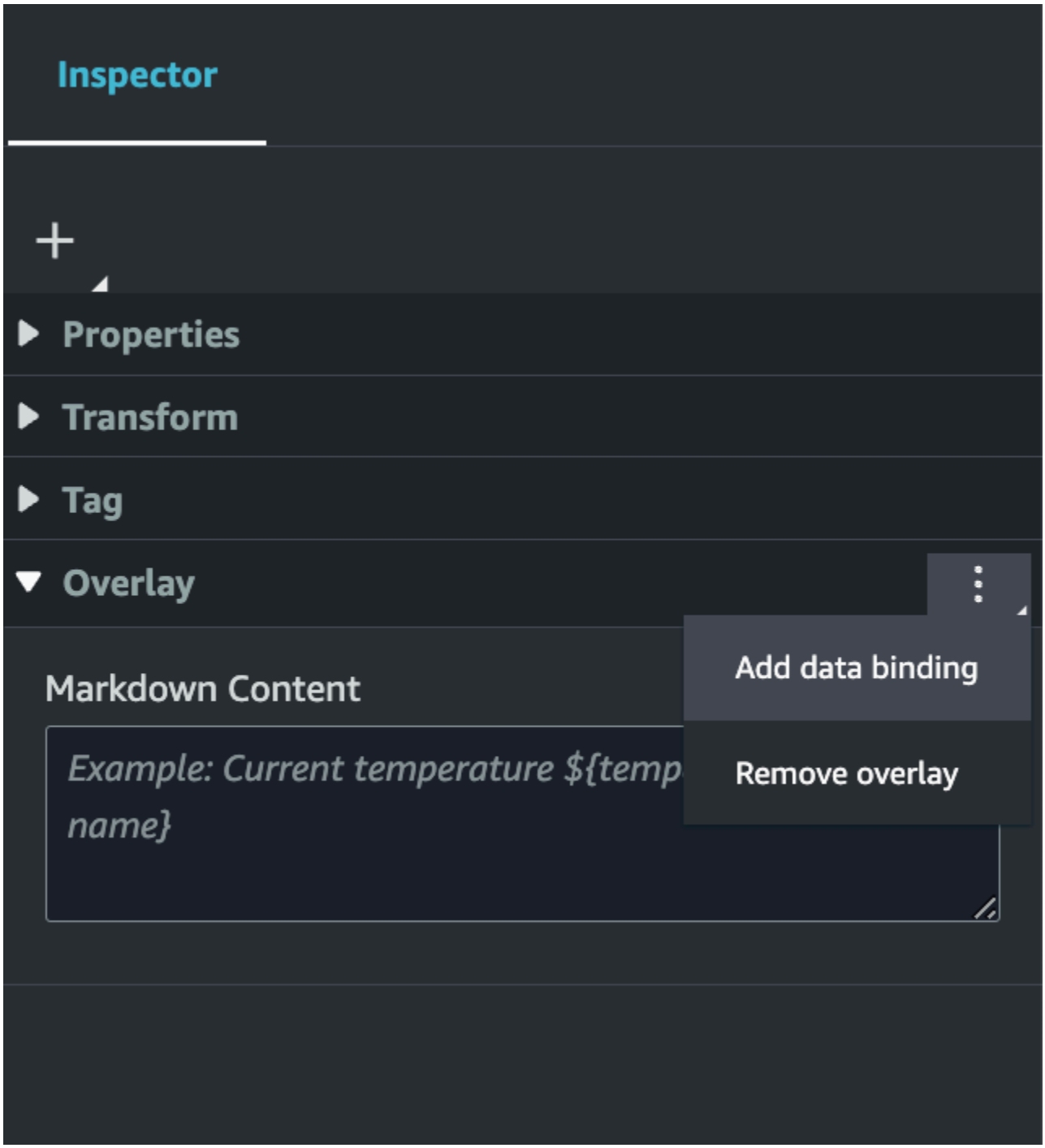
3. En el panel Inspector de la derecha, selecciona el botón + (símbolo más) y, a continuación, selecciona Añadir superposición.


Inspector
Add overlay
Add entity binding
Default Icon
Choose an icon
Entity Id
Component Name
Select an option
Property Name
Select an option
Rule Id
Choose a rule
Link Target

4. En la sintaxis de Markdown, escriba el texto que quiere que muestre la superposición.

   Para obtener más información sobre cómo escribir en Markdown, consulte la documentación oficial sobre la sintaxis de Markdown, Syntaxis básica.

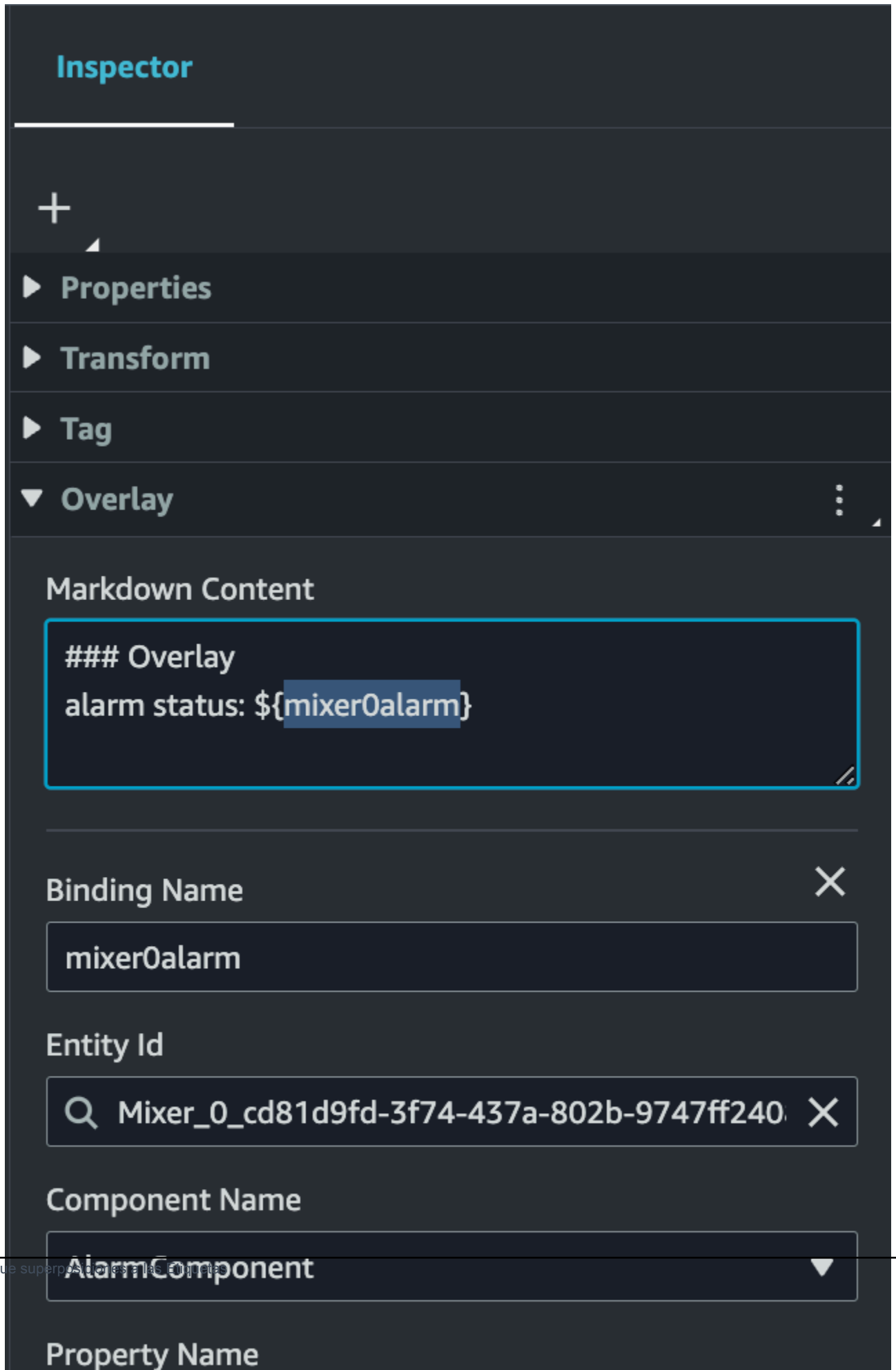
5. Para vincular los datos de AWS IoT TwinMaker la escena a una superposición, selecciona Añadir enlace de datos.



Añada el nombre del enlace y el identificador de la entidad y, a continuación, seleccione el nombre del componente y el nombre de la propiedad de la entidad de la que desee extraer los datos.

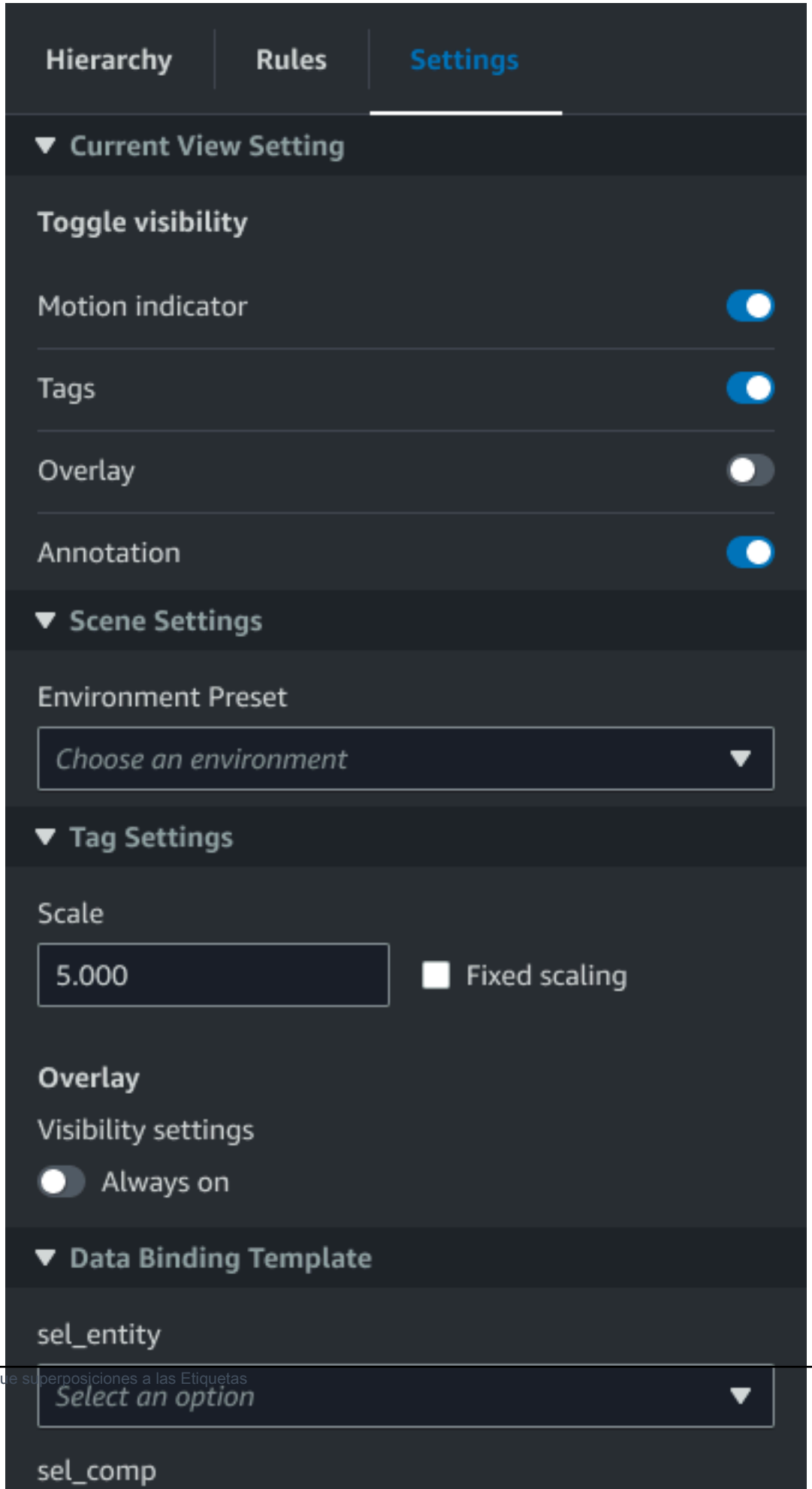
6. Puede mostrar el valor histórico más reciente de los datos de la serie temporal de una entidad en la superposición mediante AWS IoT TwinMaker la sintaxis variable:. `${variable-name}`

   Como ejemplo, esta superposición muestra el valor de `mixer0alarm`, en la superposición con la sintaxis `${mixer0alarm}`.


Inspector
Properties
Transform
Tag
Overlay
Markdown Content
Overlay
alarm status: ${mixer0alarm}
Binding Name
mixer0alarm
Entity Id
Mixer_0_cd81d9fd-3f74-437a-802b-9747ff240
Component Name
AlarmComponent
Property Name

7. Para activar la visibilidad de las superposiciones, abre la pestaña Configuración en la parte superior izquierda y asegúrate de que la opción Superposición esté activada para que todas las superposiciones estén visibles a la vez.

> **Note**
>
> De forma predeterminada, una superposición solo está visible en una escena cuando se selecciona la etiqueta asociada a ella.


Hierarchy
Rules
Settings
Current View Setting
Toggle visibility
Motion indicator
Tags
Overlay
Annotation
Scene Settings
Environment Preset
Choose an environment
Tag Settings
Scale
5.000
Fixed scaling
Overlay
Visibility settings
Always on
Data Binding Template
sel_entity
Select an option
sel_comp

Edite sus escenas

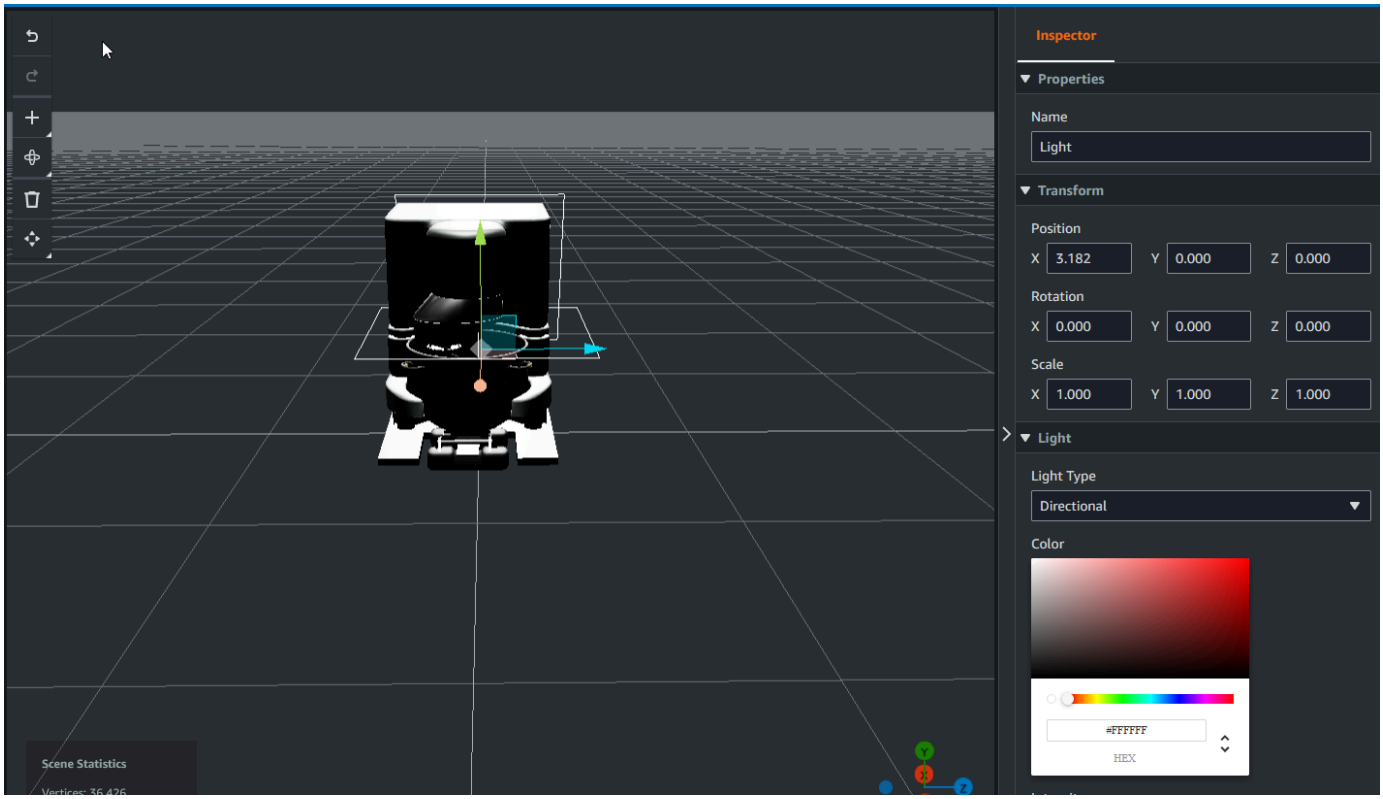
Una vez que haya creado una escena, puede agregar entidades y componentes y configurar widgets aumentados en la escena. Use componentes y widgets de entidades para modelar el gemelo digital y proporcionar una funcionalidad que se adapte a su caso de uso.

Agregue modelos a sus escenas

Para agregar modelos a su escena, use el siguiente procedimiento.

> **Note**
>
> Para agregar modelos a su escena, primero debe cargar los modelos en la biblioteca de recursos de AWS IoT TwinMaker . Para obtener más información, consulte Cargue los recursos a la biblioteca de AWS IoT TwinMaker recursos.

1. En la página del compositor de escenas, elija el signo más (+) y, a continuación, elija Agregar modelo 3D.
2. En la ventana Añadir un recurso desde la biblioteca de recursos, selecciona el CookieFactorMixerarchivo.glb y, a continuación, selecciona Añadir. Se abre el compositor de escenas.
3. Opcional: elija el signo más (+) y, a continuación, seleccione Agregar luz.
4. Elija cada opción de luz para ver cómo afectan a la escena.



> **Note**
>
> Las escenas tienen una iluminación ambiental predeterminada. Para evitar la pérdida de velocidad de fotogramas, considere limitar la cantidad de luces adicionales que se colocan en la escena.

Agrega widgets de interfaz de usuario aumentados con modelador y sombreador a tu escena

Los widgets de sombreado de modelos pueden cambiar el color de un objeto en las condiciones que tú definas. Por ejemplo, puede crear un widget de color que cambie el color de un mezclador de cookies de la escena en función de los datos de temperatura del mezclador.

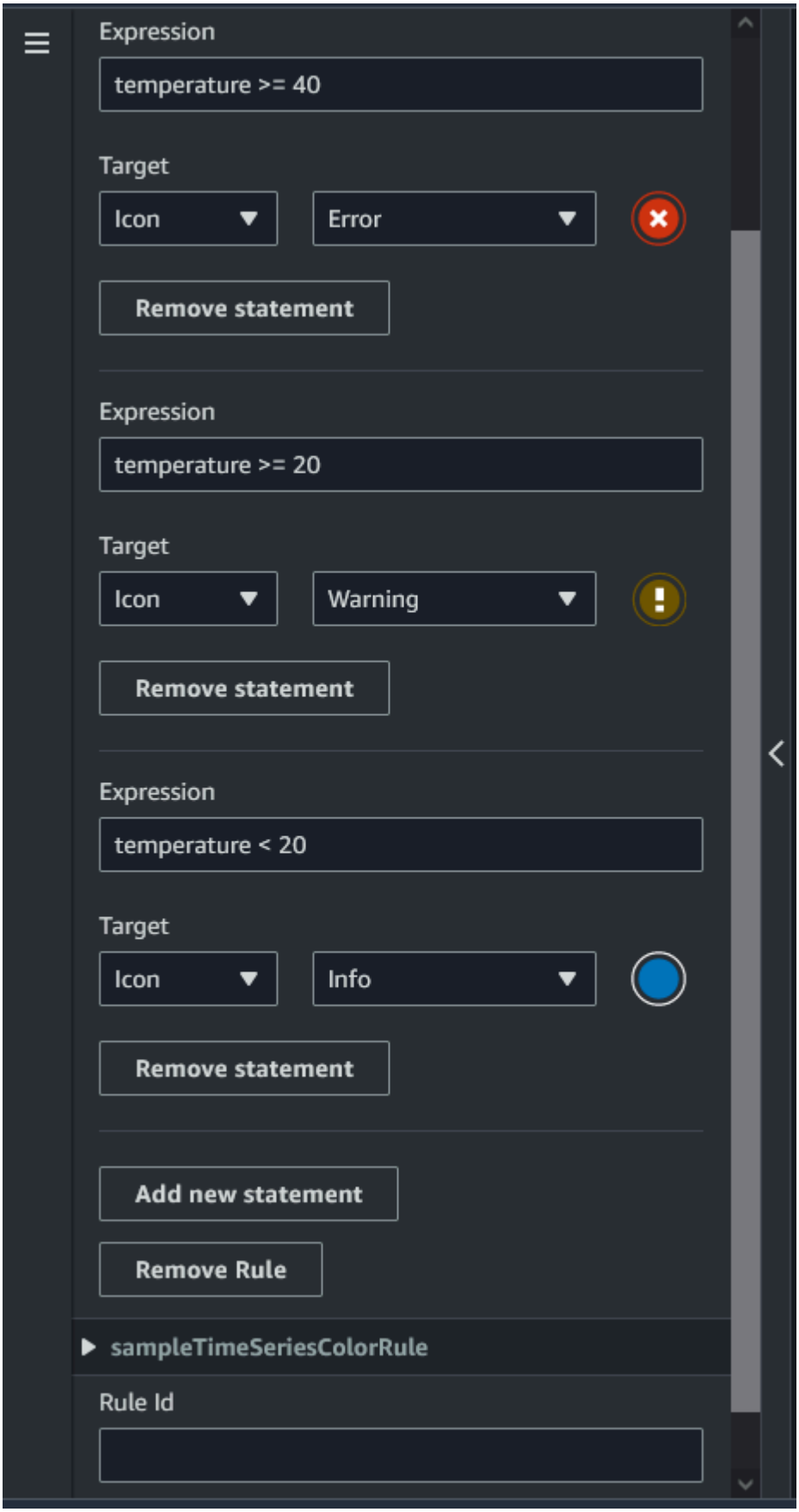
Utilice el siguiente procedimiento para añadir widgets de sombreado de modelos a un objeto seleccionado.

1. Seleccione un objeto de la jerarquía al que desee agregar un widget. Pulse el botón + y, a continuación, elija Model Shader.

2. Para añadir un nuevo grupo de reglas visual, primero siga las instrucciones que aparecen a continuación para crear el y ColorRule, a continuación, en el panel del Inspector para el objeto del ID de regla, elija ColorRule.
3. Seleccione el ID de entidad al PropertyName que desee vincular el sombreador del modelo. ComponentName

Cree reglas visuales para sus escenas

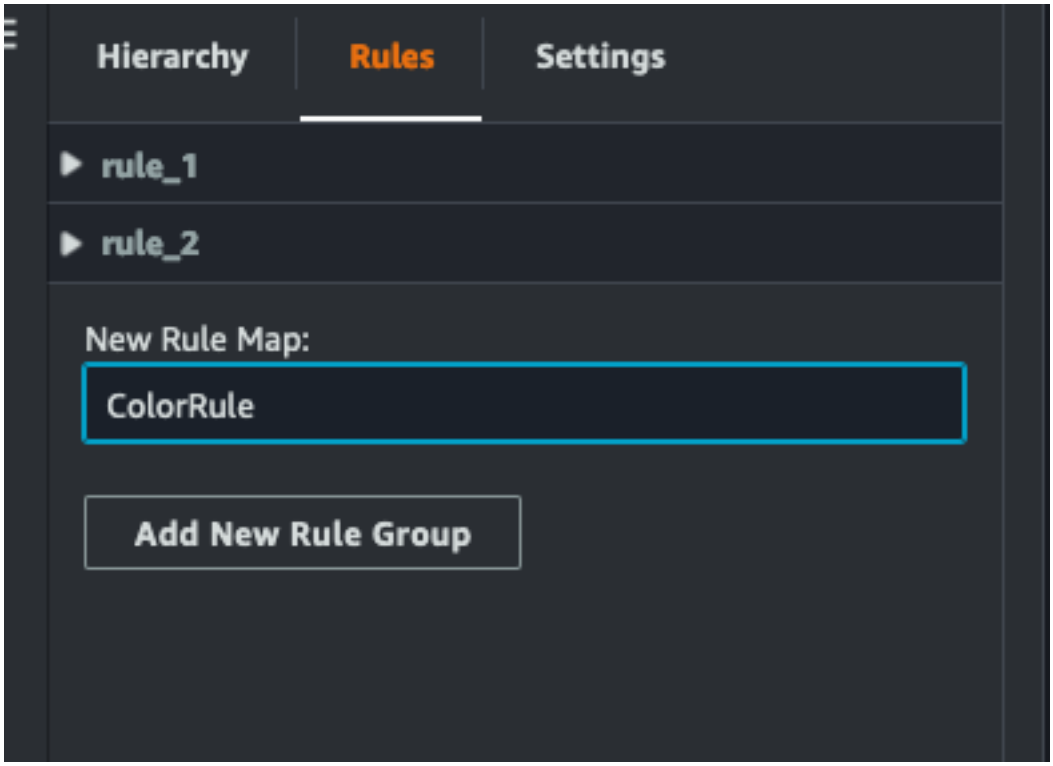
Puede utilizar mapas de reglas visuales para especificar las condiciones basadas en los datos que cambian la apariencia visual de un widget de interfaz de usuario aumentada, como una etiqueta o un sombreador de modelos. Se proporcionan ejemplos de reglas, pero también puede crear las suyas propias. El siguiente ejemplo muestra una regla visual.


Expression
temperature >= 40
Target
Icon
Error
Remove statement
Expression
temperature >= 20
Target
Icon
Warning
Remove statement
Expression
temperature < 20
Target
Icon
Info
Remove statement
Add new statement
Remove Rule
sampleTimeSeriesColorRule
Rule Id

La imagen de arriba muestra una regla para comparar una propiedad de datos previamente definida con el identificador «temperatura» con un valor determinado. Por ejemplo, si la «temperatura» es mayor o igual a 40, el estado cambiará el aspecto de la etiqueta a un círculo rojo. Cuando se elige el objetivo en el panel de control de Grafana, se rellena un panel de detalles que está configurado para usar el mismo origen de datos.

En el procedimiento a continuación se muestra cómo agregar un nuevo grupo de reglas visuales para la capa de interfaz de usuario aumentada con coloración de la malla.

1. En la pestaña de reglas de la consola, introduce un nombre como el que aparece ColorRule en el campo de texto y selecciona Añadir nuevo grupo de reglas.



2. Defina una nueva regla para el caso de uso. Por ejemplo, puede crear uno basado en la propiedad de datos «temperatura», donde el valor indicado es inferior a 20. Use la siguiente sintaxis para las expresiones de reglas: Menor que es <, mayor que es >, menor o igual es <=, mayor que o igual es >= e igual es ==. (Para obtener más información, consulte la sintaxis JEXL de Apache Commons).
3. Defina el objetivo en un color. Para definir un color, por ejemplo`#fcba03`, utilice valores hexadecimales. (Para obtener más información sobre los valores hexadecimales, consulte Hexadecimal).

Creación de etiquetas para sus escenas

Una etiqueta es una anotación que se agrega a una posición de coordenadas `x,y,z` específica de una escena. La etiqueta usa una propiedad de la entidad para conectar una parte de la escena al gráfico de conocimiento. Puede usar una etiqueta para configurar el comportamiento o la apariencia visual de un elemento de la escena, como una alarma.

> **Note**
>
> Para agregar funcionalidad a las etiquetas, debe aplicar las reglas visuales a estas.

Utilice el siguiente procedimiento para agregar etiquetas a la escena.

1. Seleccione un objeto de la jerarquía, elija el botón + y, a continuación, elija Agregar etiqueta.
2. Nombre la etiqueta. A continuación, para aplicar una regla visual, seleccione un identificador de grupo visual.
3. En las listas desplegables, elija ComponentName EntityID, y. PropertyName
4. Para rellenar el campo Ruta de datos, elija Crear. DataFrameLabel

Formato de modelo de teselas 3D

Uso de mosaicos 3D en la escena

Si experimenta largos tiempos de espera al cargar escenas 3D AWS IoT TwinMaker o tiene un rendimiento de renderizado deficiente cuando navega por un modelo 3D complejo, puede que desee convertir sus modelos en mosaicos 3D. En esta sección se describe el formato de teselas 3D y las herramientas de terceros disponibles. Siga leyendo para decidir si los mosaicos 3D son adecuados para su caso de uso y si necesita ayuda para empezar.

Caso de uso de un modelo complejo

Un modelo 3D en la AWS IoT TwinMaker escena puede provocar problemas de rendimiento, como tiempos de carga lentos y retrasos en la navegación, si el modelo es:

- Grande: el tamaño del archivo es superior a 100 MB.
- Denso: está formado por cientos o miles de mallas distintas.

- Complejo: la geometría de malla tiene millones de triángulos para formar formas complejas.

Formato de mosaicos 3D

El formato 3D Tiles es una solución para transmitir la geometría del modelo y mejorar el rendimiento de la representación 3D. Permite la carga instantánea de modelos 3D en una AWS IoT TwinMaker escena y optimiza las interacciones 3D al cargar fragmentos de un modelo en función de lo que se ve en la vista de la cámara.

El formato 3D Tiles fue creado por Cesium. Cesium tiene un servicio gestionado para convertir modelos 3D en mosaicos 3D llamado Cesium Ion. Actualmente, esta es la mejor solución para crear mosaicos 3D, y la recomendamos para sus modelos complejos en los formatos compatibles. Puede registrar Cesium y elegir el plan de suscripción adecuado según los requisitos de su empresa en la página de precios de Cesium.

Para preparar un modelo de mosaicos 3D que puedas añadir a una AWS IoT TwinMaker escena, sigue las instrucciones documentadas por Cesium Ion:

- Importa un modelo a Cesium Ion

Cargue los mosaicos 3D de Cesium a AWS

Una vez que el modelo se haya convertido en mosaicos 3D, descargue los archivos del modelo y cárguelos en el bucket de Amazon S3 de su AWS IoT TwinMaker espacio de trabajo:

1. Cree y descargue su archivo de modelos de 3D Tiles.
2. Descomprime el archivo en una carpeta.
3. Cargue toda la carpeta 3D Tiles en el depósito de Amazon S3 asociado a su AWS IoT TwinMaker espacio de trabajo. (Consulte Carga de objetos en la Guía del usuario de Amazon S3).
4. Si su modelo de 3D Tiles se ha cargado correctamente, verá una ruta de carpeta de Amazon S3 en su biblioteca de AWS IoT TwinMaker recursos con el tipo`Tiles3D`.

> **Note**
>
> La biblioteca AWS IoT TwinMaker de recursos no admite la carga directa de modelos de 3D Tiles.

Uso de mosaicos 3D en AWS IoT TwinMaker

AWS IoT TwinMaker tiene conocimiento de cualquier modelo de teselas 3D cargado en el depósito S3 de su espacio de trabajo. El modelo debe tener todos los archivos dependientes (.gltf, .b3dm, .i3dm, .cmpt, .pnts) disponibles en el mismo directorio de Amazon S3. `tileset.json` La ruta del directorio de Amazon S3 aparecerá en la biblioteca de recursos con el tipo`Tiles3D`.

Para añadir el modelo de teselas 3D a la escena, siga estos pasos:

1. En la página del compositor de escenas, elija el signo más (+) y, a continuación, elija Agregar modelo 3D.
2. En la ventana Añadir un recurso de la biblioteca de recursos, elija la ruta a su modelo de teselas 3D con el tipo y`Tiles3D`, a continuación, seleccione Añadir.
3. Haz clic en el lienzo para colocar el modelo en tu escena.

Diferencias de 3D Tiles

Actualmente, 3D Tiles no admite metadatos geométricos y semánticos, lo que significa que la jerarquía de malla del modelo original no está disponible para la función de selección de submodelos. Aún puedes añadir widgets a tu modelo de teselas 3D, pero no puedes usar funciones ajustadas a los submodelos: el sombreador de modelos, las transformaciones 3D separadas o la vinculación de entidades para una malla de submodelos.

Se recomienda utilizar la conversión de mosaicos 3D para activos de gran tamaño que sirvan de contexto para el fondo de una escena. Si quieres desglosar y anotar con más detalle un submodelo, debes extraerlo como un glTF/glb recurso independiente y añadirlo directamente a la escena. Esto se puede hacer con herramientas 3D comunes y gratuitas, como Blender.

Ejemplo de caso de uso:

- Tiene un modelo de fábrica de 1 GB con salas y pisos de máquinas detallados, cajas eléctricas y tuberías de plomería. Las cajas y tuberías eléctricas deben brillar en rojo cuando los datos de propiedad asociados cruzan un umbral.

- Se aíslan las mallas de cajas y tubos del modelo y se exportan a un glTF independiente con Blender.
- Convierte la fábrica sin elementos eléctricos ni de fontanería en un modelo 3D Tiles y lo carga en S3.
- Añades tanto el modelo 3D Tiles como el modelo glTF a una AWS IoT TwinMaker escena en el origen (0,0,0).
- Se añaden componentes de sombreado de modelos a los submodelos de cajas eléctricas y tuberías del glTF para hacer que las mallas se pongan rojas según las reglas de propiedad.

Escenas dinámicas

AWS IoT TwinMaker las escenas aprovechan el potencial del gráfico de conocimiento al almacenar los nodos y los ajustes de la escena en un componente de entidad. Utilice la AWS IoT TwinMaker consola para crear escenas dinámicas y gestionar, crear y renderizar escenas 3D con mayor facilidad.

Características principales:

- Todos los objetos, ajustes y enlaces de datos de los nodos de escenas 3D se representan de forma «dinámica» en función de las consultas de los gráficos de conocimiento.
- Si utilizas el Visor de escenas de solo lectura en una aplicación Grafana o personalizada, puedes obtener actualizaciones de tus escenas en un intervalo de 30 segundos.

Escenas estáticas versus dinámicas

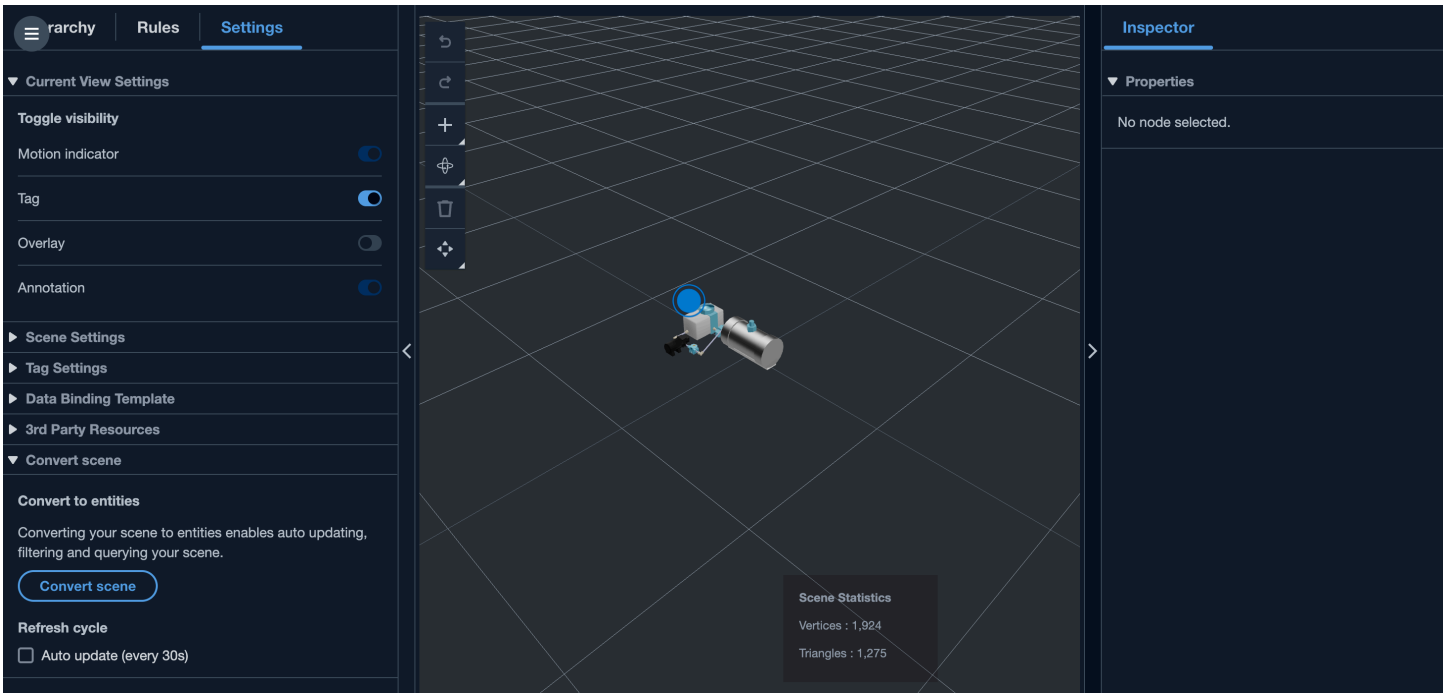
Las escenas estáticas se componen de un archivo JSON de escenas almacenado en S3 que contiene detalles de todos los nodos y ajustes de la escena. Cualquier cambio en la escena debe realizarse en el documento JSON y guardarse en S3. Una escena estática es la única opción si tienes un plan de precios básico.

Las escenas dinámicas se componen de un archivo JSON de escena que contiene la configuración global de la escena, mientras que todos los demás nodos de la escena y la configuración de los nodos se almacenan como componentes de la entidad en el gráfico de conocimiento. Las escenas dinámicas solo se admiten en los planes de precios estándar y por paquetes escalonados. Consulte Cambiar AWS IoT TwinMaker los modos de precios para obtener información sobre cómo actualizar su plan de precios).

Puede convertir una escena estática existente en una escena dinámica siguiendo estos pasos:

- Navegue hasta su escena en la consola de AWS IoT TwinMaker.
- En el panel de la izquierda, haz clic en la pestaña Configuración.
- Expande la sección Convertir escena en la parte inferior del panel.
- Haga clic en el botón Convertir escena y, a continuación, en Confirmar.

> **Warning**
>
> La conversión de una escena estática a una dinámica es irreversible.



Tipos y entidades de componentes de la escena

Para crear componentes de entidades específicos de la escena, se admiten los siguientes tipos de componentes 1P:

- com.amazon.iottwinmaker.3d.component.camera Tipo de componente que almacena la configuración de un widget de cámara.
- com.amazon.iottwinmaker.3d.component.dataoverlay Tipo de componente que almacena la configuración de una superposición de un widget de anotación o etiqueta.

- com.amazon.iottwinmaker.3d.component.light Tipo de componente que almacena la configuración de un widget de iluminación.
- com.amazon.iottwinmaker.3d.component.modelref Tipo de componente que almacena la configuración y la ubicación en S3 de un modelo 3D utilizado en una escena.
- com.amazon.iottwinmaker.3d.component.modelshader Tipo de componente que almacena la configuración de un sombreador de modelos en un modelo 3D.
- com.amazon.iottwinmaker.3d.component.motionindicator Tipo de componente que almacena la configuración de un widget indicador de movimiento.
- com.amazon.iottwinmaker.3d.component.submodelref Tipo de componente que almacena la configuración de un submodelo de un modelo 3D.
- com.amazon.iottwinmaker.3d.component.tag Tipo de componente que almacena la configuración de un widget de etiquetas.
- com.amazon.iottwinmaker.3d.node Tipo de componente que almacena la configuración básica de un nodo de escena, como su transformación 3D, su nombre y sus propiedades genéricas.

Conceptos de escena dinámica

Las entidades de escena dinámicas se almacenan en una entidad global etiquetada`$SCENES`. Cada escena se compone de una entidad raíz y una jerarquía de entidades secundarias que coinciden con la jerarquía de nodos de la escena. Cada nodo de escena situado debajo de la raíz tiene un componente com.amazon.iottwinmaker.3d.node y un componente para el tipo de nodo (modelo 3D, widget, etc.).

> **Warning**
>
> No elimines manualmente ninguna entidad de la escena o la escena podría estar rota. Si desea eliminar parcial o totalmente una escena, utilice la página del compositor de escenas para añadir y eliminar nodos de escena, y utilice la página de escenas para seleccionar y eliminar una escena.

Cree una aplicación web personalizada mediante componentes de AWS IoT TwinMaker interfaz de usuario

AWS IoT TwinMaker proporciona componentes de interfaz de usuario de código abierto para desarrolladores AWS IoT de aplicaciones. Con esos componentes de la interfaz de usuario, los desarrolladores pueden crear aplicaciones web personalizadas con AWS IoT TwinMaker funciones habilitadas para sus gemelos digitales.

AWS IoT TwinMaker Los componentes de la interfaz de usuario forman parte del AWS IoT Application Kit, una biblioteca de código abierto del lado del cliente que permite a los desarrolladores de aplicaciones de IoT simplificar el desarrollo de aplicaciones de IoT complejas

AWS IoT TwinMaker Los componentes de la interfaz de usuario incluyen:

- AWS IoT TwinMaker fuente:

  Un componente de conector de datos que le permite recuperar datos e interactuar con sus AWS IoT TwinMaker datos y sus gemelos digitales.

  Para obtener más información, consulte la documentación de origen de AWS IoT TwinMaker.
- Visor de escenas:

  Un componente de renderizado 3D creado sobre `@react-three/fiber` que renderiza su gemelo digital y le permite interactuar con él.

  Para obtener más información, consulte la documentación de Visor de escenas.
- Reproductor de video:

  Un componente de reproductor de vídeo que le permite transmitir un vídeo desde Kinesis Video Streams AWS IoT TwinMaker a través de él.

  Para obtener más información, consulte la documentación de Reproductor de vídeo.

Para obtener más información sobre el uso del kit de AWS IoT aplicaciones, visite la página de Github del kit de AWS IoT aplicaciones.

Para obtener instrucciones sobre cómo iniciar una nueva aplicación web con AWS IoT Application Kit, visite la página de documentación oficial del IoT App Kit.

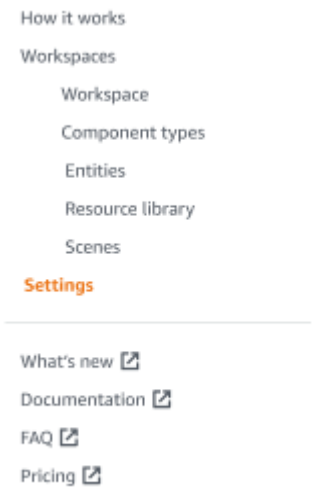
Cambiar AWS IoT TwinMaker los modos de precios

AWS IoT TwinMaker actualmente tiene tres modos de precios: paquete básico, estándar o escalonado. El modo de precios estándar se establece como el modo de precios predeterminado.

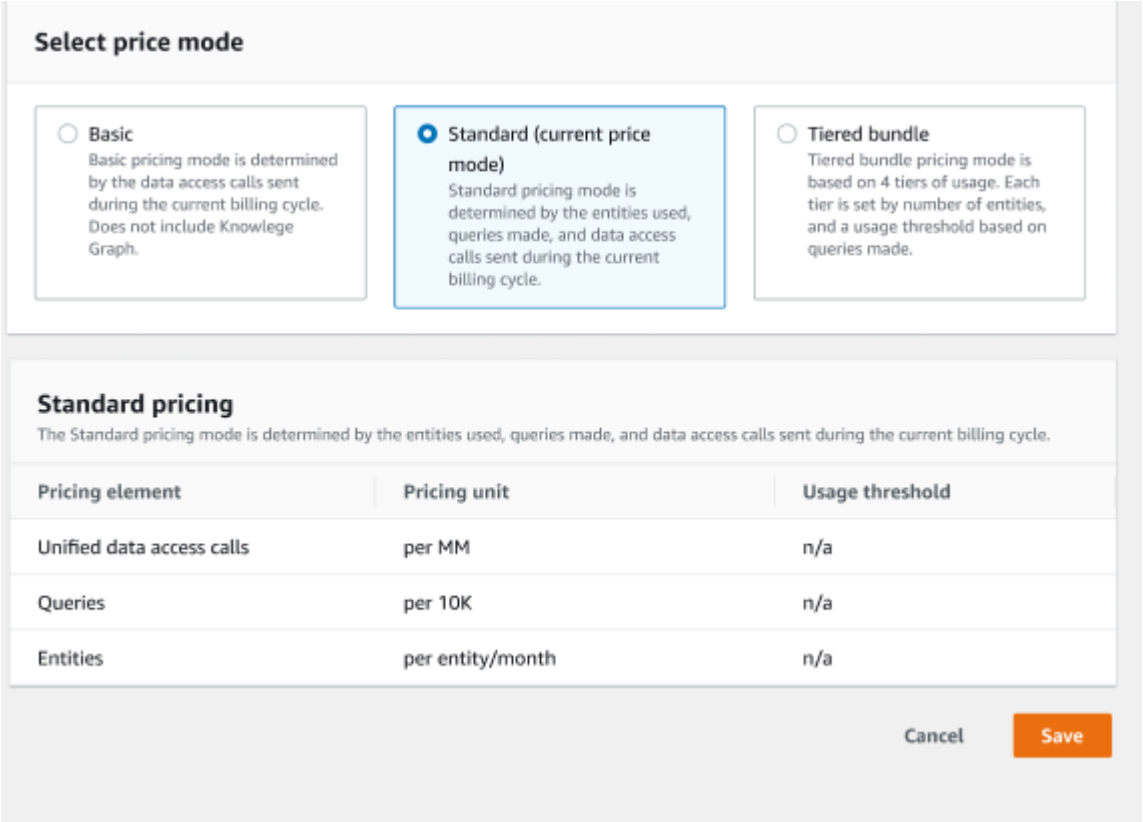
Puede cambiar del modo de precios basado en el uso al basado en niveles en cualquier momento, pero el cambio entrará en vigor al principio del siguiente ciclo de facturación. Una vez que haya cambiado del modo de precios basado en el uso al modo de precios por niveles, no podrá volver al modo de precios basado en el uso durante los tres ciclos de uso siguientes. Si cambia del básico al estándar, el cambio se hará efectivo inmediatamente. Para obtener detalles e información sobre los costos, consulta los precios AWS IoT TwinMaker

Este procedimiento muestra cómo cambiar el modo de precios en la consola de AWS IoT TwinMaker:

1. Abra la consola de AWS CloudFormation.
2. En el panel de navegación izquierdo, selecciona Configuración. Se abre la página de precios.



3. Seleccione el Cambiar modo de precio.
4. Seleccione el modo de paquete estándar o escalonado, como se muestra en la siguiente captura de pantalla.



5. Seleccione Guardar para confirmar el nuevo modo de precios.
6. Ahora ha cambiado el modo de precios.

> **Note**
>
> Puede cambiar del modo de precios basado en el uso al basado en niveles en cualquier momento, pero el cambio entrará en vigor al principio del siguiente ciclo de facturación. Una vez que haya cambiado del modo de precios basado en el uso al modo de precios por niveles, no podrá volver al modo de precios basado en el uso durante los tres ciclos de uso siguientes. Si cambia del básico al estándar, el cambio se hará efectivo inmediatamente.

AWS IoT TwinMaker gráfico de conocimiento

El gráfico de AWS IoT TwinMaker conocimiento organiza toda la información contenida en sus AWS IoT TwinMaker espacios de trabajo y la presenta en un formato de gráfico visual. Puede ejecutar consultas sobre sus entidades, componentes y tipos de componentes para generar gráficos visuales que muestren las relaciones entre sus recursos de AWS IoT TwinMaker .

En los siguientes temas, se le mostrará cómo utilizar e integrar el gráfico de conocimiento.

Temas

- AWS IoT TwinMaker conceptos básicos del gráfico de conocimiento
- ¿Cómo ejecutar consultas de gráficos de AWS IoT TwinMaker conocimiento
- Integración de escenas del gráfico de conocimiento
- Cómo usar el gráfico de AWS IoT TwinMaker conocimiento con Grafana
- AWS IoT TwinMaker gráfico de conocimiento, recursos adicionales

AWS IoT TwinMaker conceptos básicos del gráfico de conocimiento

En este tema se tratan los conceptos clave y el vocabulario de la función del gráfico de conocimiento.

Cómo funciona el gráfico de conocimiento:

> El gráfico de conocimiento crea relaciones entre las entidades y sus componentes con las CreateEntityo existentes UpdateEntity APIs. Una relación es solo una propiedad de un tipo de dato especial, una RELACIÓN que se define en un componente de una entidad. AWS IoT TwinMaker Knowledge Graph llama a la ExecuteQueryAPI para realizar una consulta basada en cualquier dato de las entidades o en las relaciones entre ellas. Knowledge Graph utiliza el flexible lenguaje de consultas PartiQL (utilizado por muchos AWS servicios), al que se ha añadido recientemente la compatibilidad con la sintaxis de coincidencia de gráficos para ayudarte a escribir tus consultas. Una vez realizadas las llamadas, puede ver los resultados en forma de tabla o visualizarlos como un gráfico de nodos y bordes conectados.

Términos clave del gráfico de conocimiento:

- Gráfico de entidades: conjunto de nodos y aristas dentro de un espacio de trabajo.

- Nodo: cada entidad del espacio de trabajo se convierte en un nodo del gráfico de entidades.
- Borde: cada propiedad de relación definida en un componente de una entidad se convierte en un borde en el gráfico de entidades. Además, una relación jerárquica padre-hijo definida mediante el parentEntityId campo de una entidad también se convierte en una arista en el gráfico de la entidad con un nombre de relación «isChildOf». Todas las aristas son aristas direccionales.
- Relación: una AWS IoT TwinMaker relación es un tipo especial de propiedad del componente de una entidad. Puede usar la UpdateEntityAPI AWS IoT TwinMaker CreateEntityo para definir y editar una relación. En AWS IoT TwinMaker, se debe definir una relación en un componente de una entidad. Una relación no se puede definir como un recurso aislado. Una relación debe ser direccional de una entidad a otra.

¿Cómo ejecutar consultas de gráficos de AWS IoT TwinMaker conocimiento

Antes de utilizar el gráfico de AWS IoT TwinMaker conocimiento, asegúrese de haber cumplido los siguientes requisitos previos:

- Cree un AWS IoT TwinMaker espacio de trabajo. Puede crear un bucket utilizando la consola de AWS IoT TwinMaker.
- Familiarícese con AWS IoT TwinMaker el sistema de componentes de entidades y con cómo crear entidades. Para obtener más información, consulte Crear su primera entidad.
- Familiarícese con los conectores AWS IoT TwinMaker de datos. Para obtener más información, consulte AWS IoT TwinMaker conectores de datos.

> **Note**
>
> Para utilizar el gráfico de AWS IoT TwinMaker conocimiento, debes utilizar el modo de precios estándar o por paquetes escalonados. Para obtener más información, consulte Cambiar AWS IoT TwinMaker los modos de precios.

Los procedimientos que aparecen a continuación le muestran cómo escribir, ejecutar, guardar y editar consultas.

Abra el editor de consultas

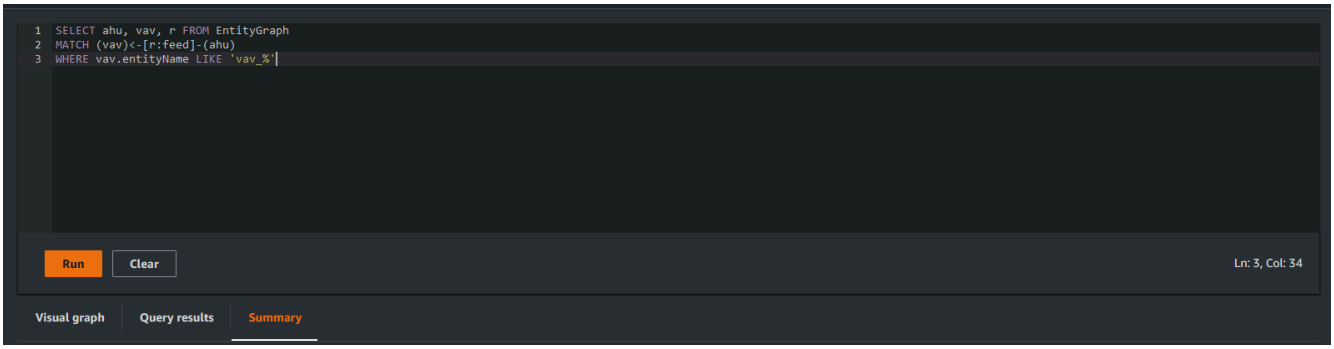
Para navegar al editor de consultas de Knowledge Graph

1. Abra la consola de AWS IoT TwinMaker.
2. Abra el espacio de trabajo en el que desee utilizar el gráfico de conocimiento.
3. En el panel de navegación izquierdo, elija Editor de consultas.
4. Se abre el editor de consultas. Ahora se encuentra preparado para ejecutar consultas en los recursos de su espacio de trabajo.

Ejecutar una consulta

Para ejecutar una consulta y generar un gráfico

1. En el editor de consultas, elija la pestaña Editor para abrir el editor de sintaxis.
2. En el espacio del editor, escribe la consulta que deseas ejecutar con los recursos de tu espacio de trabajo.



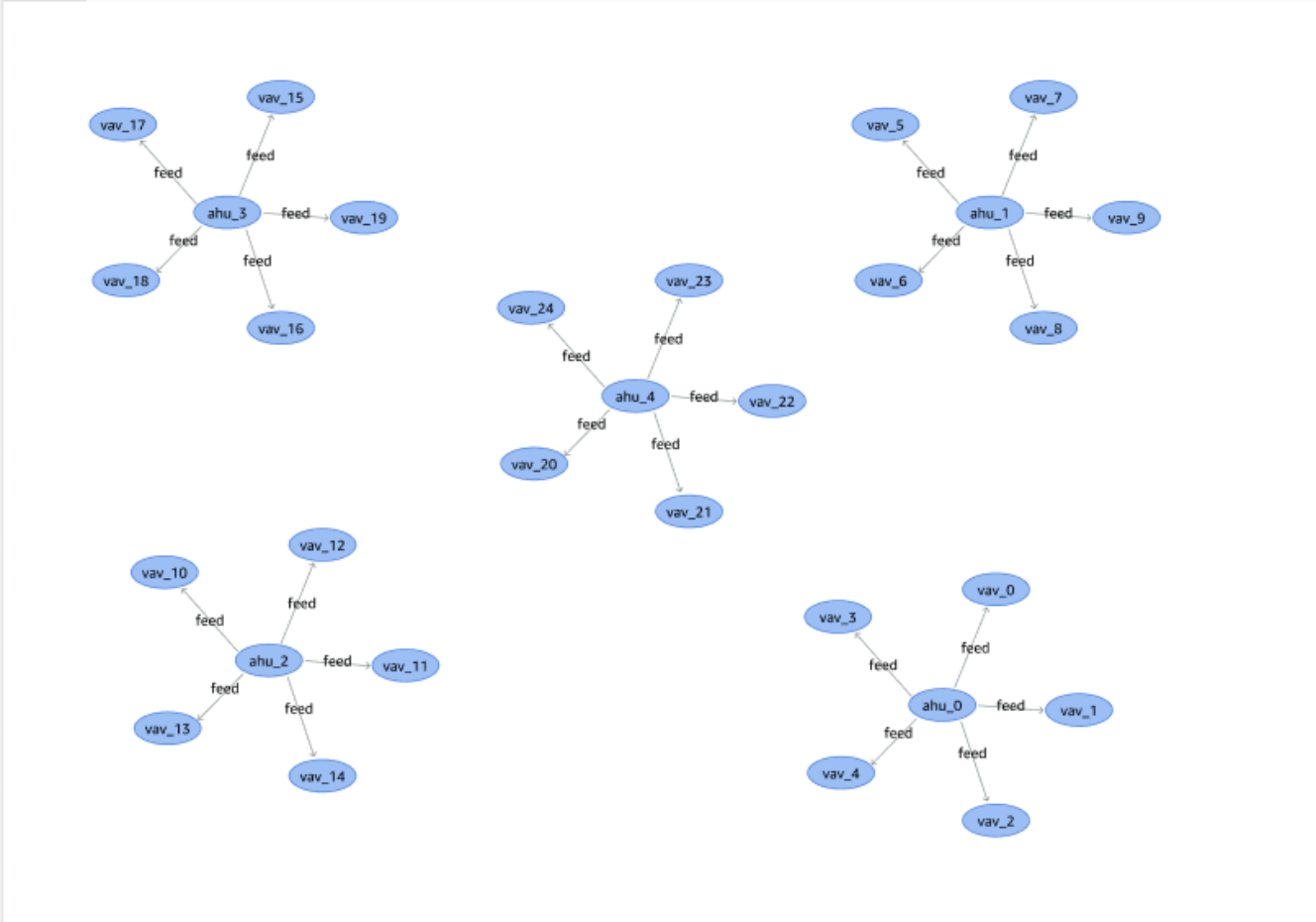
En el ejemplo que se muestra, la solicitud busca entidades que contengan `vav_%` su nombre y, a continuación, las organiza según la `feed` relación entre ellas mediante el siguiente código.

```
SELECT ahu, vav, r FROM EntityGraph
MATCH (vav)<-[r:feed]-(ahu)
WHERE vav.entityName LIKE 'vav_%'
```

> **Note**
>
> La sintaxis del gráfico de conocimiento usa PartiQL. Para obtener información sobre esta sintaxis, consulteAWS IoT TwinMaker gráfico de conocimiento, recursos adicionales.

3. Seleccione Ejecutar consulta para ejecutar la solicitud que ha creado.

   Se genera un gráfico en función de su solicitud.



   El gráfico de ejemplo que se muestra arriba se basa en el ejemplo de consulta del paso 2.

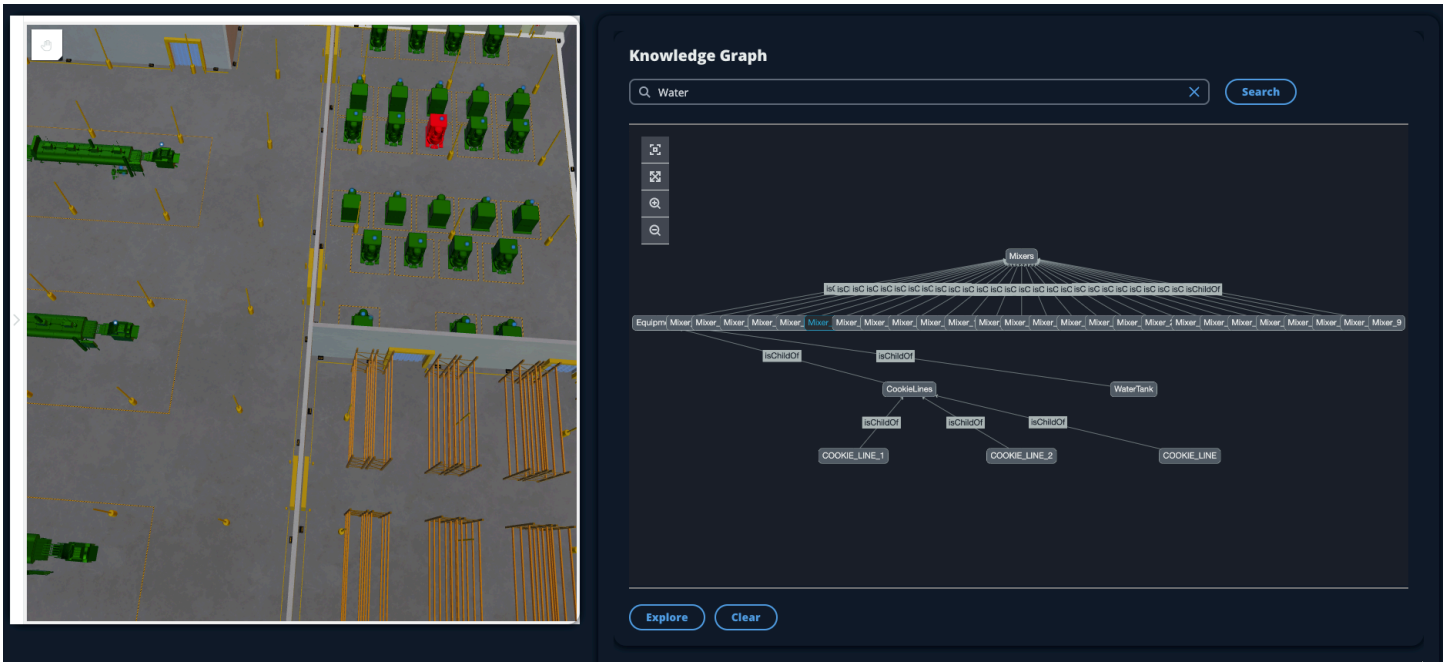
4. Los resultados de la consulta también aparecen en una lista. Seleccione los resultados para ver los resultados de la consulta en una lista.

5. Si lo desea, seleccione Exportar como para exportar los resultados de la consulta en formato JSON o CSV.

Esto cubre el uso básico del gráfico de conocimiento en la consola. Para obtener más información y ejemplos que muestran la sintaxis del gráfico de conocimiento, consulte AWS IoT TwinMaker gráfico de conocimiento, recursos adicionales.

Integración de escenas del gráfico de conocimiento

Puedes usar los componentes del kit de AWS IoT aplicaciones para crear una aplicación web que integre el gráfico de conocimiento en tus AWS IoT TwinMaker escenas. Esto le permite generar gráficos basados en los nodos 3D (los modelos 3D que representan sus equipos o sistemas) que están presentes en la escena. Para crear una aplicación que grafique los nodos 3D de su escena, primero vincule los nodos 3D a las entidades de su espacio de trabajo. Con este mapeo, AWS IoT TwinMaker grafica las relaciones entre los modelos 3D presentes en la escena y las entidades del espacio de trabajo. A continuación, puede crear una aplicación web, seleccionar modelos 3D con su escena y explorar sus relaciones con otras entidades en un formato gráfico.



Para ver un ejemplo de una aplicación web en funcionamiento que utiliza los componentes del kit de AWS IoT aplicaciones para generar gráficos en una AWS IoT TwinMaker escena, consulta el AWS IoT TwinMaker ejemplo de la aplicación React en GitHub.

AWS IoT TwinMaker requisitos previos del gráfico de escena

Antes de crear una aplicación web que utilice el gráfico de AWS IoT TwinMaker conocimiento en las escenas, complete los siguientes requisitos previos:

- Cree un AWS IoT TwinMaker espacio de trabajo. Puede crear un bucket utilizando la consola de AWS IoT TwinMaker.
- Familiarícese con AWS IoT TwinMaker el sistema de componentes de entidades y con cómo crear entidades. Para obtener más información, consulte Crear su primera entidad.
- Cree una AWS IoT TwinMaker escena repleta de modelos 3D.
- Familiarícese con AWS IoT TwinMaker los componentes del kit de AWS IoT aplicaciones. Para obtener más información sobre los AWS IoT TwinMaker componentes, consulteCree una aplicación web personalizada mediante componentes de AWS IoT TwinMaker interfaz de usuario.
- Familiarícese con los conceptos y la terminología clave de los gráficos de conocimiento. Consulte AWS IoT TwinMaker conceptos básicos del gráfico de conocimiento.

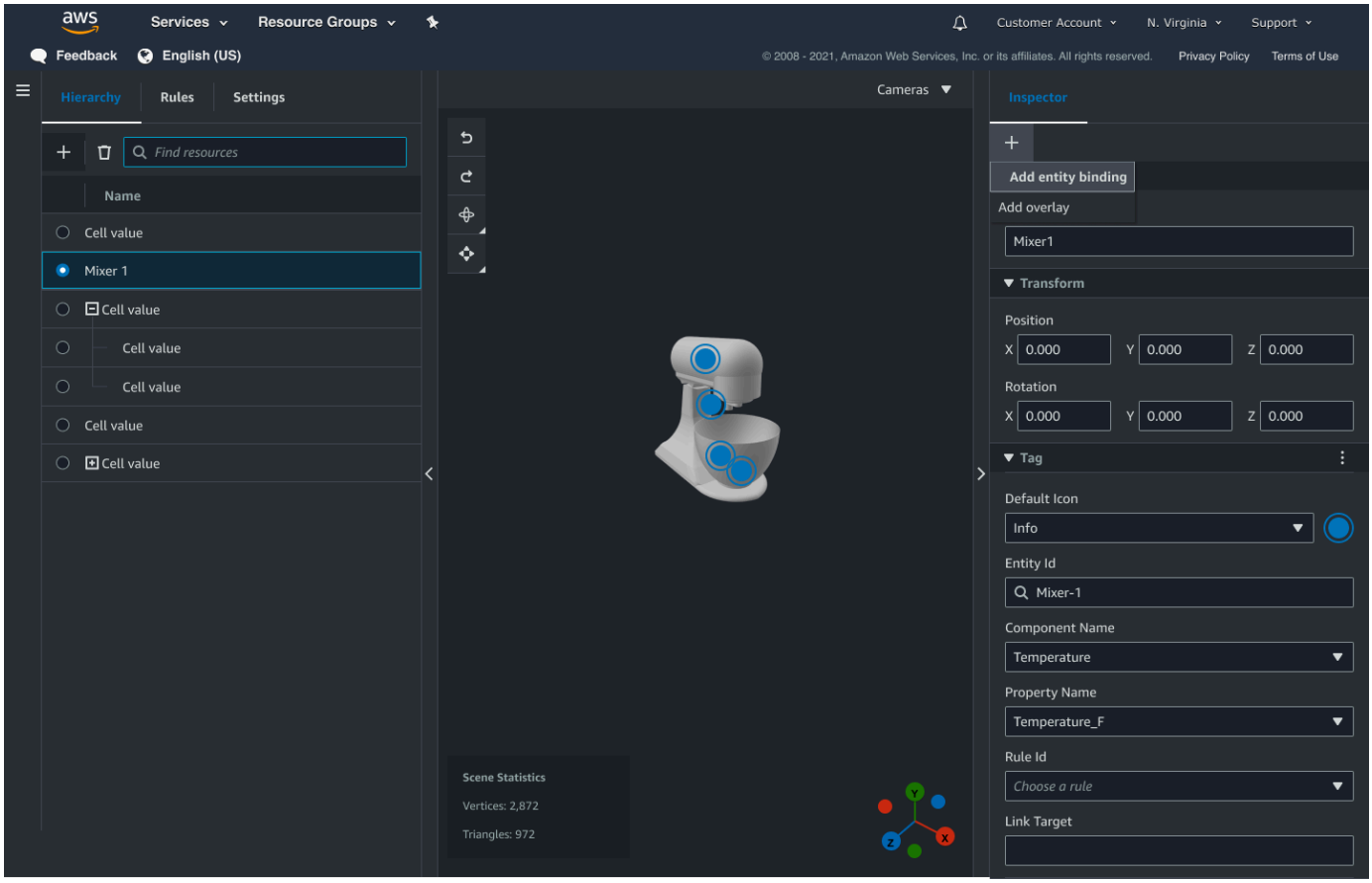
> **Note**
>
> Para usar el gráfico de AWS IoT TwinMaker conocimiento y cualquier función relacionada, debes usar los modos de precios estándar o por paquetes escalonados. Para obtener más información sobre AWS IoT TwinMaker los precios, consulteCambiar AWS IoT TwinMaker los modos de precios.

Enlace nodos 3D en su escena

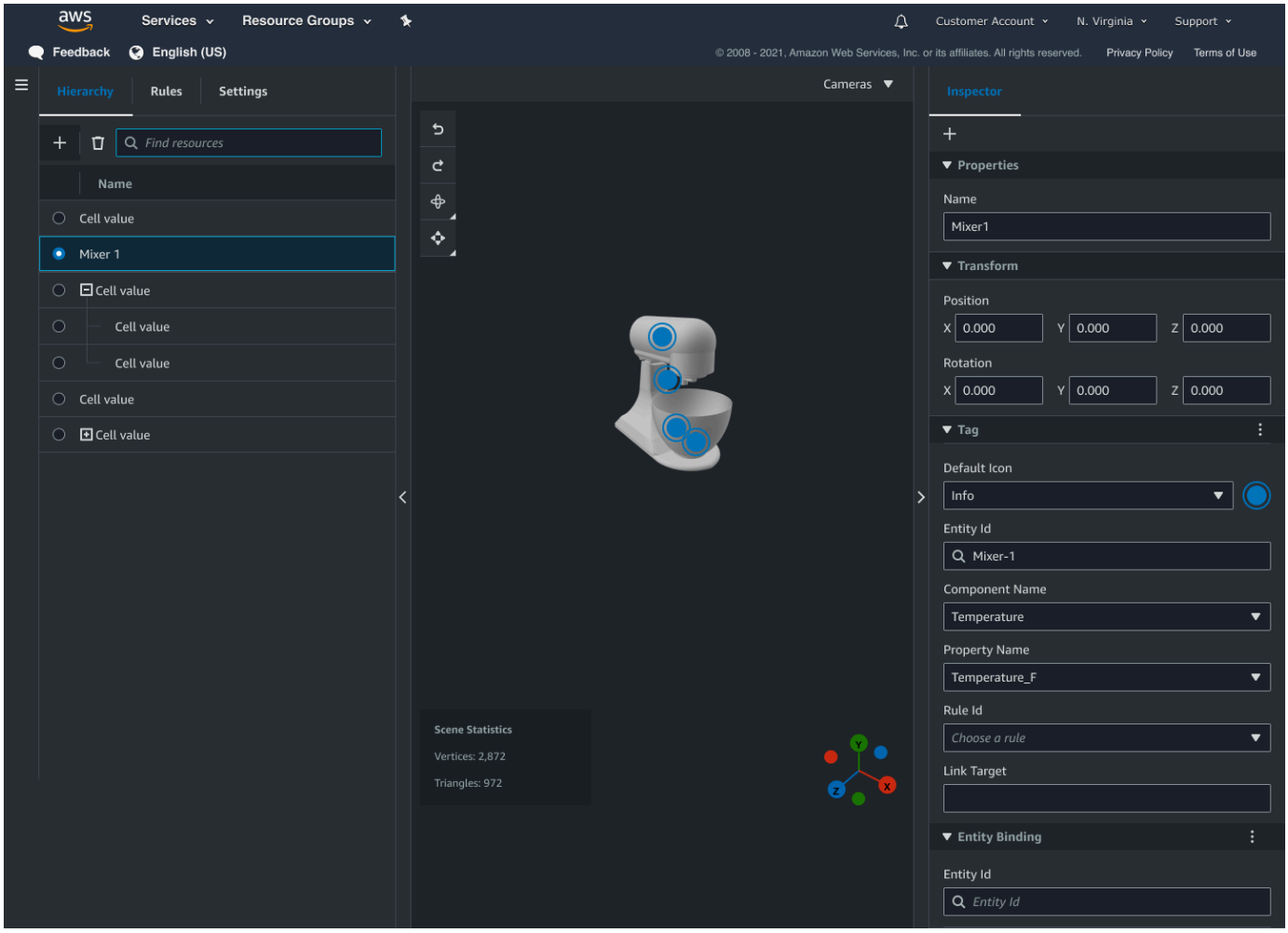
Antes de crear una aplicación web que integre el gráfico de conocimiento con la escena, vincule los modelos 3D, denominados nodos 3D, que están presentes en la escena a la entidad de espacio de trabajo asociada. Por ejemplo, si tiene un modelo de equipo mezclador en una escena y la entidad correspondiente denominada`mixer_0`, cree un enlace de datos entre el modelo del mezclador y la entidad que representa al mezclador, de modo que el modelo y la entidad puedan graficarse.

Para realizar una acción de enlace de datos

1. Inicie sesión en la consola de AWS IoT TwinMaker.
2. Abra su espacio de trabajo y seleccione una escena con los nodos 3D que desee vincular.
3. Seleccione un nodo (modelo 3D) en el compositor de escenas. Al seleccionar un nodo, se abrirá un panel de inspección en la parte derecha de la pantalla.
4. En el panel del inspector, navega hasta la parte superior del panel y selecciona el botón +. A continuación, selecciona la opción Añadir enlace de entidades. Se abrirá un menú desplegable en el que podrá seleccionar una entidad para vincularla al nodo actualmente seleccionado.



5. En el menú desplegable de enlace de datos, seleccione el identificador de la entidad que desee asignar al modelo 3D. En los campos Nombre del componente y Nombre de la propiedad, seleccione los componentes y las propiedades que desee vincular.



Una vez que haya seleccionado los campos ID de entidad, nombre de componente y nombre de propiedad, se habrá completado el enlace.

6. Repita este proceso para todos los modelos y entidades que desee representar gráficamente.

> **Note**
>
> Se puede realizar la misma operación de enlace de datos en las etiquetas de escena. Solo tiene que seleccionar una etiqueta en lugar de una entidad y seguir el mismo proceso para vincular la etiqueta a un nodo.

Crear una aplicación web

Después de vincular las entidades, utilice la biblioteca del kit de AWS IoT aplicaciones para crear una aplicación web con un widget de gráficos de conocimiento que le permita ver la escena y explorar las relaciones entre los nodos y las entidades de la escena.

Use los siguientes recursos para crear su propia aplicación:

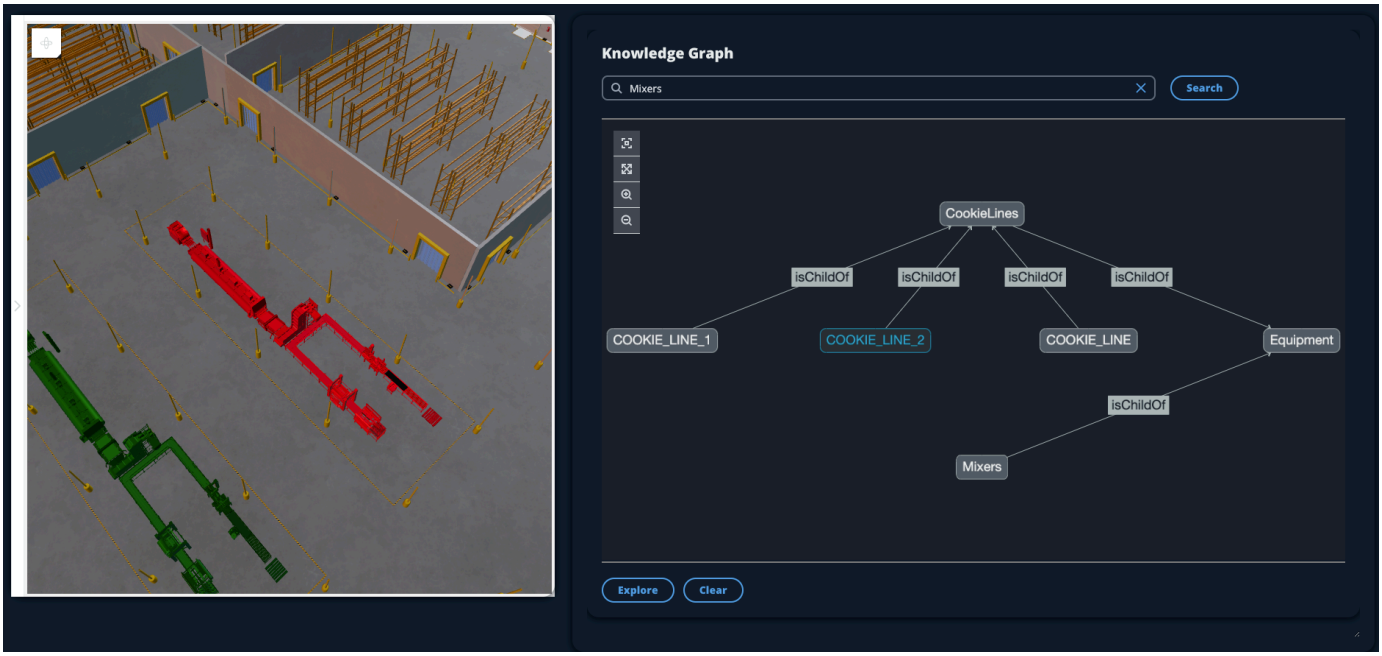
- El AWS IoT TwinMaker ejemplo de documentación readme de la aplicación React en github.
- El AWS IoT TwinMaker ejemplo de la fuente de la aplicación React en github.
- La documentación de introducción del kit de AWS IoT aplicaciones.
- La documentación del componente Video Player del kit de AWS IoT aplicaciones.
- La documentación del componente Scene Viewer del kit de AWS IoT aplicaciones.

El siguiente procedimiento demuestra la funcionalidad del componente del visor de escenas en una aplicación web.

> **Note**
>
> Este procedimiento se basa en la implementación del componente de visor de escenas del kit de AWS IoT aplicaciones en la aplicación React de AWS IoT TwinMaker muestra.

1. Abre el componente del visor de escenas de la aplicación React de AWS IoT TwinMaker muestra. En el campo de búsqueda, escriba un nombre de entidad o un nombre de entidad parcial (búsqueda que distinga mayúsculas de minúsculas) y, a continuación, seleccione el botón Buscar. Si un modelo está vinculado al identificador de la entidad, se resaltará el modelo de la escena y se mostrará un nodo de la entidad en el panel del visor de escenas.

2. Para generar un gráfico de todas las relaciones, seleccione un nodo en el widget del visor de escenas y pulse el botón Explorar.



3. Pulse el botón Borrar para borrar la selección de gráficos actual y volver a empezar.

Cómo usar el gráfico de AWS IoT TwinMaker conocimiento con Grafana

En esta sección, se muestra cómo añadir un panel de edición de consultas a su panel de AWS IoT TwinMaker Grafana para ejecutar y mostrar consultas.

AWS IoT TwinMaker requisitos previos del editor de consultas

Antes de usar el gráfico de AWS IoT TwinMaker conocimiento en Grafana, complete los siguientes requisitos previos:

- Crea un espacio de trabajo AWS IoT TwinMaker . Puede crear un bucket utilizando la consola de AWS IoT TwinMaker.
- Configure AWS IoT TwinMaker para su uso con Grafana. Para obtener instrucciones, consulte Integración del panel de control de AWS IoT TwinMaker Grafana.

> **Note**
>
> Para usar el gráfico de AWS IoT TwinMaker conocimiento, debes usar los modos de precios estándar o por paquetes escalonados. Para obtener más información, consulte Cambiar AWS IoT TwinMaker los modos de precios.

AWS IoT TwinMaker permisos del editor de consultas

Para usar el editor de AWS IoT TwinMaker consultas en Grafana, debes tener un rol de IAM con permiso para la acción. `iottwinmaker:ExecuteQuery` Añade ese permiso a tu rol en el panel de control de tu espacio de trabajo, como se muestra en este ejemplo:

JSON

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

```
                "s3:GetObject"
            ],
            "Resource": [
                "{s3Arn}",
                "{s3Arn}/"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iottwinmaker:Get",
                "iottwinmaker:List",
                "iottwinmaker:ExecuteQuery"
            ],
            "Resource": [
                "{workspaceArn}",
                "{workspaceArn}/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": "iottwinmaker:ListWorkspaces",
            "Resource": "*"
        }
    ]
}
```

> **Note**
>
> Cuando configure la fuente de datos de AWS IoT TwinMaker Grafana, asegúrese de usar el rol con este permiso para el campo ARN de asumir rol. Después de añadirlo, puede seleccionar su espacio de trabajo en el menú desplegable situado junto a Espacio de trabajo.

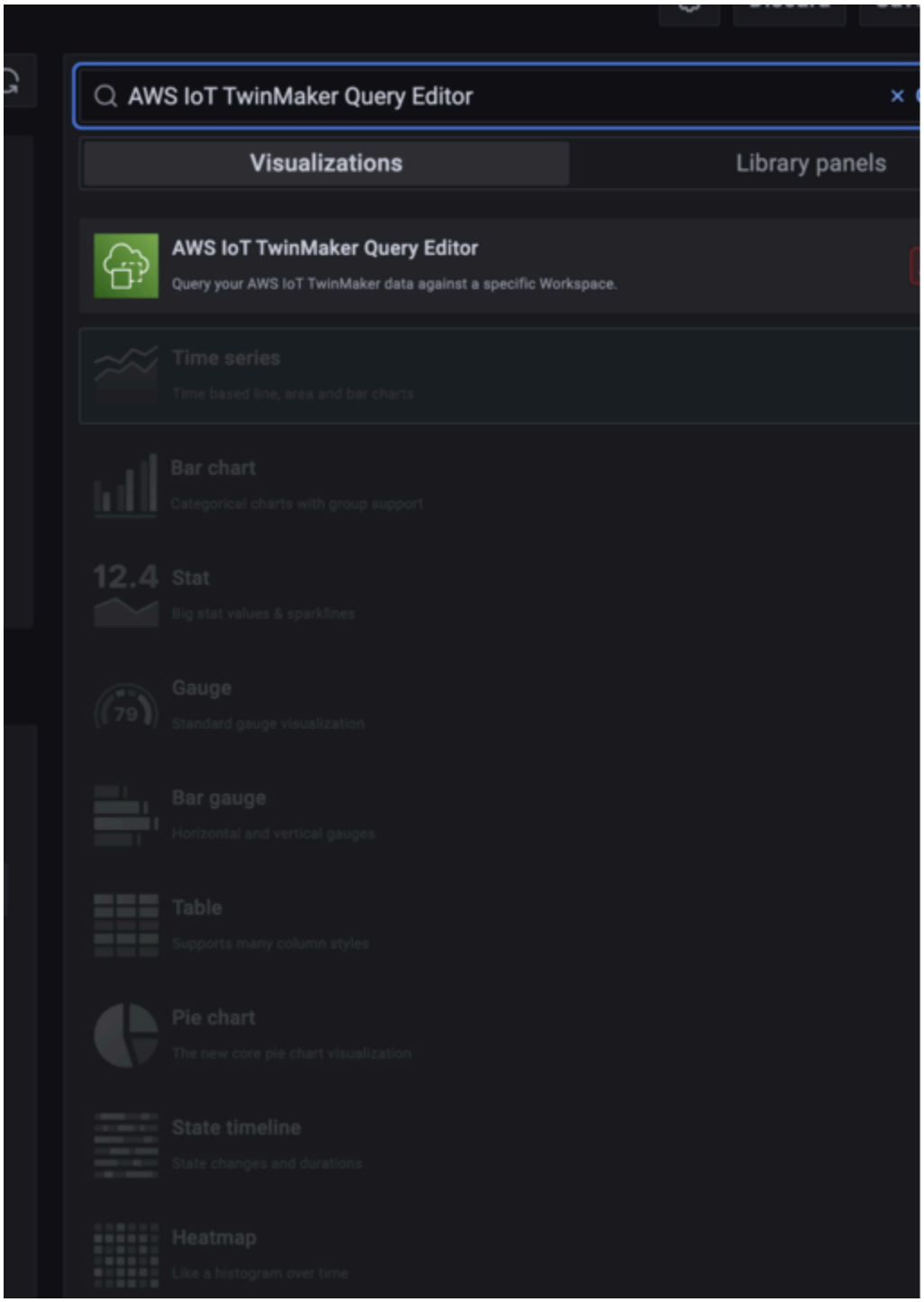
Para obtener más información, consulte Creación de un rol de IAM en el panel.

Configura el panel del editor de AWS IoT TwinMaker consultas

Para configurar un nuevo panel de control de Grafana para Knowledge Graph

1. Abre el panel de control de AWS IoT TwinMaker Grafana.

2. Crea un nuevo panel de control. Para ver los pasos detallados sobre cómo crear un panel, consulte Crear un panel en la documentación de Grafana.
3. En la lista de visualizaciones, selecciona Editor de AWS IoT TwinMaker consultas.



4. Seleccione el origen de datos con la que se van a ejecutar las consultas.
5. (Opcional) Añada un nombre para el nuevo panel en el campo proporcionado.

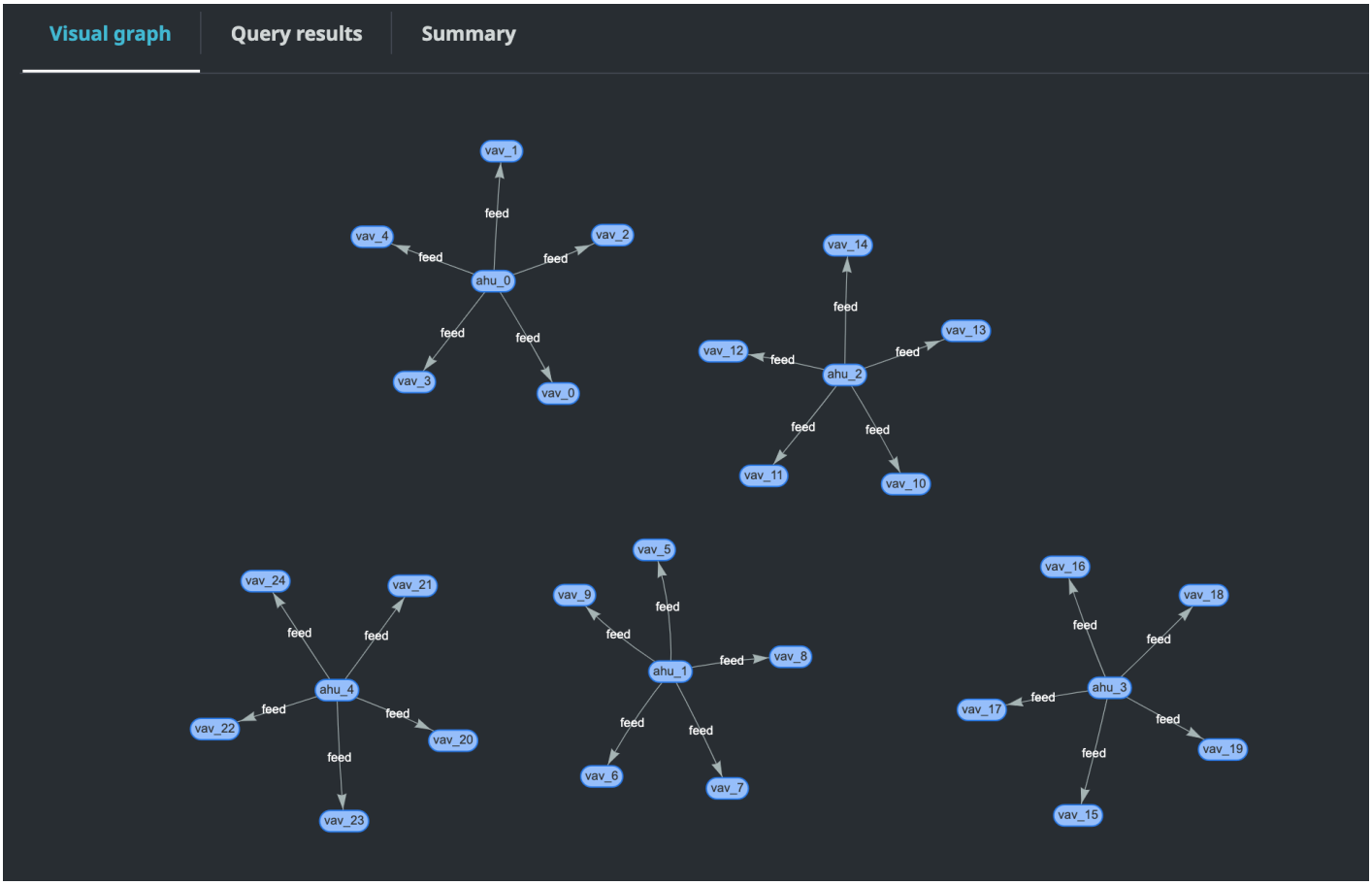
6. Selecciona Aplicar para guardar y confirmar tu nuevo panel.

El panel del gráfico de conocimiento funciona de forma similar al editor de consultas incluido en la AWS IoT TwinMaker consola. Puede ejecutar, escribir y borrar las consultas que realice en el panel. Para obtener más información sobre cómo escribir consultas, consulte[AWS IoT TwinMaker gráfico de conocimiento, recursos adicionales](#).

Cómo utilizar el editor AWS IoT TwinMaker de consultas

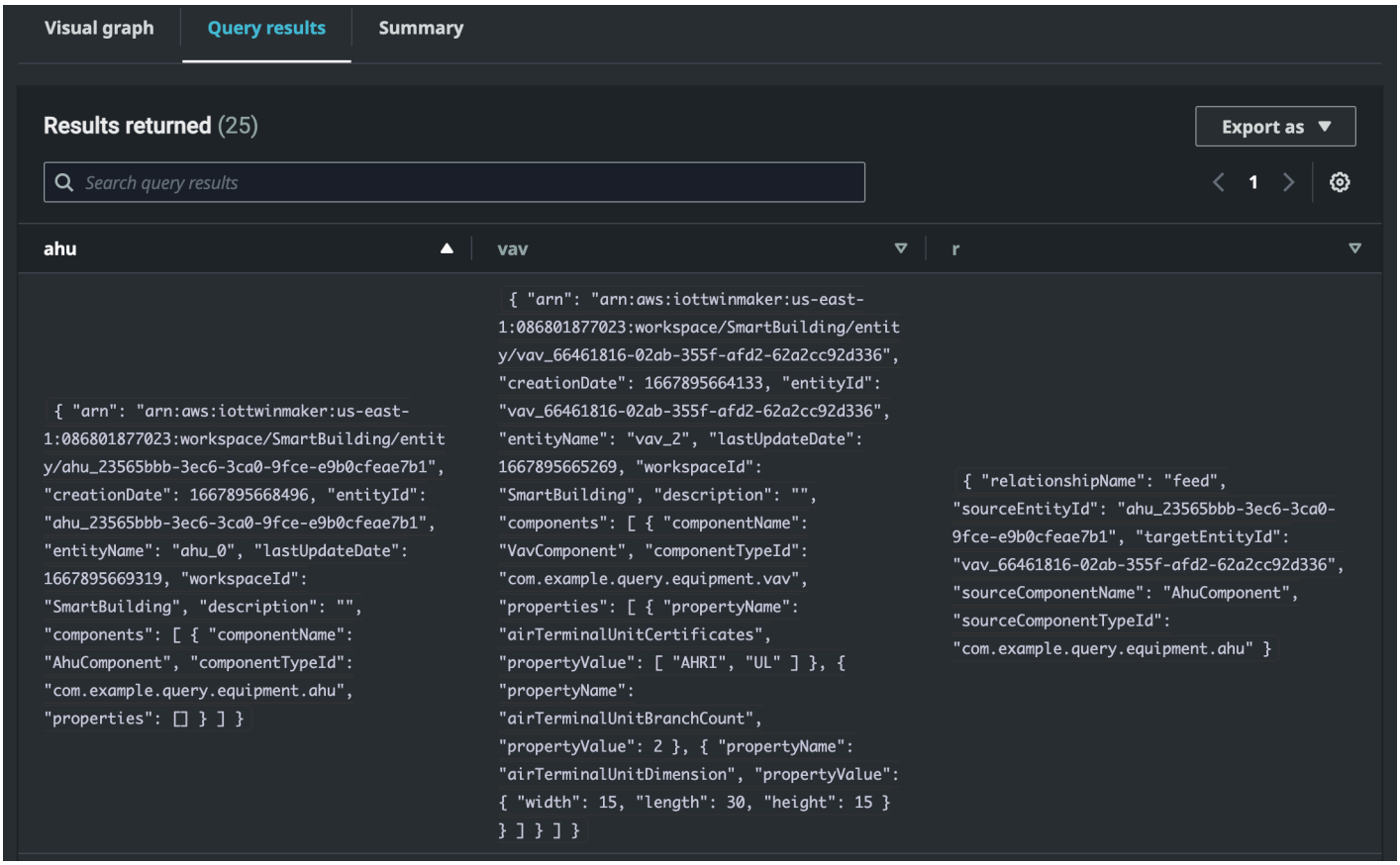
Los resultados de las consultas se muestran de tres formas, como se muestra en las siguientes imágenes: se visualizan en un gráfico, se muestran en una tabla o se presentan como un resumen de la ejecución.

- Visualización de gráficos:



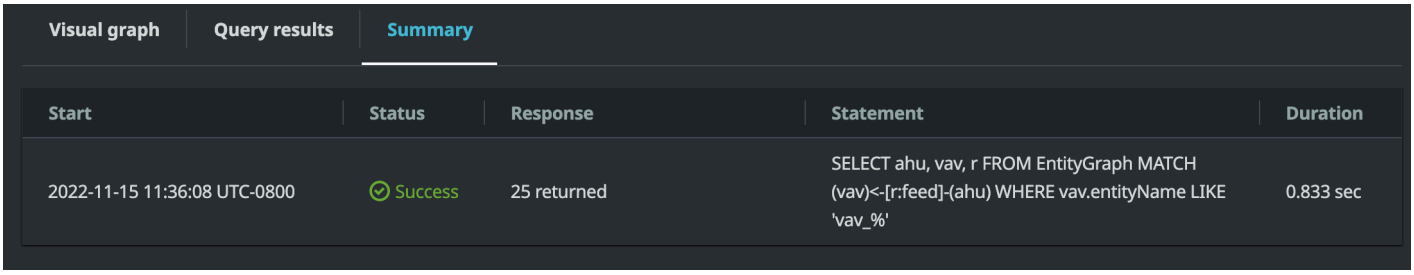
El gráfico visual solo muestra los datos de las consultas que tienen al menos una relación en el resultado. El gráfico muestra las entidades como nodos y las relaciones como bordes dirigidos en el gráfico.

- Datos tabulares:



El formato de datos tabulares muestra los datos de todas las consultas. Puede buscar en la tabla resultados específicos o subconjuntos de los resultados. Los datos se pueden exportar en formato JSON o CSV.

- Resumen de ejecución



El resumen de la ejecución muestra la consulta y los metadatos sobre el estado de la consulta.

AWS IoT TwinMaker gráfico de conocimiento, recursos adicionales

Esta sección proporciona ejemplos básicos de la sintaxis PartiQL utilizada para escribir consultas en el gráfico de conocimiento, así como enlaces a la documentación de PartiQL que proporciona información sobre el modelo de datos del gráfico de conocimiento.

- Documentación del modelo de datos de gráficos partiQL
- Documentación sobre consultas de gráficos partiQL

Este conjunto de ejemplos muestra consultas básicas con sus respuestas. Utilízalo como referencia para escribir tus propias consultas.

Consultas básicas

- Obtenga todas las entidades con un filtro

```
SELECT entity
FROM EntityGraph MATCH (entity)
WHERE entity.entityName = 'room_0'
```

Esta consulta devuelve todas las entidades de un espacio de trabajo con el nombre`room_0`.

`FROM`cláusula: `EntityGraph` es la colección de gráficos que contiene todas las entidades y sus relaciones en un espacio de trabajo. Esta colección se crea y administra automáticamente en AWS IoT TwinMaker función de las entidades de su espacio de trabajo.

Cláusula `MATCH`: especifica un patrón que coincide con una parte del gráfico. En este caso, el patrón `(entity)` coincide con todos los nodos del gráfico y está vinculado a la variable de entidad. La cláusula `FROM` debe ir seguida de la cláusula `MATCH`.

`WHERE`cláusula: especifica un filtro en el `entityName` campo del nodo, donde el valor debe coincidir`room_0`.

`SELECT`cláusula: especifica la `entity` variable para que se devuelva todo el nodo de la entidad.

Respuesta:

```
{
  "columnDescriptions": [
```

```
    {
      "name": "entity",
      "type": "NODE"
    }
  ],
  "rows": [
    {
      "rowData": [
        {
          "arn": "arn:aws:iottwinmaker:us-east-1: 577476956029: workspace /
 SmartBuilding8292022 / entity / room_18f3ef90 - 7197 - 53 d1 - abab -
 db9c9ad02781 ",
          "creationDate": 1661811123914,
          "entityId": "room_18f3ef90-7197-53d1-abab-db9c9ad02781",
          "entityName": "room_0",
          "lastUpdateDate": 1661811125072,
          "workspaceId": "SmartBuilding8292022",
          "description": "",
          "components": [
            {
              "componentName": "RoomComponent",
              "componentTypeId": "com.example.query.construction.room",
              "properties": [
                {
                  "propertyName": "roomFunction",
                  "propertyValue": "meeting"
                },
                {
                  "propertyName": "roomNumber",
                  "propertyValue": 0
                }
              ]
            }
          ]
        }
      ]
    }
  ]
}
```


`columnDescriptions`Devuelve metadatos sobre la columna, como el nombre y el tipo. El tipo devuelto es `NODE`. Esto indica que se ha devuelto todo el nodo. Otros valores del tipo

pueden `EDGE` indicar una relación o `VALUE` indicar un valor escalar, como un entero o una cadena.

`rows` devuelve una lista de filas. Como solo coincidió una entidad, se devuelve un `rowData` que contiene todos los campos de una entidad.

> **Note**
>
> A diferencia de SQL, donde solo puede devolver valores escalares, puede devolver un objeto (como JSON) mediante PartiQL.

Cada nodo contiene todos los campos a nivel de entidad, por ejemplo`entityId`, `arn` y los campos a nivel de componente`components`, así como los campos a nivel de propiedad`componentName`, `properties` como `componentTypeId` y, todo ello en formato JSON `propertyName` anidado. `propertyValue`

- Obtenga todas las relaciones con un filtro:

```
SELECT relationship
FROM EntityGraph MATCH (e1)-[relationship]->(e2)
WHERE relationship.relationshipName = 'isLocationOf'
```

Esta consulta devuelve todas las relaciones de un espacio de trabajo con el nombre de la relación `isLocationOf`.

La `MATCH` cláusula: especifica un patrón que coincide con dos nodos (indicados por`( )`) que están conectados por un borde dirigido (indicados por`-[]->`) y enlazados a una variable llamada. `relationship`

La `WHERE` cláusula: especifica un filtro en el `relationshipName` campo del borde, donde está el valor`isLocationOf`.

Cláusula `SELECT`: especifica la variable de relación para que se devuelva todo el nodo de borde.

Respuesta

```
{
    "columnDescriptions": [{
```

```
        "name": "relationship",
        "type": "EDGE"
    }],
    "rows": [{
        "rowData": [{
            "relationshipName": "isLocationOf",
            "sourceEntityId": "floor_83faea7a-ea3b-56b7-8e22-562f0cf90c5a",
            "targetEntityId": "building_4ec7f9e9-e67e-543f-9d1b- 235df7e3f6a8",
            "sourceComponentName": "FloorComponent",
            "sourceComponentTypeId": "com.example.query.construction.floor"
        }]
    },
        ... //rest of the rows are omitted
    ]
}
```

El tipo de columna `columnDescriptions` es un`EDGE`.

Cada uno `rowData` representa una arista con campos como`relationshipName`.
Es el mismo que el nombre de la propiedad de relación definido en la entidad. Los `sourceEntityId` `sourceComponentName` y `sourceComponentTypeId` proporcionan información sobre la entidad y el componente en los que se definió la propiedad de relación. `targetEntityId`Especifica a qué entidad apunta esta relación.

- Obtenga todas las entidades con una relación específica con una entidad específica

```
SELECT e2.entityName
      FROM EntityGraph MATCH (e1)-[r]->(e2)
      WHERE relationship.relationshipName = 'isLocationOf'
      AND e1.entityName = 'room_0'
```

Esta consulta devuelve todos los nombres de las entidades que tienen una `isLocationOf` relación con la `room_0` entidad.

La `MATCH` cláusula: especifica un patrón que coincide con dos nodos (`e1`,`e2`) que tengan una arista dirigida (`r`).

Cláusula `WHERE`: especifica un filtro en el nombre de la relación y el nombre de la entidad de origen.

La `SELECT` cláusula: devuelve el `entityName` campo del `e2` nodo.

Respuesta

```
{
  "columnDescriptions": [
    {
       "name": "entityName",
       "type": "VALUE"
    }
  ],
   "rows": [
    {
       "rowData": [
         "floor_0"
      ]
    }
  ]
}
```

En las descripciones de columnas, el tipo de columna es `VALUE` ya que `entityName` es una cadena.

Se devuelve una entidad,`floor_0`,.

MATCH

Una `MATCH` cláusula admite los siguientes patrones:

- Haga coincidir el nodo «b» con el nodo «a»:

```
FROM EntityGraph MATCH (a)-[rel]-(b)
```

- Haga coincidir el nodo «a» con el nodo «b»:

```
FROM EntityGraph MATCH (a)-[]->(b)
```

No hay ninguna variable vinculada a una relación, suponiendo que no es necesario especificar un filtro en la relación.

- Haga coincidir el nodo «a» que apunta al nodo «b» y el nodo «b» que apunta al nodo «a»:

```
FROM EntityGraph MATCH (a)-[rel]-(b)
```

Esto devolverá dos coincidencias: una de «a» a «b» y otra de «b» a «a», por lo que se recomienda utilizar bordes dirigidos siempre que sea posible.

- El nombre de la relación también es una etiqueta del gráfico de propiedades`EntityGraph`, por lo que simplemente puede especificar el nombre de la relación después de dos puntos (:)) en lugar de especificar un filtro `rel.relationshipName` en la `WHERE` cláusula.

```
FROM EntityGraph MATCH (a)-[:isLocationOf]-(b)
```

- Encadenamiento: los patrones se pueden encadenar para que coincidan en varias relaciones.

```
FROM EntityGraph MATCH (a)-[rel1]->(b)-[rel2]-(c)
```

- Los patrones de saltos variables también pueden abarcar varios nodos y bordes:

```
FROM EntityGraph MATCH (a)-[]->{1,5}(b)
```

Esta consulta hace coincidir cualquier patrón con los bordes salientes del nodo «a» con un intervalo de 1 a 5 saltos. Los cuantificadores permitidos son:

`{m,n}`: entre m y n repeticiones

`{m,}`: m o más repeticiones.

FROM:

Un nodo de entidad puede contener datos anidados, como componentes que, a su vez, contienen más datos anidados, como propiedades. Se puede acceder a ellos deshaciendo el resultado del patrón MATCH.

```
SELECT e
FROM EntityGraph MATCH (e), e.components AS c, c.properties AS p
WHERE c.componentTypeId = 'com.example.query.construction.room',
AND p.propertyName = 'roomFunction'
AND p.propertyValue = 'meeting'
```

Acceda a los campos anidados colocando puntos `.` en una variable. Se usa una coma (,) para separar (o unir) las entidades con los componentes dentro y, después, las propiedades dentro de esos componentes. ASse usa para vincular una variable a las variables no anidadas para que puedan usarse en las cláusulas o. `WHERE` `SELECT` Esta consulta devuelve todas las

entidades que contienen una propiedad denominada `roomFunction` con un valor `meeting` en un componente con el id de tipo de componente `com.example.query.construction.room`

Para acceder a varios campos anidados de un campo, como varios componentes de una entidad, utilice la notación de coma para realizar una unión.

```
SELECT e
FROM EntityGraph MATCH (e), e.components AS c1, e.components AS c2
```

SELECT

- Devuelve un nodo:

  ```
  SELECT e
  FROM EntityGraph MATCH (e)
  ```

- Devuelve una ventaja:

  ```
  SELECT r
  FROM EntityGraph MATCH (e1)-[r]->(e2)
  ```

- Devuelve un valor escalar:

  ```
  SELECT floor.entityName, room.description, p.propertyValue AS roomfunction
  FROM EntityGraph MATCH (floor)-[:isLocationOf]-(room),
  room.components AS c, c.properties AS p
  ```

  Formatee el nombre del campo de salida asignándole un alias con AS. Aquí, en lugar de `propertyValue` como nombre de columna en la respuesta, se devuelve `roomfunction`.

- Devolver alias:

  ```
  SELECT floor.entityName AS floorName, luminaire.entityName as luminaireName
  FROM EntityGraph MATCH (floor)-[:isLocationOf]-(room)-[:hasPart]-
  (lightingZone)-[:feed]-(luminaire)
  WHERE floor.entityName = 'floor_0'
  AND luminaire.entityName like 'lumin%'
  ```

  Se recomienda encarecidamente utilizar alias para ser explícito, aumentar la legibilidad y evitar cualquier ambigüedad en las consultas.

WHERE:

- Los operadores lógicos compatibles sonAND, y. NOT OR
- Los operadores de comparación compatibles son <, <=, >, =>, = y !=.
- Utilice la IN palabra clave si desea especificar varias OR condiciones en el mismo campo.
- Filtre por un campo de entidad, componente o propiedad:

```
FROM EntityGraph MATCH (e), e.components AS c, c.properties AS p
WHERE e.entityName = 'room_0'
AND c.componentTypeId = 'com.example.query.construction.room',
AND p.propertyName = 'roomFunction'
AND NOT p.propertyValue = 'meeting'
OR p.propertyValue = 'office'
```

- Filtra por `configuration` propiedad. Aquí `unit` está la clave del mapa de configuración y `Celsius` es el valor.

```
WHERE p.definition.configuration.unit = 'Celsius'
```

- Compruebe si una propiedad del mapa contiene una clave y un valor determinados:

```
WHERE p.propertyValue.length = 20.0
```

- Compruebe si una propiedad del mapa contiene una clave determinada:

```
WHERE NOT p.propertyValue.length IS MISSING
```

- Compruebe si una propiedad de la lista contiene un valor determinado:

```
WHERE 10.0 IN p.propertyValue
```

- Utilice la función `lower()` para realizar comparaciones que no distingan mayúsculas de minúsculas. De forma predeterminada, la comparación distingue entre mayúsculas y minúsculas.

```
WHERE lower(p.propertyValue) = 'meeting'
```

LIKE:

Es útil si no conoce el valor exacto de un campo y puede realizar una búsqueda de texto completo en el campo especificado. % representa cero o más.

```
WHERE e.entityName LIKE '%room%'
```

- Búsqueda de infijos: `%room%`
- Búsqueda de prefijos: `room%`
- Búsqueda de sufijos: `%room`
- Si sus valores son «%», coloque un carácter de escape en `LIKE` y especifique el carácter de escape con`ESCAPE`.

```
WHERE e.entityName LIKE 'room\%' ESCAPE '\'
```

DISTINCT:

```
SELECT DISTINCT c.componentTypeId
FROM EntityGraph MATCH (e), e.components AS c
```

- La palabra clave `DISTINCT` elimina los duplicados del resultado final.

  `DISTINCT` no se admite en tipos de datos complejos.

COUNT

```
SELECT COUNT(e), COUNT(c.componentTypeId)
FROM EntityGraph MATCH (e), e.components AS c
```

- La `COUNT` palabra clave calcula el número de elementos del resultado de una consulta.
- `COUNT`no se admite en campos complejos anidados ni en campos de patrones de gráficos.
- `COUNT`la agregación no es compatible con `DISTINCT` las consultas anidadas.

  Por ejemplo, `COUNT(DISTINCT e.entityId)` no se admite.

RUTA

Las siguientes proyecciones de patrones son compatibles con las consultas mediante la proyección de trayectoria:

- Consultas de saltos variables

```
SELECT p FROM EntityGraph MATCH p = (a)-[]->{1, 3}(b)
```

Esta consulta hace coincidir los metadatos de los nodos de cualquier patrón y los proyecta, con los bordes salientes del nodo a en un intervalo de 1 a 3 saltos.

- Consultas de salto fijo

```
SELECT p FROM EntityGraph MATCH p = (a)-[]->(b)<-[]-(c)
```

Esta consulta coincide con los metadatos de las entidades y los bordes entrantes y los proyecta en b.

- Consultas no dirigidas

```
SELECT p FROM EntityGraph MATCH p = (a)-[]-(b)-[]-(c)
```

Esta consulta compara y proyecta los metadatos de los nodos en patrones de 1 salto que conectan a y c a través de b.

```
{
    "columnDescriptions": [
        {
            "name": "path",
            "type": "PATH"
        }
    ],
    "rows": [
        {
            "rowData": [
                {
                    "path": [
                        {
                            "entityId": "a",
                            "entityName": "a"
                        },
                        {
                            "relationshipName": "a-to-b-relation",
                            "sourceEntityId": "a",
                            "targetEntityId": "b"
                        },
                        {
                            "entityId": "b",
                            "entityName": "b"
                        }
```

```
                    ]
                }
            ]
        },
        {
            "rowData": [
                {
                    "path": [
                        {
                            "entityId": "b",
                            "entityName": "b"
                        },
                        {
                            "relationshipName": "b-to-c-relation",
                            "sourceEntityId": "b",
                            "targetEntityId": "c"
                        },
                        {
                            "entityId": "c",
                            "entityName": "c"
                        }
                    ]
                }
            ]
        }
    ]
}
```

Esta respuesta a la PATH consulta se compone únicamente de metadatos que identifican todos los nodos y bordes de cada uno de ellos path/pattern entre a y c a través de b.

LÍMITE Y COMPENSACIÓN:

```
SELECT e.entityName
FROM EntityGraph MATCH (e)
WHERE e.entityName LIKE 'room_%'
LIMIT 10
OFFSET 5
```

`LIMIT` especifica el número de resultados que se devolverán en la consulta y `OFFSET` el número de resultados que se van a omitir.

LIMIT y MaxResults:

El siguiente ejemplo muestra una consulta que devuelve 500 resultados en total, pero solo muestra 50 a la vez por llamada a la API. Este patrón se puede usar cuando necesite limitar la cantidad de resultados mostrados, por ejemplo, si solo puede mostrar 50 resultados en una interfaz de usuario.

```
aws iottwinmaker execute-query \
--workspace-id exampleWorkspace \
--query-statement "SELECT e FROM EntityGraph MATCH (e) LIMIT 500"\
--max-results 50
```

- La `LIMIT` palabra clave afecta a la consulta y limita las filas resultantes. Si necesitas controlar el número de resultados devueltos por llamada a la API sin limitar el número total de resultados devueltos, utiliza`LIMIT`.
- `max-results`es un parámetro opcional para la acción de la ExecuteQuery API. `max-results`solo se aplica a la API y a la forma en que se leen los resultados dentro de los límites de la consulta anterior.

  El uso `max-results` en una consulta permite reducir el número de resultados mostrados sin limitar el número real de resultados devueltos.

La siguiente consulta recorre la siguiente página de resultados. Esta consulta utiliza la llamada a la `ExecuteQuery` API para devolver las filas 51 a 100, donde la siguiente página de resultados se especifica mediante el símbolo `next-token` (en este caso, el token es:). `"H7kyGmvK376L"`

```
aws iottwinmaker execute-query \
--workspace-id exampleWorkspace \
--query-statement "SELECT e FROM EntityGraph MATCH (e) LIMIT 500"\
--max-results 50
--next-token "H7kyGmvK376L"
```

- La `next-token` cadena especifica la siguiente página de resultados. Para obtener más información, consulta la acción ExecuteQueryde la API.

AWS IoT TwinMaker La consulta de Knowledge Graph tiene los siguientes límites:

Nombre del límite	Cuota	Ajustable
Tiempo de espera de ejecución de la consulta	10 segundos	No
Número máximo de saltos	10	Sí
Número máximo de sí mismos `JOIN`	20	Sí
Número máximo de campos proyectados	20	Sí
Número máximo de expresion es condicionales (`AND`,`OR`,`NOT`)	10	Sí
Longitud máxima de un patrón de `LIKE` expresión (incluidos los caracteres comodín y los escapes)	20	Sí
Número máximo de elementos que se pueden especificar en una cláusula `IN`	10	Sí
Valor máximo para `OFFSET`	3 000	Sí
Valor máximo para `LIMIT`	3 000	Sí
Valor máximo para los recorridos (+) `OFFSET` `LIMIT`	3 000	Sí

Sincronización de activos con AWS IoT SiteWise

AWS IoT TwinMaker admite la sincronización de activos (sincronización de activos) para sus AWS IoT SiteWise activos y modelos de activos. Al usar el tipo de AWS IoT SiteWise componente, la sincronización de AWS IoT SiteWise activos toma los activos y modelos de activos existentes y los convierte en AWS IoT TwinMaker entidades, componentes y tipos de componentes. En las siguientes secciones, se explica cómo configurar la sincronización de activos y qué AWS IoT SiteWise activos y modelos de activos se pueden sincronizar con su AWS IoT TwinMaker espacio de trabajo.

Temas

- Uso de la sincronización de activos con AWS IoT SiteWise
- Diferencias entre los espacios de trabajo personalizados y predeterminados
- Recursos sincronizados desde AWS IoT SiteWise
- Analice el estado y los errores de la sincronización
- Eliminar un trabajo de sincronización
- Límites de la sincronización de activos

Uso de la sincronización de activos con AWS IoT SiteWise

En este tema, se muestra cómo activar y configurar la sincronización AWS IoT SiteWise de activos. Siga los procedimientos adecuados en función del tipo de espacio de trabajo que utilice.

> **Important**
>
> Consulte the section called “Diferencias entre los espacios de trabajo personalizados y predeterminados” para obtener información sobre las diferencias entre los espacios de trabajo personalizados y predeterminados.

Temas

- Uso de un espacio de trabajo personalizado
- Uso del lo TSite WiseDefaultWorkspace

Uso de un espacio de trabajo personalizado

Revise estos requisitos previos antes de activar la sincronización de activos.

Requisitos previos

Antes de usarlo AWS IoT SiteWise, debes completar lo siguiente:

- Dispones de un AWS IoT TwinMaker espacio de trabajo.
- Tienes activos y modelos de activos AWS IoT SiteWise. Para obtener más información, consulte Creación de modelos de activos.
- Un rol de IAM existente con permisos de lectura para las siguientes AWS IoT SiteWise acciones:
  - `ListAssets`
  - `ListAssetModels`
  - `DescribeAsset`
  - `DescribeAssetModel`
- El rol de IAM debe tener los siguientes permisos de escritura para: AWS IoT TwinMaker
  - `CreateEntity`
  - `UpdateEntity`
  - `DeleteEntity`
  - `CreateComponentType`
  - `UpdateComponentType`
  - `DeleteComponentType`
  - `ListEntities`
  - `GetEntity`
  - `ListComponentTypes`

  Utilice la siguiente función de IAM como plantilla para la función requerida:

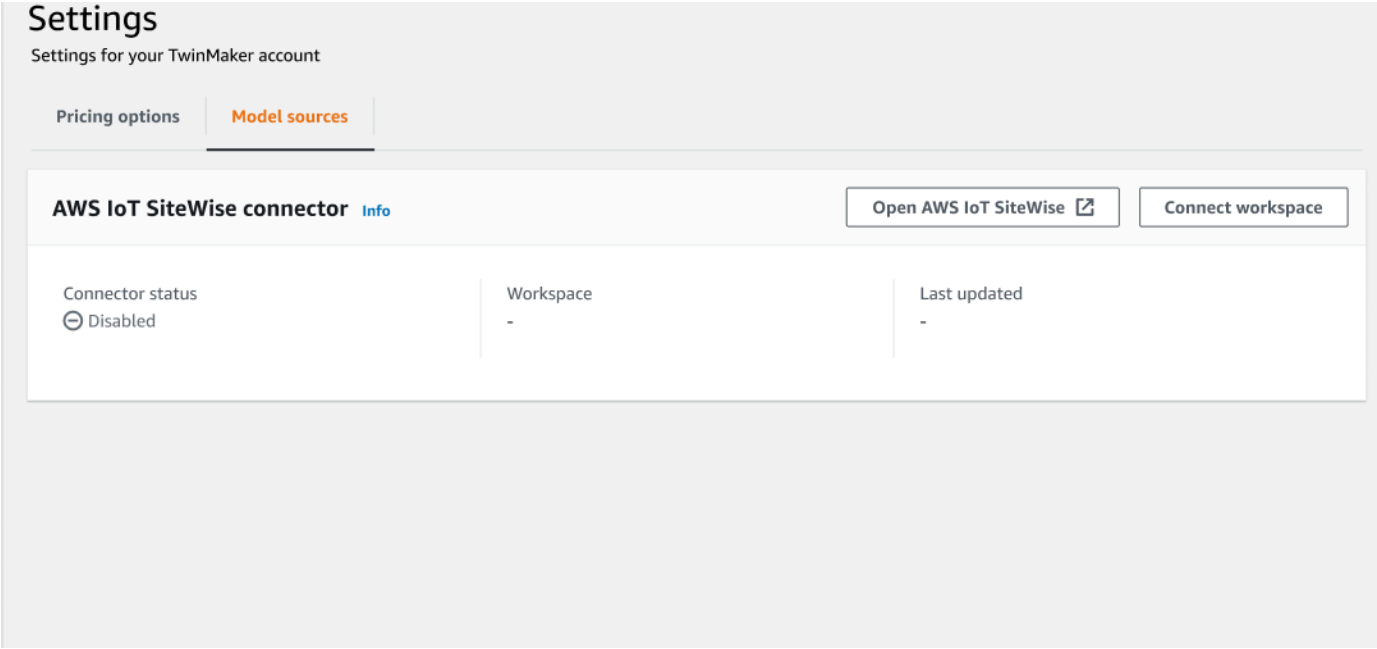
```
// trust relationships
 {
    {
        "Version": "2012-10-17",
        "Statement": [
```

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

// permissions - replace ACCOUNT_ID, REGION, WORKSPACE_ID with actual values
{
    "Version": "2012-10-17",
    "Statement": [{
            "Sid": "SiteWiseAssetReadAccess",
            "Effect": "Allow",
            "Action": [
                "iotsitewise:DescribeAsset"
            ],
            "Resource": [
                "arn:aws:iotsitewise:REGION:ACCOUNT_ID:asset/*"
            ]
        },
        {
            "Sid": "SiteWiseAssetModelReadAccess",
            "Effect": "Allow",
            "Action": [
                "iotsitewise:DescribeAssetModel"
            ],
            "Resource": [
                "arn:aws:iotsitewise:REGION:ACCOUNT_ID:asset-model/*"
            ]
        },
        {
            "Sid": "SiteWiseAssetModelAndAssetListAccess",
            "Effect": "Allow",
            "Action": [
                "iotsitewise:ListAssets",
                "iotsitewise:ListAssetModels"
            ],
            "Resource": [
                "*"
```

```
            ]
        },
        {
            "Sid": "TwinMakerAccess",
            "Effect": "Allow",
            "Action": [
                "iottwinmaker:GetEntity",
                "iottwinmaker:CreateEntity",
                "iottwinmaker:UpdateEntity",
                "iottwinmaker:DeleteEntity",
                "iottwinmaker:ListEntities",
                "iottwinmaker:GetComponentType",
                "iottwinmaker:CreateComponentType",
                "iottwinmaker:UpdateComponentType",
                "iottwinmaker:DeleteComponentType",
                "iottwinmaker:ListComponentTypes"
            ],
            "Resource": [
                "arn:aws:iottwinmaker:REGION:ACCOUNT_ID:workspace/WORKSPACE_ID",
                "arn:aws:iottwinmaker:REGION:ACCOUNT_ID:workspace/WORKSPACE_ID/*"

            ]
        }
    ]
}
```

Utilice el siguiente procedimiento para activar y configurar la sincronización de activos de AWS IoT SiteWise .

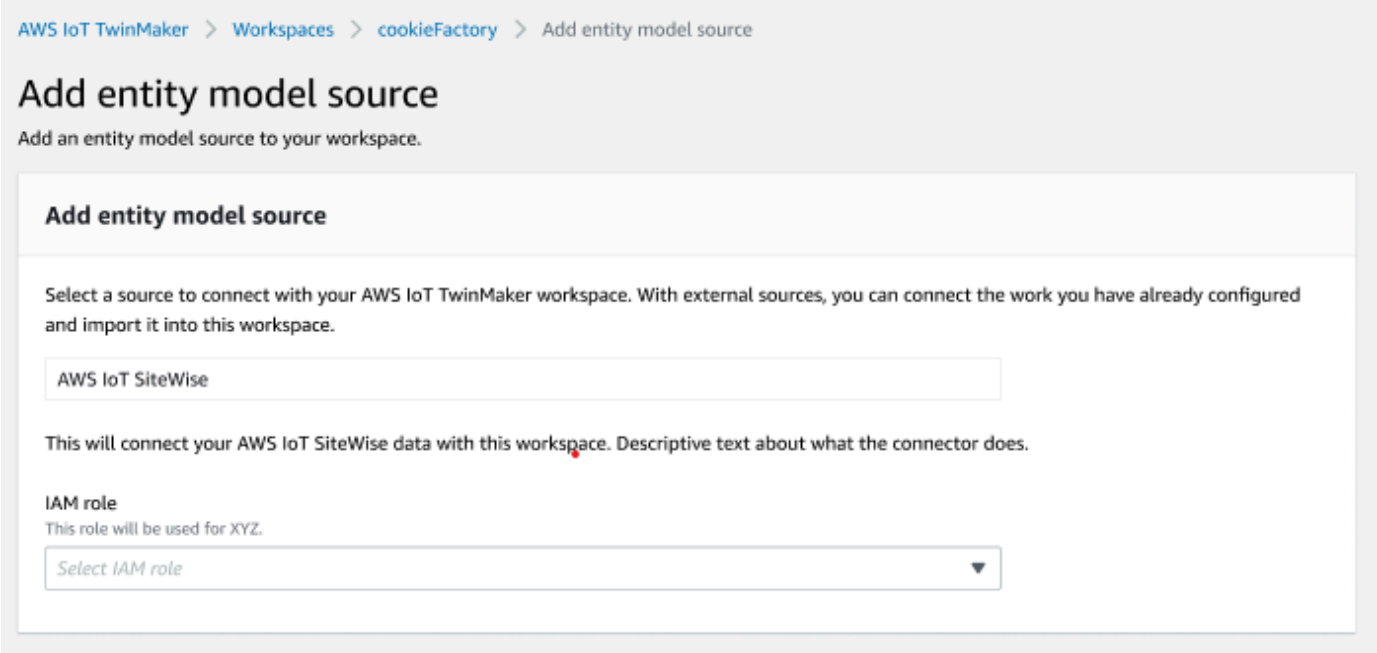
1. En la consola de AWS IoT TwinMaker, vaya a la página Ajustes.
2. Abra la pestaña Orígenes de modelos.



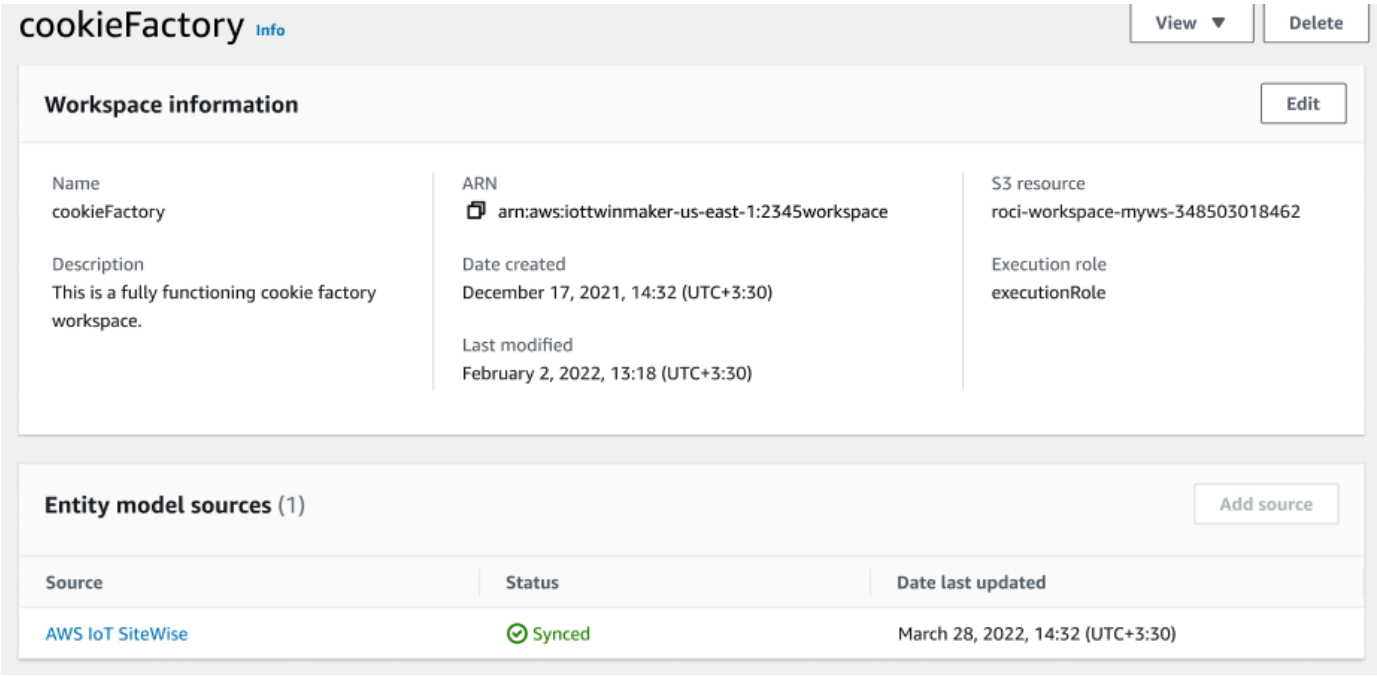
3. Elija Connect workspace para vincular su AWS IoT TwinMaker espacio de trabajo a sus AWS IoT SiteWise activos.

> **Note**
>
> Solo puedes usar la sincronización de activos con un único AWS IoT TwinMaker espacio de trabajo. Debe desconectar la sincronización de un espacio de trabajo y conectarse a otro si desea sincronizar otro espacio de trabajo.

4. A continuación, vaya al espacio de trabajo en el que desea usar la sincronización de activos.
5. Seleccione Add sources (Agregar orígenes). Se abrirá la página Agregar un origen de modelo de la entidad.



6. En la página Agregar un origen del modelo de la entidad, confirme que en el campo origen se muestre AWS IoT SiteWise. Seleccione el rol de IAM que creó como requisito previo para el rol de IAM.
7. Ya has activado la sincronización AWS IoT SiteWise de activos. Debería ver un banner de confirmación en la parte superior de la página del Espacio de trabajo seleccionado para confirmar que la sincronización de activos está activa. También debería ver el origen de la sincronización en la sección Orígenes del modelo de la entidad.



Uso del Io TSite WiseDefaultWorkspace

Al optar por la AWS IoT SiteWiseAWS IoT TwinMaker integración, se crea un espacio de trabajo predeterminado denominado y `IoTSiteWiseDefaultWorkspace` se sincroniza automáticamente con AWS IoT SiteWiseél.

También puedes usar la AWS IoT TwinMaker `CreateWorkspace` API para crear un espacio de trabajo con el nombre`IoTSiteWiseDefaultWorkspace`.

Requisitos previos

Antes de crear`IoTSiteWiseDefaultWorkspace`, asegúrate de haber hecho lo siguiente:

- Cree un rol AWS IoT TwinMaker vinculado a un servicio. Para obtener más información, consulte Cómo utilizar roles vinculados a servicios de AWS IoT TwinMaker.
- Abra la consola de IAM en https://console.aws.amazon.com/iam/.

  Revise el rol o el usuario y compruebe que tiene permiso para hacerlo. `iotsitewise:EnableSiteWiseIntegration`

  Si es necesario, añada permiso al rol o al usuario:

  JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "iotsitewise:EnableSiteWiseIntegration",
            "Resource": "*"
        }
    ]
}
```

Diferencias entre los espacios de trabajo personalizados y predeterminados

> **Important**
>
> AWS IoT SiteWise Las nuevas funciones, como `CompositionModel`, solo están disponibles en`IoTSiteWiseDefaultWorkspace`. Le recomendamos que utilice un espacio de trabajo predeterminado en lugar de un espacio de trabajo personalizado.

Al usar el`IoTSiteWiseDefaultWorkspace`, existen algunas diferencias notables con respecto al uso de un espacio de trabajo personalizado con sincronización de activos.

- Al crear un espacio de trabajo predeterminado, la ubicación de Amazon S3 y el rol de IAM son opcionales.

  > **Note**
  >
  > Puede utilizarlos `UpdateWorkspace` para proporcionar la ubicación de Amazon S3 y la función de IAM.

- `IoTSiteWiseDefaultWorkspace`No tiene un límite de recuento de recursos con el que sincronizar AWS IoT SiteWise los recursos. AWS IoT TwinMaker
- Cuando sincronices los recursos desde AWS IoT SiteWise, `SyncSource` serán los suyos`SITEWISE_MANAGED`. Esto incluye `Entities` y`ComponentTypes`.
- Nuevas AWS IoT SiteWise funciones, como `CompositionModel` las que solo están disponibles en`IoTSiteWiseDefaultWorkspace`.

Existen algunas limitaciones específicas`IoTSiteWiseDefaultWorkspace`, que son:

- El espacio de trabajo predeterminado no se puede eliminar.
- Para eliminar recursos, primero debe eliminar los AWS IoT SiteWise recursos y, a continuación, AWS IoT TwinMaker se eliminan los recursos correspondientes.

Recursos sincronizados desde AWS IoT SiteWise

En este tema, se enumeran los activos desde los que puedes sincronizar AWS IoT SiteWise con tu AWS IoT TwinMaker espacio de trabajo.

> **Important**
>
> Consulte Diferencias entre los espacios de trabajo personalizados y predeterminados para obtener información sobre las diferencias entre los espacios de trabajo personalizados y predeterminados.

Espacios de trabajo personalizados y predeterminados

Los siguientes recursos están sincronizados y disponibles en los espacios de trabajo personalizados y predeterminados:

Modelos de activos

AWS IoT TwinMaker crea un nuevo tipo de componente para cada modelo de activo en. AWS IoT SiteWise

- El componente `TypeId` del modelo de activos utilizará uno de los siguientes patrones:
  - Espacio de trabajo personalizado - `iotsitewise.assetmodel:`*`assetModelId`*
  - Espacio de trabajo predeterminado - *`assetModelId`*
- Cada propiedad del modelo de activos es una propiedad nueva en el tipo de componente, con uno de los siguientes patrones de nomenclatura:
  - Espacio de trabajo personalizado - `Property_`*`propertyId`*
  - Espacio de trabajo predeterminado - *`propertyId`*

  El nombre de la propiedad en AWS IoT SiteWise se guarda tal como aparece `displayName` en la definición de la propiedad.
- Cada jerarquía del modelo de activos es una nueva propiedad de tipo `LIST` y `nestedType` se encuentra `RELATIONSHIP` en el tipo de componente. La jerarquía se asigna a la propiedad con un nombre precedido por una de las siguientes opciones:
  - Espacio de trabajo personalizado - *`Hierarchy_hierarchyId`*
  - Espacio de trabajo predeterminado - *`hierarchyId`*

Activo

AWS IoT TwinMaker crea una nueva entidad para cada activo de AWS IoT SiteWise.

- El `entityId` es el mismo que el `assetId` de entrada AWS IoT SiteWise.
- Estas entidades tienen un único componente denominado `sitewiseBase`, que tiene el tipo de componente correspondiente al modelo de activo para este activo.
- Cualquier anulación con respecto a los activos, como la configuración del alias o de la unidad de medida de la propiedad, se refleja en la entidad en AWS IoT TwinMaker.

Solo espacio de trabajo predeterminado

Los siguientes activos están sincronizados y disponibles solo en el espacio de trabajo predeterminado,`IoTSiteWiseDefaultWorkspace`.

AssetModelComponents

AWS IoT TwinMaker crea un nuevo tipo de componente para cada pulgada`AssetModelComponents`. AWS IoT SiteWise

- El componente `TypeId` del modelo de activos utiliza el siguiente patrón:`assetModelId`.
- Cada propiedad del modelo de activos es una propiedad nueva en el tipo de componente, con el nombre de la propiedad como `propertyId`. El nombre de la propiedad en AWS IoT SiteWise se almacena como `displayName` en la definición de la propiedad.
- Cada jerarquía del modelo de activos es una nueva propiedad de tipo `LIST` y `nestedType` se encuentra `RELATIONSHIP` en el tipo de componente. La jerarquía se asigna a la propiedad con un nombre con el prefijo `hierarchyId`.

AssetModelCompositeModel

AWS IoT TwinMaker crea un nuevo tipo de componente para cada `AssetModelCompositeModel` pulgada AWS IoT SiteWise.

- El componente `TypeId` del modelo de activos utiliza el siguiente patrón:`assetModelId_assetModelCompositeModelId`.
- Cada propiedad del modelo de activos es una propiedad nueva en el tipo de componente, con el nombre de la propiedad como `propertyId`. El nombre de la propiedad en AWS IoT SiteWise se almacena como `displayName` en la definición de la propiedad.

AssetCompositeModels

AWS IoT TwinMaker crea un nuevo componente compuesto para cada `AssetCompositeModel` pulgada AWS IoT SiteWise.

- El `componentName` es el mismo que `assetModelCompositeModelId` el interior AWS IoT SiteWise.

Los recursos no están sincronizados

Los siguientes recursos no están sincronizados:

Activos y modelos de activos no sincronizados

- Los modelos de alarma se sincronizarán como CompositeModels, pero los datos correspondientes del activo relacionados con las alarmas no se sincronizarán.
- AWS IoT SiteWise los flujos de datos no están sincronizados. Solo se sincronizan las propiedades modeladas en el modelo de activos.
- Los valores de propiedad para el cálculo de los atributos, las medidas, las transformaciones, los agregados y los metadatos, como la fórmula y el periodo, no se sincronizan. Solo se sincronizan los metadatos de las propiedades, como el alias, la unidad de medida y el tipo de datos. Los valores se pueden consultar mediante la API del conector de AWS IoT TwinMaker datos normal,. GetPropertyValueHistory

Utilice entidades sincronizadas y tipos de componentes en AWS IoT TwinMaker

Una vez sincronizados los activos AWS IoT SiteWise, los tipos de componentes sincronizados son de solo lectura. AWS IoT TwinMaker Cualquier acción de actualización o eliminación debe realizarse y esos cambios se sincronizan AWS IoT TwinMaker si el SyncJob sigue activo. AWS IoT SiteWise

Las entidades sincronizadas y el componente AWS IoT SiteWise base también son de solo lectura. AWS IoT TwinMaker Puede agregar componentes no sincronizados adicionales a la entidad sincronizada, siempre que no se actualice ningún atributo a nivel de la entidad, como la descripción o `entityName`.

Se aplican algunas restricciones a la forma de interactuar con las entidades sincronizadas. No se pueden crear entidades secundarias en una entidad sincronizada de la jerarquía de la entidad

sincronizada. Además, no puede crear tipos de componentes no sincronizados que se extiendan desde un tipo de componente sincronizado.

> **Note**
>
> Los componentes adicionales se eliminan junto con la entidad si se elimina el activo en el trabajo de sincronización AWS IoT SiteWise o si se elimina el mismo.

Puede usar estas entidades sincronizadas en los paneles de Grafana y agregarlas como etiquetas en el compositor de escenas como entidades normales. También puede hacer consultas de gráficos de conocimiento para estas entidades sincronizadas.

> **Note**
>
> Las entidades sincronizadas sin modificación son sin cargo, pero sí tendrán un costo si se las modificó en AWS IoT TwinMaker. Por ejemplo, si añades un componente no sincronizado a una entidad sincronizada, esa entidad ahora se carga. AWS IoT TwinMaker Para más información, consulte Precios de AWS IoT TwinMaker.

Analice el estado y los errores de la sincronización

En este tema se proporciona orientación sobre cómo analizar los errores y los estados de sincronización.

> **Important**
>
> Consulte the section called “Diferencias entre los espacios de trabajo personalizados y predeterminados” para obtener información sobre las diferencias entre los espacios de trabajo personalizados y predeterminados.

Estados de los trabajos de sincronización

Un trabajo de sincronización tiene uno de los siguientes estados, según su estado.

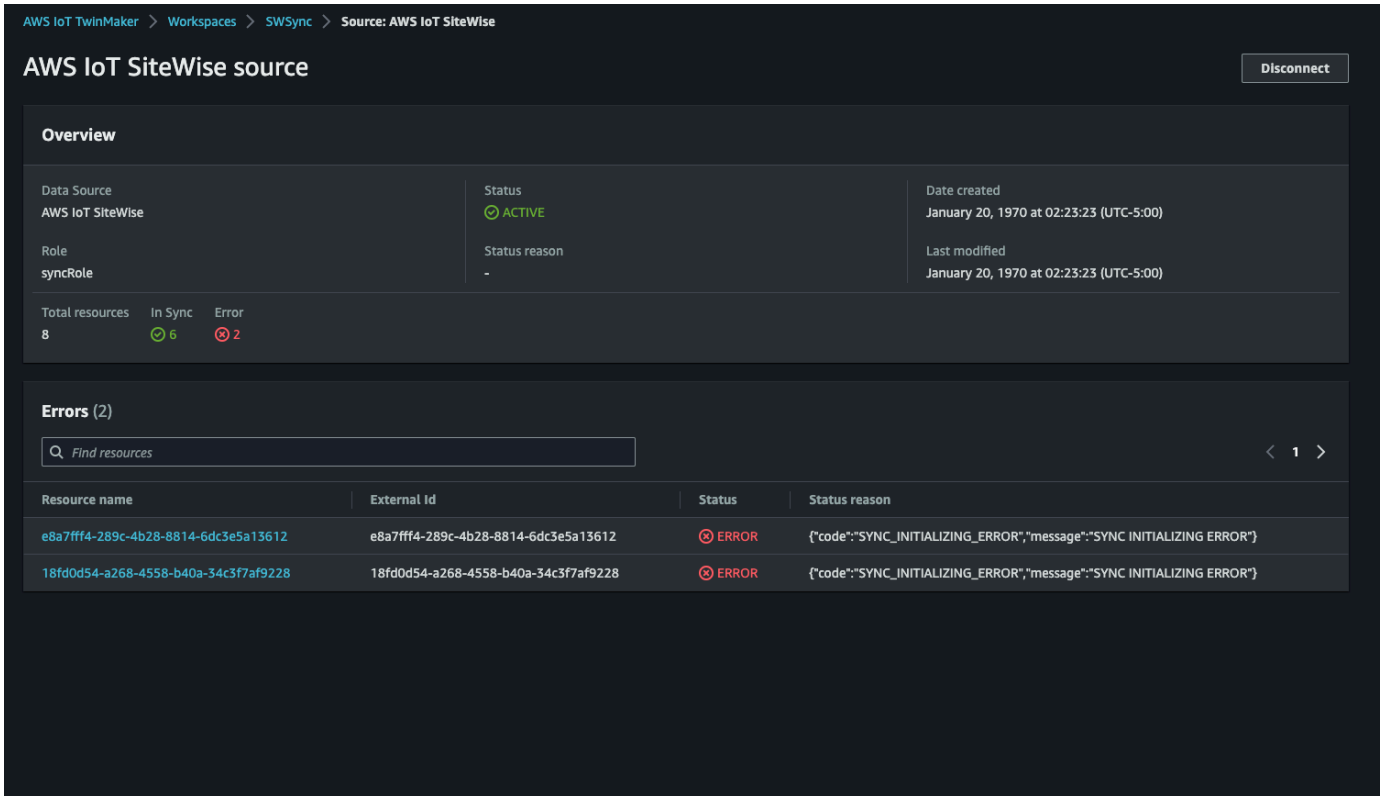
- El `CREATING` estado del trabajo de sincronización significa que el trabajo está comprobando los permisos y cargando datos AWS IoT SiteWise para preparar la sincronización.

- El `INITIALIZING` estado del trabajo de sincronización significa que todos los recursos existentes AWS IoT SiteWise están sincronizados. AWS IoT TwinMaker Este paso puede tardar más en completarse si el usuario tiene una gran cantidad de activos y modelos de activos en AWS IoT SiteWise. Puede supervisar la cantidad de recursos que se sincronizaron comprobando el trabajo de sincronización en la consola de AWS IoT TwinMaker o llamando a la API de `ListSyncResources`.
- El estado del trabajo de sincronización `ACTIVE` significa que se completó el paso de inicialización. El trabajo ya está listo para sincronizar cualquier actualización nueva desde AWS IoT SiteWise.
- El estado del trabajo de sincronización `ERROR` indica un error en alguno de los estados anteriores. Revise el mensaje de error. Es posible que haya un problema con la configuración del rol de IAM. Si desea utilizar una nueva función de IAM, elimine la tarea de sincronización en la que se produjo el error y cree una nueva con la nueva función.

Los errores de sincronización aparecen en la página de origen del modelo, a la que se accede desde la tabla de orígenes del modelo de la entidad del espacio de trabajo. La página de origen del modelo muestra una lista de los recursos que no se pudieron sincronizar. El trabajo de sincronización vuelve a intentar con la mayoría de los errores automáticamente, pero si el recurso requiere una acción, entonces permanece en ese estado de `ERROR`. También puedes obtener una lista de errores mediante la ListSyncResourcesAPI.

Para ver todos los errores del origen actual, lleve a cabo el siguiente procedimiento.

1. Navegue hasta el espacio de trabajo en la consola de AWS IoT TwinMaker.
2. Seleccione la AWS IoT SiteWise fuente que aparece en el modal de fuentes del modelo de entidad para abrir la página de detalles de la sincronización de activos.



3. Como se muestra en la captura de pantalla anterior, todos los recursos con errores persistentes se enumeran en la tabla de Errores. Puede usar esta tabla para localizar y corregir errores relacionados con recursos específicos.

Entre los errores posibles, se incluyen:

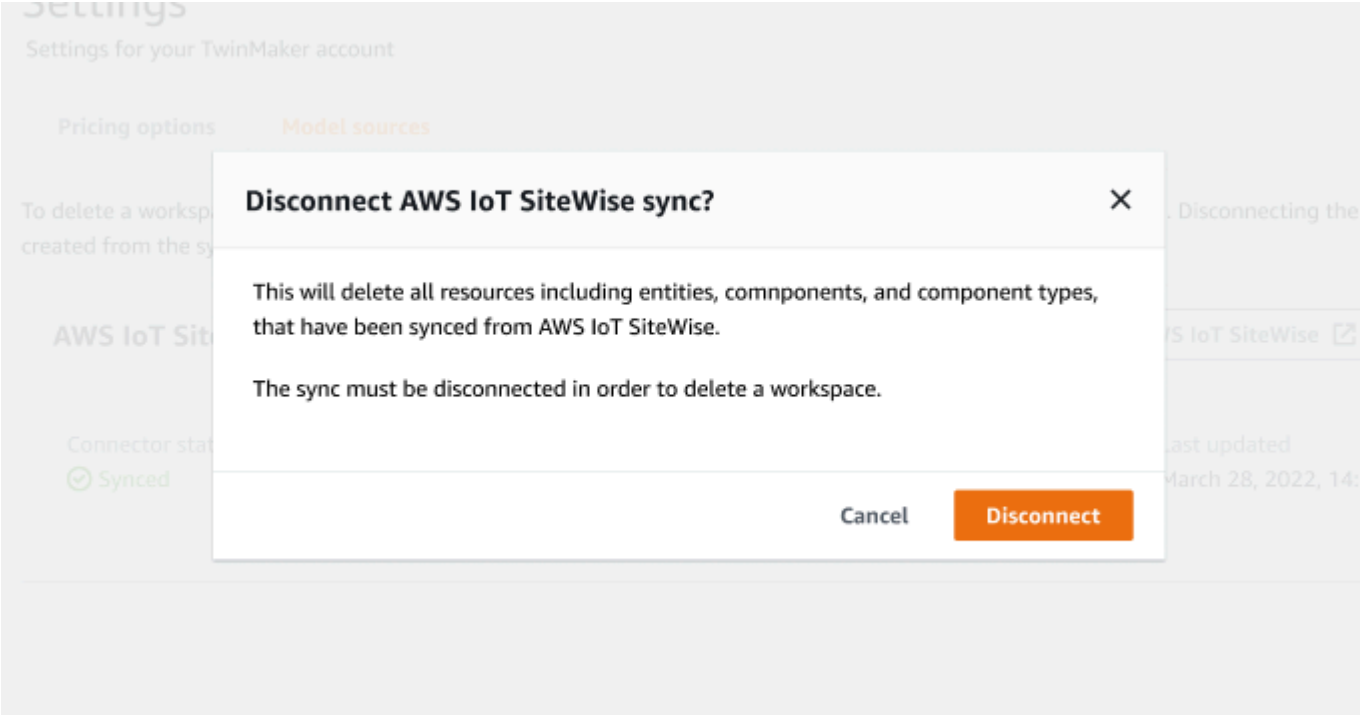
- Si bien AWS IoT SiteWise admite nombres de activos duplicados, AWS IoT TwinMaker solo los admite a R00T nivel, no bajo la misma entidad principal. Si tiene dos activos con el mismo nombre en una entidad principal AWS IoT SiteWise, uno de ellos no se sincronizará. Para corregir este error, elimine uno de los activos o mueva uno a otro activo principal AWS IoT SiteWise antes de sincronizarlo.
- Si ya tienes una entidad con el mismo ID que el ID del AWS IoT SiteWise activo, ese activo no se sincronizará hasta que elimines la entidad existente.

Eliminar un trabajo de sincronización

Utilice el siguiente procedimiento para eliminar un trabajo de sincronización.

> **Important**
>
> Consulte the section called “Diferencias entre los espacios de trabajo personalizados y predeterminados” para obtener información sobre las diferencias entre los espacios de trabajo personalizados y predeterminados.

1. Vaya a la consola de AWS IoT TwinMaker.
2. Abra el espacio de trabajo del que desee eliminar el trabajo de sincronización.
3. En Orígenes de modelos de la entidad, seleccione el origen de AWS IoT SiteWise para abrir la página de detalles del origen.
4. Para detener el trabajo de sincronización, elija Desconectar. Confirme su elección para eliminar completamente el trabajo de sincronización.



Una vez que se elimina un trabajo de sincronización, puede volver a crearlo en el mismo espacio de trabajo o en uno diferente.

No puedes eliminar un espacio de trabajo si hay trabajos de sincronización en ese espacio de trabajo. Elimine primero los trabajos de sincronización antes de eliminar un espacio de trabajo.

Si se produce algún error durante la eliminación del trabajo de sincronización, el trabajo de sincronización permanece en ese estado de `DELETING` y se vuelve a intentar automáticamente. Ahora puede eliminar manualmente cualquiera de las entidades o tipos de componentes sincronizados si se produce algún error relacionado con la eliminación de un recurso.

> **Note**
>
> Todos los recursos desde los que se sincronizaron AWS IoT SiteWise se eliminan primero y, a continuación, se elimina el trabajo de sincronización propiamente dicho.

Límites de la sincronización de activos

> **Important**
>
> Consulte the section called “Diferencias entre los espacios de trabajo personalizados y predeterminados” para obtener información sobre las diferencias entre el espacio de trabajo personalizado y el predeterminado.

Debido a que las cuotas de AWS IoT SiteWise son superiores a las cuotas predeterminadas de AWS IoT TwinMaker, vamos a aumentar los siguientes límites para las entidades y los tipos de componentes sincronizados desde AWS IoT SiteWise.

- 1000 tipos de componentes sincronizados en un espacio de trabajo, ya que solo se pueden sincronizar 1000 modelos de activos. AWS IoT SiteWise
- 100 000 entidades sincronizadas en un espacio de trabajo, ya que solo puede sincronizar 100 000 activos desde él. AWS IoT SiteWise
- 2000 entidades secundarias como máximo por entidad principal. Sincroniza 2000 activos secundarios por activo principal único.

> **Note**
>
> La GetEntityAPI solo devuelve las 50 primeras entidades secundarias de una propiedad jerárquica, pero puedes utilizarla para paginar y recuperar la lista de todas las entidades secundarias. GetPropertyValue

- 600 propiedades por componente sincronizado desde AWS IoT SiteWise, que permiten sincronizar modelos de activos con un total de 600 propiedades y jerarquías.

> **Note**
>
> Estos límites solo se aplican a las entidades sincronizadas. Solicite un aumento de la cuota si necesita aumentar estos límites para los recursos no sincronizados.

Integración del panel de control de AWS IoT TwinMaker Grafana

AWS IoT TwinMaker admite la integración de Grafana a través de un complemento de aplicación. Utilice Grafana 10.4.0 y versiones posteriores para interactuar con su aplicación gemela digital. El AWS IoT TwinMaker complemento proporciona paneles personalizados, plantillas de paneles y una fuente de datos para conectarse a los datos de su gemelo digital.

Para obtener más información sobre cómo incorporar Grafana y configurar permisos para su panel de control, consulte los siguientes temas:

Temas

- Configuración CORS para el visor de escenas de Grafana
- Configuración del entorno de Grafana
- Creación de un rol de IAM en el panel
- Crear una política AWS IoT TwinMaker de reproducción de vídeo

> **Note**
>
> Debe modificar la configuración CORS (intercambio de recursos de origen cruzado) del bucket de Amazon S3 para permitir que la interfaz de usuario de Grafana cargue recursos del bucket. Para obtener las instrucciones, consulte Configuración CORS para el visor de escenas de Grafana.

Para obtener más información sobre el complemento AWS IoT TwinMaker Grafana, consulte la documentación de la AWS IoT TwinMaker aplicación.

Para obtener más información sobre los componentes clave del complemento Grafana, consulte lo siguiente:

- AWS IoT TwinMaker origen de datos (objeto)
- Plantillas de paneles
- Panel del visor de escenas

- Panel reproductor de vídeo

Configuración CORS para el visor de escenas de Grafana

El complemento AWS IoT TwinMaker Grafana requiere una configuración CORS (intercambio de recursos de origen cruzado), que permite que la interfaz de usuario de Grafana cargue recursos del bucket de Amazon S3. Sin la configuración CORS, recibirá un mensaje de error con el mensaje «Error al cargar la escena 3D por un error de red» en el visor de escenas, ya que el dominio de Grafana no puede acceder a los recursos del bucket de Amazon S3.

Para configurar su bucket de Amazon S3 con CORS, siga estos pasos:

1. Inicie sesión en la consola de IAM y abra la consola de Amazon S3.
2. En la lista de cubos, elige el nombre del depósito que utilizas como depósito de recursos de tu AWS IoT TwinMaker espacio de trabajo.
3. Elija Permisos.
4. En la sección Compartir recursos entre orígenes [CORS], elija Editar para abrir el editor de CORS.
5. En el cuadro de texto del editor de configuración CORS, escriba o copie y pegue la siguiente configuración JSON CORS sustituyendo el dominio *GRAFANA-WORKSPACE-DOMAIN* del espacio de trabajo de Grafana por el suyo.

> **Note**
>
> Debe mantener el asterisco al principio * del elemento JSON. `"AllowedOrigins":`

```
    [
{
    "AllowedHeaders": [
        "*"
    ],
    "AllowedMethods": [
        "GET",
        "PUT",
        "POST",
        "DELETE",
```

```
            "HEAD"
        ],
        "AllowedOrigins": [
            "*GRAFANA-WORKSPACE-DOMAIN"
        ],
        "ExposeHeaders": [
            "ETag"
        ]
    }
]
```

6. Seleccione Guardar cambios para finalizar la configuración de CORS.

Para obtener más información sobre CORS con buckets de Amazon S3, consulte Uso del uso compartido de recursos entre orígenes (CORS).

Configuración del entorno de Grafana

Puede utilizar Amazon Managed Grafana para un servicio totalmente gestionado o configurar un entorno de Grafana que gestione usted mismo. Con Amazon Managed Grafana, puede implementar, operar y escalar Grafana de código abierto rápidamente según sus necesidades. Si lo desea, también puede configurar su propia infraestructura para administrar servidores Grafana.

Para obtener más información sobre las dos opciones de entorno de Grafana, consulte los siguientes temas:

- Amazon Managed Grafana
- Grafana autoadministrado

Amazon Managed Grafana

Amazon Managed Grafana proporciona un AWS IoT TwinMaker complemento para que puedas integrarte rápidamente con AWS IoT TwinMaker Grafana. Como Amazon Managed Grafana administra los servidores Grafana por usted, puede visualizar sus datos sin tener que crear, empaquetar o implementar ningún hardware o cualquier otra infraestructura de Grafana. Para obtener más información sobre Amazon Managed Grafana, consulte ¿Qué es Amazon Managed Grafana?.

> **Note**
>
> Actualmente, Amazon Managed Grafana es compatible con la versión 1.3.1 del complemento AWS IoT TwinMaker Grafana.

Requisitos previos de Amazon Managed Grafana

Para usarlo AWS IoT TwinMaker en un panel de control de Grafana gestionado por Amazon, primero complete el siguiente requisito previo:

- Crea un AWS IoT TwinMaker espacio de trabajo. Para obtener más información sobre la creación de espacios de trabajo, consulte Primeros pasos con AWS IoT TwinMaker.

> **Note**
>
> La primera vez que creas un espacio de trabajo de Grafana gestionado por Amazon en la consola AWS de administración, AWS IoT TwinMaker no aparece en la lista. Sin embargo, el complemento ya está instalado en todos los espacios de trabajo. Puedes encontrar el complemento de AWS IoT TwinMaker en la lista de complementos de Grafana de código abierto. Para encontrar la fuente de datos AWS IoT TwinMaker , seleccione Añadir una fuente de datos en la página Fuentes de datos.

Al crear un espacio de trabajo de Amazon Managed Grafana, se crea automáticamente un rol de IAM para gestionar los permisos de la instancia de Grafana. Esto se denomina rol de IAM en el espacio de trabajo. Es la opción del proveedor de autenticación que utilizará para configurar todas las fuentes de datos de AWS IoT TwinMaker para Grafana. Amazon Managed Grafana no admite la adición automática de permisos para AWS IoT TwinMaker, por lo que debe configurar estos permisos manualmente. Para obtener más información acerca de cómo establecer los permisos manuales, consulte Creación de un rol de IAM en el panel.

Grafana autoadministrado

Puede optar por alojar su propia infraestructura para ejecutar Grafana. Para obtener información sobre cómo ejecutar Grafana localmente en su máquina, consulte Instalar Grafana. El complemento AWS IoT TwinMaker está disponible en el catálogo público de Grafana. Para obtener información

sobre la instalación de este complemento en su entorno de Grafana, consulte Aplicación AWS IoT TwinMaker.

Cuando ejecuta Grafana de forma local, no puede compartir fácilmente los paneles ni proporcionar acceso a varios usuarios. Para obtener una guía de inicio rápido con scripts sobre cómo compartir paneles con Grafana local, consulte el repositorio de ejemplos de AWS IoT TwinMaker. Este recurso explica cómo alojar un entorno de Grafana en Cloud9 y EC2 Amazon en un punto final público.

Debe determinar qué proveedor de autenticación utilizará para configurar TwinMaker las fuentes de datos. Las credenciales del entorno se configuran en función de la cadena de credenciales predeterminada (consulte Uso de la cadena de proveedores de credenciales predeterminada). Las credenciales predeterminadas pueden ser las credenciales permanentes de cualquier usuario o rol. Por ejemplo, si utilizas Grafana en Amazon EC2 , la cadena de credenciales predeterminada tiene acceso al rol de EC2 ejecución de Amazon, que entonces sería tu proveedor de autenticación. El Nombre de recurso de Amazon (ARN) de IAM del proveedor de autenticación es obligatorio en los pasos para Creación de un rol de IAM en el panel.

Creación de un rol de IAM en el panel

Con AWS IoT TwinMaker, puede controlar el acceso a los datos en sus paneles de Grafana. Los usuarios del panel de control de Grafana deben tener diferentes ámbitos de permiso para ver los datos y, en algunos casos, escribirlos. Por ejemplo, es posible que un operador de alarma no tenga permiso para ver vídeos, mientras que un administrador sí lo tiene para todos los recursos. Grafana define los permisos a través de fuentes de datos, donde se proporcionan las credenciales y un rol de IAM. La AWS IoT TwinMaker fuente de datos obtiene las AWS credenciales con permisos para ese rol. Si no se proporciona una función de IAM, Grafana utiliza el alcance de las credenciales, que no se puede reducir en nada. AWS IoT TwinMaker

Para usar sus AWS IoT TwinMaker paneles de control en Grafana, debe crear un rol de IAM y adjuntar políticas. Puede utilizar las siguientes plantillas para crear estas políticas.

Creación de una política de IAM

Cree una política de IAM llamada *YourWorkspaceId*`DashboardPolicy` en la consola de IAM. Esta política proporciona a sus espacios de trabajo acceso al bucket de Amazon S3 y a los recursos de AWS IoT TwinMaker . También puede optar por utilizar AWS IoT Greengrass Edge Connector para Amazon Kinesis Video Streams, que requiere permisos para las transmisiones de vídeo de Kinesis AWS IoT SiteWise y los activos configurados para el componente. Para adaptarse a su caso de uso, elija una de las siguientes plantillas de políticas.

1. Sin política de permisos de vídeo

   Si no quiere usar el panel del Reproductor de vídeo de Grafana, cree la política con la siguiente plantilla.

   JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject"
      ],
      "Resource": [
        "arn:aws:s3:::bucketName/*",
        "arn:aws:s3:::bucketName"
        ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "iottwinmaker:Get*",
        "iottwinmaker:List*"
      ],
      "Resource": [
        "arn:aws:iottwinmaker:region:accountId:workspace/workspaceId",
        "arn:aws:iottwinmaker:region:accountId:workspace/workspaceId/*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": "iottwinmaker:ListWorkspaces",
      "Resource": "*"
    }
  ]
}
```

   Se crea un bucket de Amazon S3 para cada espacio de trabajo. Contiene los modelos 3D y las escenas que se pueden ver en un panel de control. El SceneViewerpanel carga los elementos de este depósito.

2. Política de permisos de vídeo restringida

   Para limitar el acceso al panel del reproductor de vídeo de Grafana, agrupe los recursos de AWS IoT Greengrass Edge Connector para Amazon Kinesis Video Streams por etiquetas. Para obtener más información acerca de la administración de permisos para los recursos de vídeo, consulte Crear una política AWS IoT TwinMaker de reproducción de vídeo.

3. Todos los permisos de vídeo

   Si no desea agrupar sus vídeos, puede hacer que todos sean accesibles desde el Reproductor de Vídeo de Grafana. Cualquier persona con acceso a un espacio de trabajo de Grafana puede reproducir vídeo para cualquier transmisión de tu cuenta y tener acceso de solo lectura a cualquier AWS IoT SiteWise activo. Esto incluye cualquier recurso que se cree en el futuro.

   Cree la política con la siguiente plantilla:

   JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject"
      ],
      "Resource": [
        "arn:aws:s3:::bucketName/*",
        "arn:aws:s3:::bucketName"
        ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "iottwinmaker:Get*",
        "iottwinmaker:List*"
      ],
      "Resource": [
        "arn:aws:iottwinmaker:region:accountId:workspace/workspaceId",
        "arn:aws:iottwinmaker:region:accountId:workspace/workspaceId/*"
      ]
    },
    {
```

```
      "Effect": "Allow",
      "Action": "iottwinmaker:ListWorkspaces",
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "kinesisvideo:GetDataEndpoint",
        "kinesisvideo:GetHLSStreamingSessionURL"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "iotsitewise:GetAssetPropertyValue",
        "iotsitewise:GetInterpolatedAssetPropertyValues"
      ],
      "Resource": "*"
    },
    {
       "Effect": "Allow",
       "Action": [
        "iotsitewise:BatchPutAssetPropertyValue"
      ],
      "Resource": "*",
      "Condition": {
        "StringLike": {
          "aws:ResourceTag/EdgeConnectorForKVS": "*workspaceId*"
        }
      }
    }
  ]
}
```

La siguiente política de ejemplo proporciona los siguientes permisos:

- Acceso de solo lectura a un bucket de S3 para cargar una escena.
- Acceso de solo lectura AWS IoT TwinMaker para todas las entidades y componentes de un espacio de trabajo.
- Acceso de solo lectura para reproducir todos los vídeos de Kinesis Video Streams de su cuenta.

- Acceso de solo lectura al historial del valor de las propiedades de todos AWS IoT SiteWise los activos de su cuenta.
- Incorporación de datos a cualquier propiedad de un AWS IoT SiteWise activo etiquetado con la clave `EdgeConnectorForKVS` y el valor`workspaceId`.

Etiquetar el AWS IoT SiteWise recurso de la cámara: solicite la carga de vídeo desde Edge.

Con el reproductor de vídeo de Grafana, los usuarios pueden solicitar manualmente que el vídeo se cargue desde la memoria caché perimetral a Kinesis Video Streams. Puede activar esta función para cualquier AWS IoT SiteWise activo que esté asociado a su AWS IoT Greengrass Edge Connector para Amazon Kinesis Video Streams y que esté etiquetado con la `EdgeConnectorForKVS` clave.

El valor de la etiqueta puede ser una lista de ID de espacio de trabajo delimitada por cualquiera de los siguientes caracteres: `. : + = @ _ / -`. Por ejemplo, si desea utilizar un AWS IoT SiteWise activo asociado a un conector AWS IoT Greengrass Edge para las transmisiones de Amazon Kinesis Video Streams AWS IoT TwinMaker en todos los espacios de trabajo, puede utilizar una etiqueta que siga este patrón:. `WorkspaceA/WorkspaceB/WorkspaceC` El complemento Grafana exige que el AWS IoT TwinMaker WorkspaceID se utilice para agrupar la ingesta de datos de activos. AWS IoT SiteWise

Añade más permisos a la política de su panel de control

El complemento AWS IoT TwinMaker Grafana usa tu proveedor de autenticación para invocar AssumeRole el rol de panel de control que crees. Internamente, el complemento restringe el mayor número de permisos a los que tienes acceso mediante una política de sesión en la AssumeRole llamada. Para obtener más información sobre las políticas de sesión, consulte Políticas de sesión.

Esta es la política máxima permisiva que puede tener en su rol de panel de control para un espacio de trabajo de AWS IoT TwinMaker :

JSON

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

```
      "Action": [
        "s3:GetObject"
      ],
      "Resource": [
        "arn:aws:s3:::bucketName/*",
        "arn:aws:s3:::bucketName"
        ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "iottwinmaker:Get*",
        "iottwinmaker:List*"
      ],
      "Resource": [
        "arn:aws:iottwinmaker:region:accountId:workspace/workspaceId",
        "arn:aws:iottwinmaker:region:accountId:workspace/workspaceId/*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": "iottwinmaker:ListWorkspaces",
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "kinesisvideo:GetDataEndpoint",
        "kinesisvideo:GetHLSStreamingSessionURL"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "iotsitewise:GetAssetPropertyValue",
        "iotsitewise:GetInterpolatedAssetPropertyValues"
      ],
      "Resource": "*"
    },
    {
       "Effect": "Allow",
       "Action": [
        "iotsitewise:BatchPutAssetPropertyValue"
```

```
      ],
      "Resource": "*",
      "Condition": {
        "StringLike": {
          "aws:ResourceTag/EdgeConnectorForKVS": "*workspaceId*"
        }
      }
    }
  ]
}
```

Si añades instrucciones con `Allow` más permisos, no funcionarán en el AWS IoT TwinMaker plugin. Esto está diseñado para garantizar que el complemento utilice los permisos mínimos necesarios.

Sin embargo, puede reducir aún más los permisos. Para obtener información, consulte Crear una política AWS IoT TwinMaker de reproducción de vídeo.

Creación del rol de IAM del panel de control de Grafana

Cree un rol de IAM llamado *YourWorkspaceId*`DashboardRole` en la consola de IAM. Adjunte el *YourWorkspaceId*`DashboardPolicy` al rol.

Para editar la política de confianza del rol del panel de control, debe dar permiso al proveedor de autenticación de Grafana para que llame a `AssumeRole` en el rol del panel de control. Actualice la política de confianza con la siguiente plantilla:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "ARN of Grafana authentication provider"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Para obtener más información sobre cómo crear un entorno de Grafana y encontrar su proveedor de autenticación, consulte Configuración del entorno de Grafana.

Crear una política AWS IoT TwinMaker de reproducción de vídeo

La siguiente es una plantilla de política con todos los permisos de vídeo que necesitas para el complemento AWS IoT TwinMaker en Grafana:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject"
      ],
      "Resource": [
        "arn:aws:s3:::bucketName/*",
        "arn:aws:s3:::bucketName"
        ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "iottwinmaker:Get*",
        "iottwinmaker:List*"
      ],
      "Resource": [
        "arn:aws:iottwinmaker:region:accountId:workspace/workspaceId",
        "arn:aws:iottwinmaker:region:accountId:workspace/workspaceId/*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": "iottwinmaker:ListWorkspaces",
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
```

```
        "kinesisvideo:GetDataEndpoint",
        "kinesisvideo:GetHLSStreamingSessionURL"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "iotsitewise:GetAssetPropertyValue",
        "iotsitewise:GetInterpolatedAssetPropertyValues"
      ],
      "Resource": "*"
    },
    {
       "Effect": "Allow",
       "Action": [
        "iotsitewise:BatchPutAssetPropertyValue"
      ],
      "Resource": "*",
      "Condition": {
        "StringLike": {
          "aws:ResourceTag/EdgeConnectorForKVS": "*workspaceId*"
        }
      }
    }
  ]
}
```

Para obtener más información sobre la política completa, consulte la plantilla de política de todos los permisos de vídeo en el tema Creación de una política de IAM.

Limite el acceso a sus recursos

El panel del reproductor de vídeo de Grafana utiliza directamente Kinesis Video Streams e IoT SiteWise para ofrecer una experiencia de reproducción de vídeo completa. Para evitar el acceso no autorizado a recursos que no están asociados a tu AWS IoT TwinMaker espacio de trabajo, añade condiciones a la política de IAM para tu función en el panel de control del espacio de trabajo.

Limite permisos GET

Puede limitar el acceso a sus activos AWS IoT SiteWise y a sus transmisiones de Amazon Kinesis Video Streams etiquetando los recursos. Es posible que ya haya etiquetado el recurso de su AWS IoT SiteWise cámara con el AWS IoT TwinMaker WorkspaceID para habilitar la función de solicitud de carga de vídeos; consulte el tema Cargar vídeo desde la periferia. Puede usar el mismo par clave-valor de etiqueta para limitar el acceso GET a AWS IoT SiteWise los activos y también para etiquetar sus transmisiones de Kinesis Video Streams de la misma manera.

A continuación, puede añadir esta condición a las instrucciones kinesisvideo e iotsitewise en *YourWorkspaceId*DashboardPolicy:

```
"Condition": {
  "StringLike": {
    "aws:ResourceTag/EdgeConnectorForKVS": "*workspaceId*"
  }
}
```

Caso de uso real: agrupar cámaras

En este escenario, tiene una gran variedad de cámaras que monitorean el proceso de cocción de las galletas en una fábrica. Los lotes de masa para galletas se hacen en la sala de rebozado, la masa se congela en la sala del congelador y las galletas se hornean en la sala de horneado. Hay cámaras en cada una de estas salas con diferentes equipos de operadores que monitorean cada proceso por separado. Desea que cada grupo de operadores esté autorizado para su sala respectiva. Al construir un gemelo digital para la fábrica de galletas, se utiliza un único espacio de trabajo, pero los permisos de la cámara deben determinarse por habitación.

Puede lograr esta separación de permisos etiquetando grupos de cámaras en función de su ID de agrupación. En este escenario, los ID de agrupación son, y. BatterRoom FreezerRoom BakingRoom La cámara de cada habitación está conectada a Kinesis Video Streams y debe tener una etiqueta con: Key = `EdgeConnectorForKVS`, Value = `BatterRoom`. El valor puede ser una lista de agrupamientos delimitados por cualquiera de los siguientes caracteres: `. : + = @ _ / -`

Para modificar el *YourWorkspaceId*`DashboardPolicy`, utilice las siguientes declaraciones de política:

```
...,
{
```

```
  "Effect": "Allow",
  "Action": [
    "kinesisvideo:GetDataEndpoint",
    "kinesisvideo:GetHLSStreamingSessionURL"
  ],
  "Resource": "*",
  "Condition": {
    "StringLike": {
      "aws:ResourceTag/EdgeConnectorForKVS": "*groupingId*"
    }
  }
},
{
  "Effect": "Allow",
  "Action": [
    "iotsitewise:GetAssetPropertyValue",
    "iotsitewise:GetInterpolatedAssetPropertyValues"
  ],
  "Resource": "*",
  "Condition": {
    "StringLike": {
      "aws:ResourceTag/EdgeConnectorForKVS": "*groupingId*"
    }
  }
},
...
```

Estas declaraciones restringen la reproducción de vídeo en streaming y el acceso al historial de AWS IoT SiteWise propiedades a recursos específicos de una agrupación. *groupingId* está definido por su caso de uso. En nuestro escenario, sería el RoomId.

Limite el permiso AWS IoT SiteWise BatchPutAssetPropertyValue

Al proporcionar este permiso, se activa la función de solicitud de carga de vídeos en el reproductor de vídeo. Cuando suba un vídeo, puede especificar un intervalo de tiempo y enviar la solicitud seleccionando Enviar en el panel del panel de control de Grafana.

Para conceder BatchPutAssetPropertyValue permisos a iotsitewise:, usa la política predeterminada:

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

```
  "Action": [
    "iotsitewise:BatchPutAssetPropertyValue"
  ],
  "Resource": "*",
  "Condition": {
    "StringLike": {
      "aws:ResourceTag/EdgeConnectorForKVS": "*workspaceId*"
    }
  }
},
...
```

Al usar esta política, los usuarios pueden BatchPutAssetPropertyValue solicitar cualquier propiedad del AWS IoT SiteWise recurso de cámara. Puede restringir la autorización de un propertyId específico especificándolo en la condición de la declaración.

```
{
  ...
  "Condition": {
    "StringEquals": {
      "iotsitewise:propertyId": "propertyId"
    }
  }
  ...
}
```

El panel Reproductor de vídeo de Grafana ingiere los datos en la propiedad de medición, denominada VideoUploadRequest, para iniciar la carga del vídeo desde la memoria caché perimetral a Kinesis Video Streams. Busque el PropertyID de esta propiedad en la consola. AWS IoT SiteWise Para modificar *YourWorkspaceId*`DashboardPolicy`, utilice las siguientes declaraciones de política:

```
...,
{
  "Effect": "Allow",
  "Action": [
    "iotsitewise:BatchPutAssetPropertyValue"
  ],
  "Resource": "*",
  "Condition": {
    "StringLike": {
      "aws:ResourceTag/EdgeConnectorForKVS": "*workspaceId*"
```

```
    },
    "StringEquals": {
      "iotsitewise:propertyId": "VideoUploadRequestPropertyId"
    }
  }
},
...
```

Esta declaración restringe la ingesta de datos a una propiedad específica del recurso de cámara AWS IoT SiteWise etiquetado. Para obtener más información, consulte Cómo funciona AWS IoT SiteWise con IAM.

Connect AWS IoT SiteWise Alarms a los paneles de AWS IoT TwinMaker Grafana

> **Note**
>
> Esta característica está en su versión preliminar y está sujeta a cambios.

AWS IoT TwinMaker es capaz de importar AWS IoT SiteWise y enviar alarmas a AWS IoT TwinMaker los componentes. Esto le permite consultar el estado de las alarmas y configurar los umbrales de las alarmas sin necesidad de implementar un conector de datos personalizado para la migración de AWS IoT SiteWise datos. Puede usar el complemento AWS IoT TwinMaker Grafana para visualizar el estado de la alarma y configurar el umbral de alarma en Grafana, sin realizar llamadas a la API AWS IoT TwinMaker ni interactuar directamente con las alarmas. AWS IoT SiteWise

> **Note**
>
> Aviso de fin de soporte: el 20 de mayo de 2026, AWS finalizará el soporte para. AWS IoT Events Después del 20 de mayo de 2026, ya no podrás acceder a la AWS IoT Events consola ni a AWS IoT Events los recursos. Para obtener más información, consulta AWS IoT Events el fin del soporte.

AWS IoT SiteWise requisitos previos de configuración de alarmas

Antes de crear alarmas e integrarlas en su panel de Grafana, asegúrese de haber revisado los siguientes requisitos previos:

- Familiarícese con AWS IoT SiteWise el modelo y el sistema de activos. Para obtener más información, consulte Creación de modelos de activos y Creación de activos en la Guía del AWS IoT SiteWise usuario.
- Familiarícese con los modelos de alarma de IoT Events y cómo adjuntarlos a un AWS IoT SiteWise modelo. Para obtener más información, consulte Definición de alarmas de AWS IoT eventos en la Guía del AWS IoT SiteWise usuario.

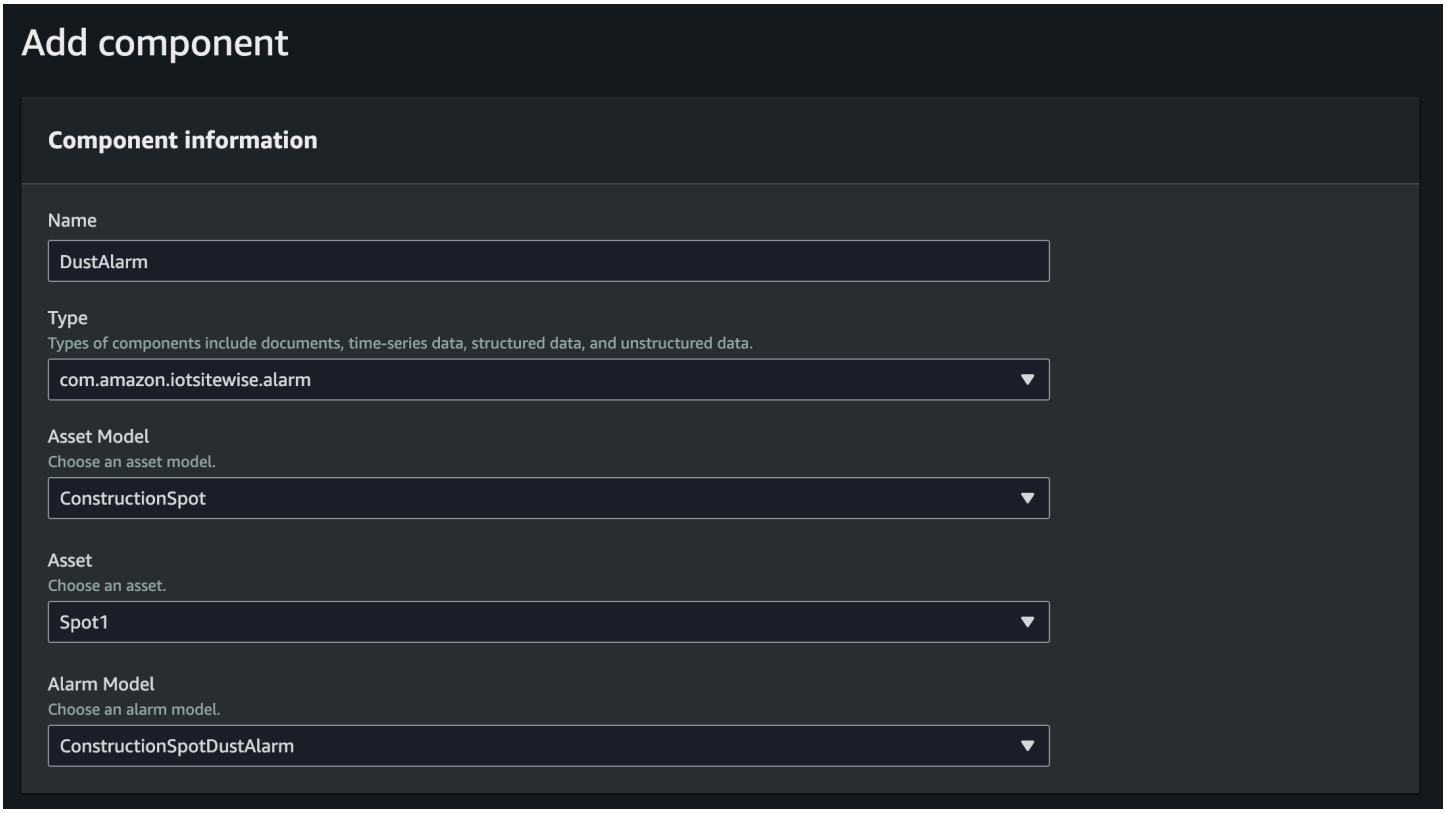
- Intégrelo AWS IoT TwinMaker con Grafana para que pueda acceder a sus AWS IoT TwinMaker recursos en Grafana. Para obtener más información, consulte, Integración del panel de control de AWS IoT TwinMaker Grafana.

Defina la función de AWS IoT SiteWise IAM del componente de alarma

AWS IoT TwinMaker utiliza la función de IAM del espacio de trabajo para consultar y configurar el umbral de alarma en Grafana. Se requieren los siguientes permisos en el rol de AWS IoT TwinMaker espacio de trabajo para interactuar con AWS IoT SiteWise las alarmas en Grafana:

```
{
"Effect": "Allow",
    "Action": [
        "iotevents:DescribeAlarmModel",
    ],
    "Resource": ["{IoTEventsAlarmModelArn}"]
},{
"Effect": "Allow",
    "Action": [
        "iotsitewise:BatchPutAssetPropertyValue"
    ],
    "Resource": ["{IoTSitewiseAssetArn}"]
}
```

En la AWS IoT TwinMaker consola, crea una entidad que represente tu AWS IoT SiteWise activo. Asegúrese de añadir un componente para esa entidad utilizando `com.amazon.iotsitewise.alarm` como tipo de componente y seleccione los modelos de activos y alarmas correspondientes.



La captura de pantalla anterior es un ejemplo de cómo crear esta entidad con este tipo. `com.amazon.iotsitewise.alarm`

Al crear este componente, importa AWS IoT TwinMaker automáticamente las propiedades de alarma relacionadas desde AWS IoT SiteWise y AWS IoT Events. Puede repetir este patrón de tipo de componente de alarma para crear componentes de alarma para todos los activos necesarios en su espacio de trabajo.

Realice consultas y actualícelas a través de la AWS IoT TwinMaker API

Tras crear los componentes de la alarma, puede consultar el estado y el umbral de la alarma y actualizar los umbrales de alarma a través de la AWS IoT TwinMaker API.

A continuación se incluye una solicitud de muestra para consultar el estado de la alarma:

```
aws iottwinmaker get-property-value-history --cli-input-json \
'{
    "workspaceId": "{workspaceId}",
```

```
    "entityId": "{entityId}",
    "componentName": "{componentName}",
    "selectedProperties": ["alarm_status"],
    "startTime": "{startTimeIsoString}",
    "endTime": "{endTimeIsoString}"
}'
```

A continuación se incluye una solicitud de muestra para consultar el umbral de la alarma.

```
aws iottwinmaker get-property-value-history --cli-input-json \
'{
    "workspaceId": "{workspaceId}",
    "entityId": "{entityId}",
    "componentName": "{componentName}",
    "selectedProperties": ["alarm_threshold"],
    "startTime": "{startTimeIsoString}",
    "endTime": "{endTimeIsoString}"
}'
```

A continuación se incluye una solicitud de muestra para actualizar el umbral de la alarma:

```
aws iottwinmaker batch-put-property-values --cli-input-json \
'{
    "workspaceId": "{workspaceId}",
    "entries": [
        {
            "entityPropertyReference": {
                "entityId": "{entityId}",
                "componentName": "{componentName}",
                "propertyName": "alarm_threshold"
            },
            "propertyValues": [
                {
                    "value": {
                        "doubleValue": "{newThreshold}"
                    },
                    "time": "{effectiveTimeIsoString}"
                }
            ]
        }
    ]
}'
```

Configura tu panel de control de Grafana para las alarmas

Es necesario crear un segundo rol de IAM en el panel con capacidad de escritura, que es un rol normal, pero con permiso para que la acción se añada `iottwinmaker:BatchPutPropertyValues` al TwinMaker espacio de trabajo arn, como en el ejemplo siguiente.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iottwinmaker:Get*",
               "iottwinmaker:List*",
                "iottwinmaker:BatchPutPropertyValues"
            ],
            "Resource": [
                "{workspaceArn}",
                "{workspaceArn}/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": "iottwinmaker:ListWorkspaces",
            "Resource": "*"
        }
    ]
}
```

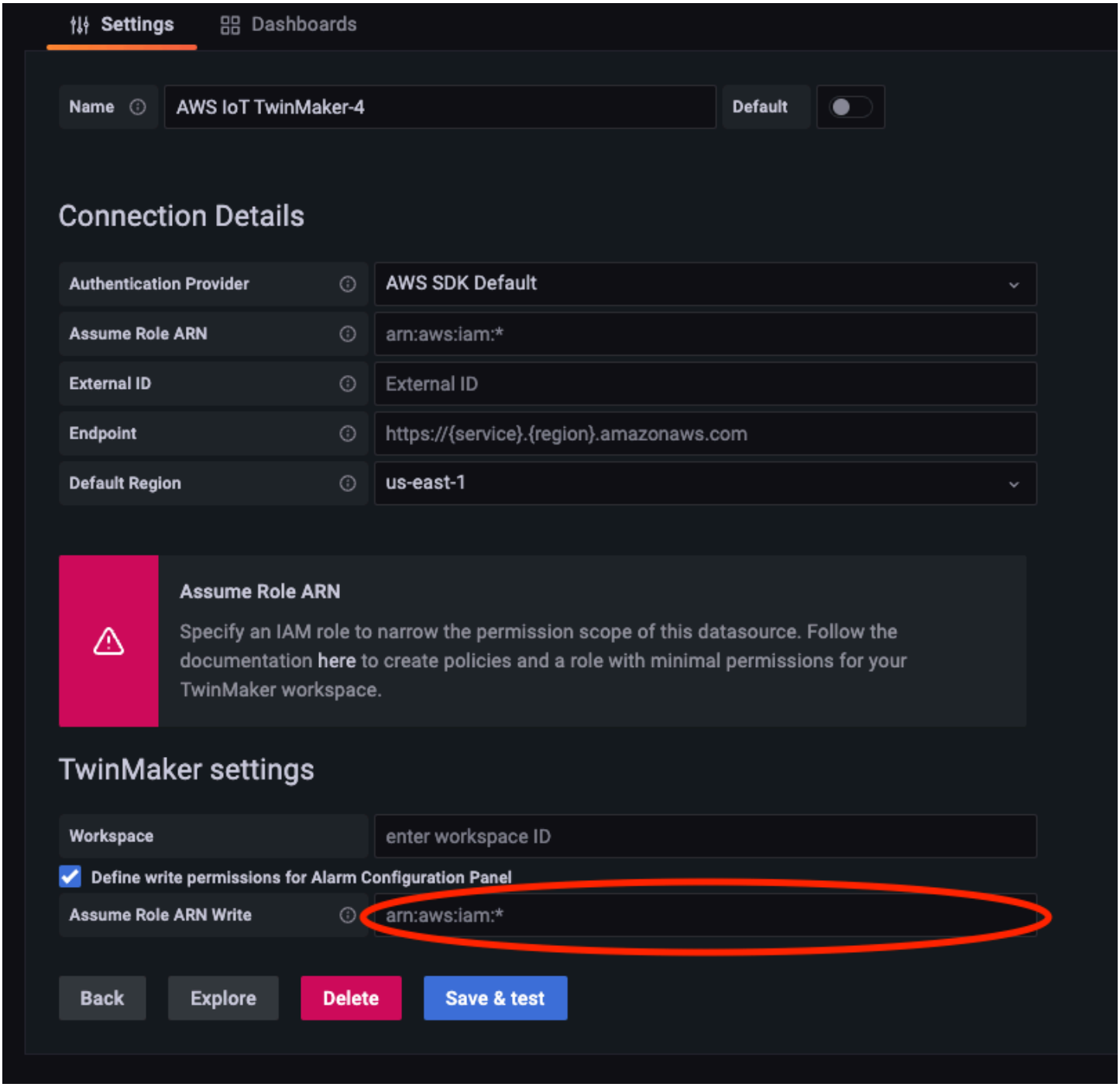
Si lo prefiere, puede agregar esta instrucción al final del rol de IAM:

```
{
    "Effect": "Allow",
    "Action": [
        "iottwinmaker:BatchPutPropertyValues"
    ],
    "Resource": [
```

```
        "{workspaceArn}",
        "{workspaceArn}/*"
    ]
}
```

El origen de datos debe tener su ARN de escritura establecido con el rol de escritura del panel que creó.

Después de modificar tu rol de IAM, inicia sesión en tu panel de control de Grafana para asumir el rol actualizado arn. Selecciona la casilla Definir permisos de escritura para el panel de configuración de alarmas y copia el arn correspondiente a la función de escritura.



Utilice el panel de control de Grafana para la visualización de las alarmas

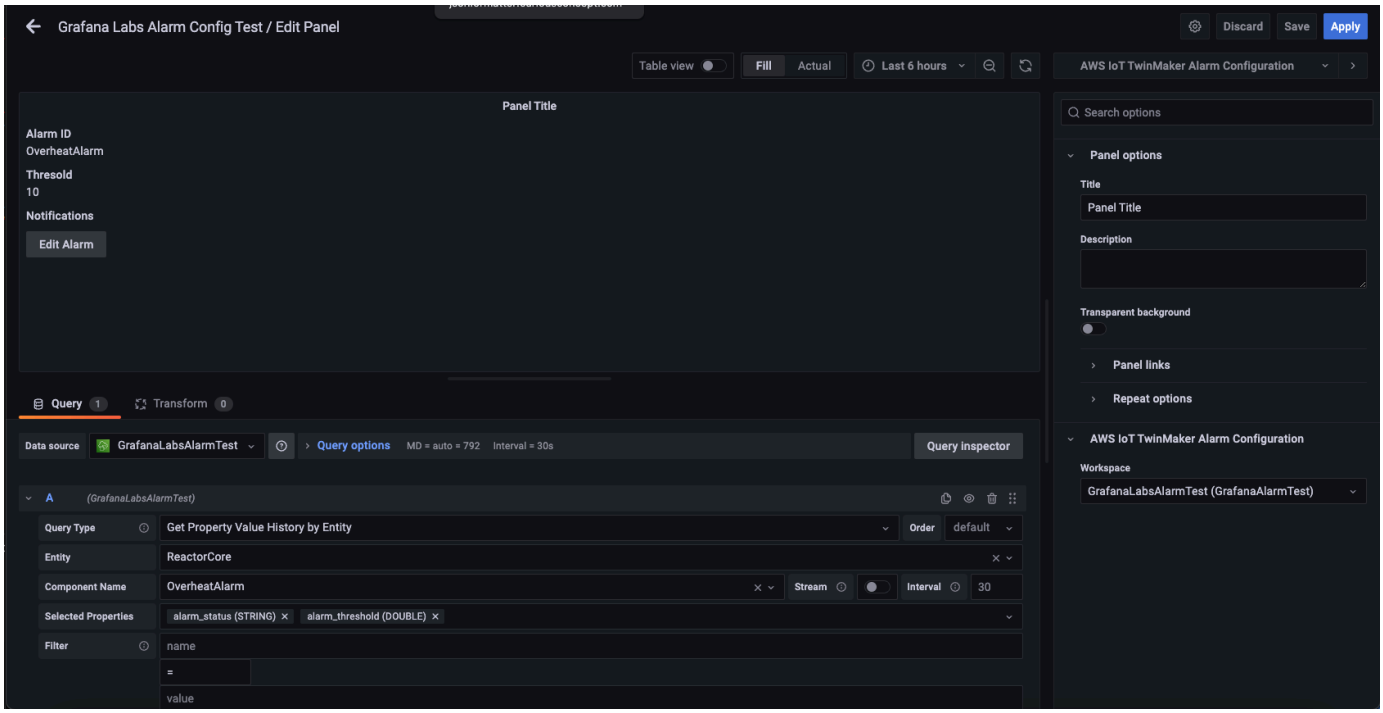
Utilice el siguiente procedimiento para agregar un panel de configuración de las alarmas a su panel y configurarlo:

1. Seleccione el espacio de trabajo en las opciones del panel.
2. Defina su origen de datos en la configuración de la consulta.
3. Utilice el siguiente tipo de consulta: `Get Property Value History by Entity`.
4. Seleccione una entidad o variable de una entidad a la que desee agregar una alarma.
5. Una vez que haya seleccionado la entidad, seleccione un componente o una variable de componente a los que desea aplicar una propiedad.
6. Para la propiedad, elija: `alarm_status` y `alarm_threshold`.

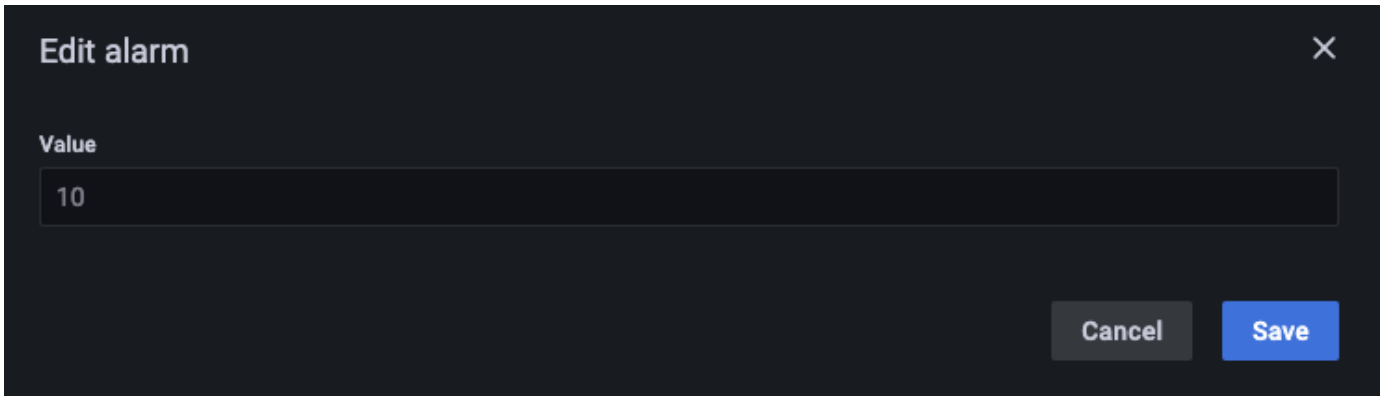
   Una vez conectado, debería ver el identificador de la alarma y su umbral actual.

   > **Note**
   >
   > En la vista previa pública, no se muestra ninguna notificación. Debe revisar el estado y el umbral de la alarma para asegurarse de que las propiedades se hayan aplicado correctamente.

7. Se debe usar el Orden ascendente de consulta predeterminado para que se muestre el último valor.
8. La sección de filtro de la Consulta se puede dejar vacía. A continuación, se muestra la configuración completa:



9. Cuando utiliza el botón Editar alarma, puede abrir un cuadro de diálogo para cambiar el umbral de la alarma actual.
10. Seleccione Guardar para establecer el nuevo valor del umbral.



> **Note**
>
> Este panel solo debe usarse con un intervalo de tiempo en vivo que incluya el presente. Si se utiliza con intervalos de tiempo que finalizan y comienzan en el pasado, es posible que se muestren valores inesperados cuando edite los umbrales de la alarma, como siempre ocurre con el umbral actual.

AWS IoT TwinMaker Integración de Matterport

Matterport ofrece una variedad de opciones de captura para escanear entornos del mundo real y crear modelos 3D inmersivos, también conocidos como gemelos digitales de Matterport. Estos modelos se denominan espacios de Matterport. AWS IoT TwinMaker admite la integración de Matterport, lo que permite importar los gemelos digitales de Matterport a las escenas de AWS IoT TwinMaker . Al combinar los gemelos digitales de Matterport con AWS IoT TwinMaker, puede visualizar y monitorear su sistema de gemelos digitales en un entorno virtual.

Para obtener más información sobre el uso de Matterport, lea la documentación de Matterport en la página de AWS IoT TwinMaker y Matterport.

Temas de integración

- Información general de la integración
- Requisitos previos para la integración de Matterport
- Genere y registre sus credenciales de Matterport
- Guarde sus credenciales de Matterport en AWS Secrets Manager
- Importe espacios de Matterport a las escenas AWS IoT TwinMaker
- Usa los espacios de Matterport en tu panel de control de Grafana AWS IoT TwinMaker

- Utilice los espacios de Matterport en su aplicación web AWS IoT TwinMaker

Información general de la integración

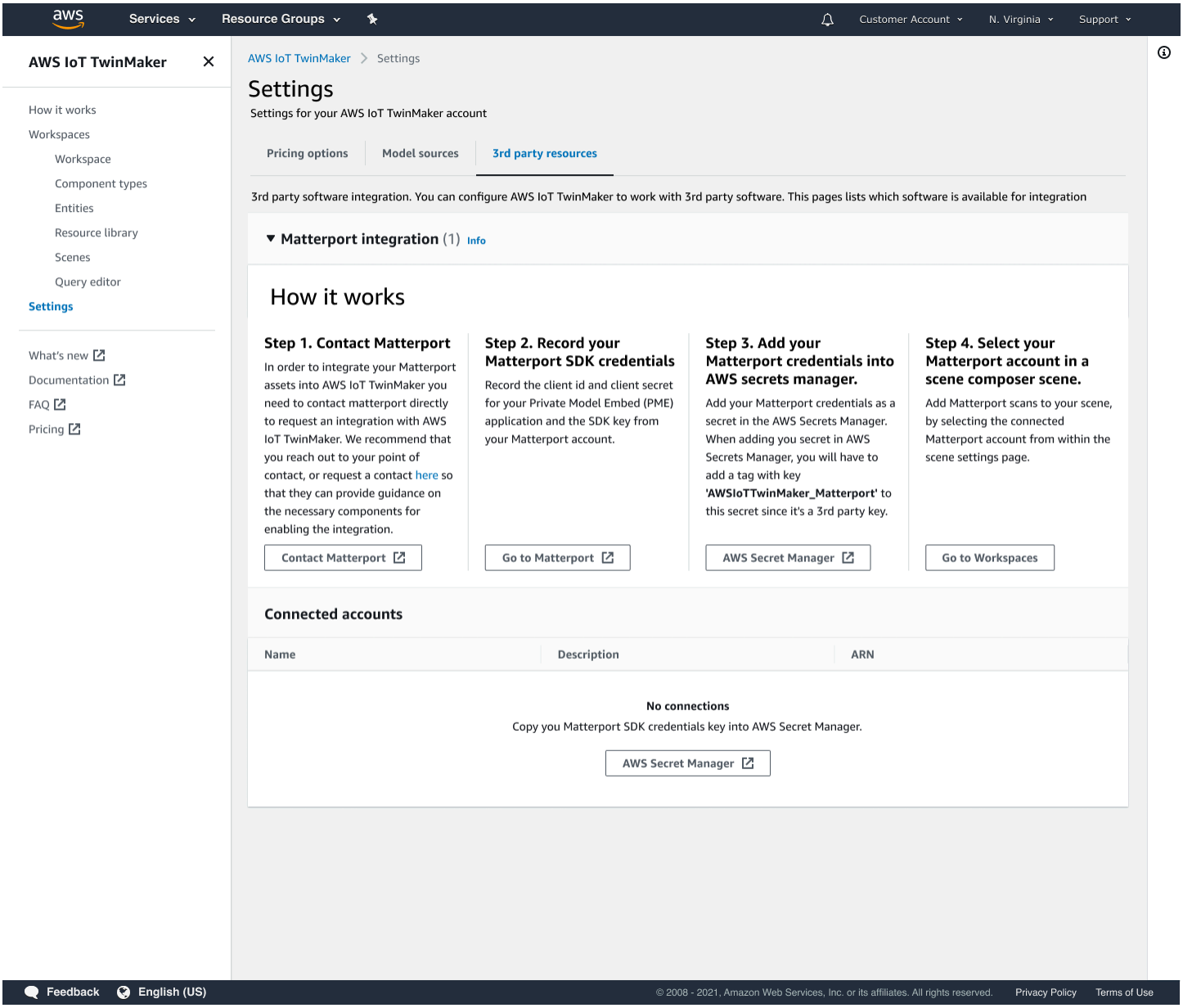
Esta integración permite hacer lo siguiente:

- Usa tus etiquetas y espacios de Matterport en el kit de aplicaciones. AWS IoT TwinMaker
- Vea los datos de Matterport importados en su panel de control de Grafana AWS IoT TwinMaker . Para obtener más información sobre el uso AWS IoT TwinMaker de Grafana, lee la documentación de integración del panel de control de Grafana.
- Importa tus espacios de Matterport a tus escenas. AWS IoT TwinMaker
- Seleccione e importe las etiquetas de Matterport que desee vincular a los datos de su escena. AWS IoT TwinMaker
- Muestre automáticamente su espacio de Matterport, etiquete los cambios en la AWS IoT TwinMaker escena y apruebe cuáles desea sincronizar.

El proceso de integración consta de 3 pasos fundamentales.

1. Genere y registre sus credenciales de Matterport
2. Guarde sus credenciales de Matterport en AWS Secrets Manager
3. Importe espacios de Matterport a las escenas AWS IoT TwinMaker

La integración se inicia en la consola de AWS IoT TwinMaker. En la página de Ajustes de la consola, en Recursos de terceros, abra la Integración de Matterport para navegar entre los distintos recursos necesarios para la integración.



Requisitos previos para la integración de Matterport

Antes de integrar Matterport con AWS IoT TwinMaker , asegúrese de cumplir los siguientes requisitos previos:

- Ha adquirido una cuenta de Matterport de nivel empresarial y los productos de Matterport necesarios para la integración. AWS IoT TwinMaker
- Dispone de AWS IoT TwinMaker un espacio de trabajo. Para obtener más información, consulta Cómo empezar con AWS IoT TwinMaker.

- Has actualizado tu rol en el AWS IoT TwinMaker espacio de trabajo. Para obtener más información sobre la creación de un rol en el espacio de trabajo, consulte Crear y administrar un rol de servicio para AWS IoT TwinMaker.

  Agregue lo siguiente al rol de su espacio de trabajo:

  ```
  {
    "Effect": "Allow",
    "Action": "secretsmanager:GetSecretValue",
    "Resource": [
     "AWS Secrets Manager secret ARN"
    ]
  }
  ```

- Debe ponerse en contacto con Matterport para configurar las licencias necesarias para habilitar la integración. Matterport también habilitará Private Model Embed (PME) para la integración.

  Si ya tiene un administrador de cuentas de Matterport, póngase en contacto con él directamente.

  Utilice el siguiente procedimiento para ponerse en contacto con Matterport y solicitar una integración si no tiene un punto de contacto en Matterport:

  1. Abra la página de Matterport y de AWS IoT TwinMaker.
  2. Pulse el botón Contáctenos para abrir el formulario de contacto.
  3. Complete la información solicitada en el formulario.
  4. Cuando esté listo, elija SAY HELLO para enviar su solicitud a Matterport.

  Una vez que haya solicitado la integración, puede generar las credenciales del SDK de Matterport y Private Model Embed (PME) necesarias para continuar con el proceso de integración.

  > **Note**
  >
  > Esto puede implicar el pago de una tarifa por la compra de nuevos productos o servicios.

Genere y registre sus credenciales de Matterport

Para integrar Matterport con AWS IoT TwinMaker, debes proporcionar AWS Secrets Manager las credenciales de Matterport. Utilice el siguiente procedimiento para generar las credenciales del SDK de Matterport.

1. Inicie sesión en su cuenta de Matterport.
2. Diríjase a la página de ajustes de su cuenta.
3. Una vez en la página de ajustes, seleccione la opción Herramientas para desarrolladores.
4. En la página de Herramientas para desarrolladores, vaya la sección Administración de claves del SDK.
5. Una vez en la sección de Administración de claves del SDK, seleccione la opción para agregar una nueva clave del SDK.
6. Una vez que tengas la clave del SDK de Matterport, añade dominios a la clave AWS IoT TwinMaker y a tu servidor Grafana. Si utilizas el kit de AWS IoT TwinMaker aplicaciones, asegúrate de añadir también tu dominio personalizado.
7. A continuación, busque la sección Administración de la integración de aplicaciones y verá su aplicación de PME en la lista. Registre la siguiente información:

   - El ID de cliente
   - El Secreto del cliente

   > **Note**
   >
   > Debido a que el Secreto del cliente solo se presenta una vez, recomendamos encarecidamente que registre su Secreto del cliente. Debe presentar su Secreto de cliente en la consola de AWS Secrets Manager para continuar con la integración de Matterport.

   Estas credenciales se crean automáticamente cuando adquiere los componentes necesarios y Matterport habilita la PME de su cuenta. Si estas credenciales no aparecen, póngase en contacto con Matterport. Para contactarse, consulte el formulario de contacto de Matterport y AWS IoT TwinMaker.

Para obtener más información sobre las credenciales del SDK de Matterport, consulte la documentación oficial del SDK de Matterport Descripción general de los documentos del SDK.

Guarde sus credenciales de Matterport en AWS Secrets Manager

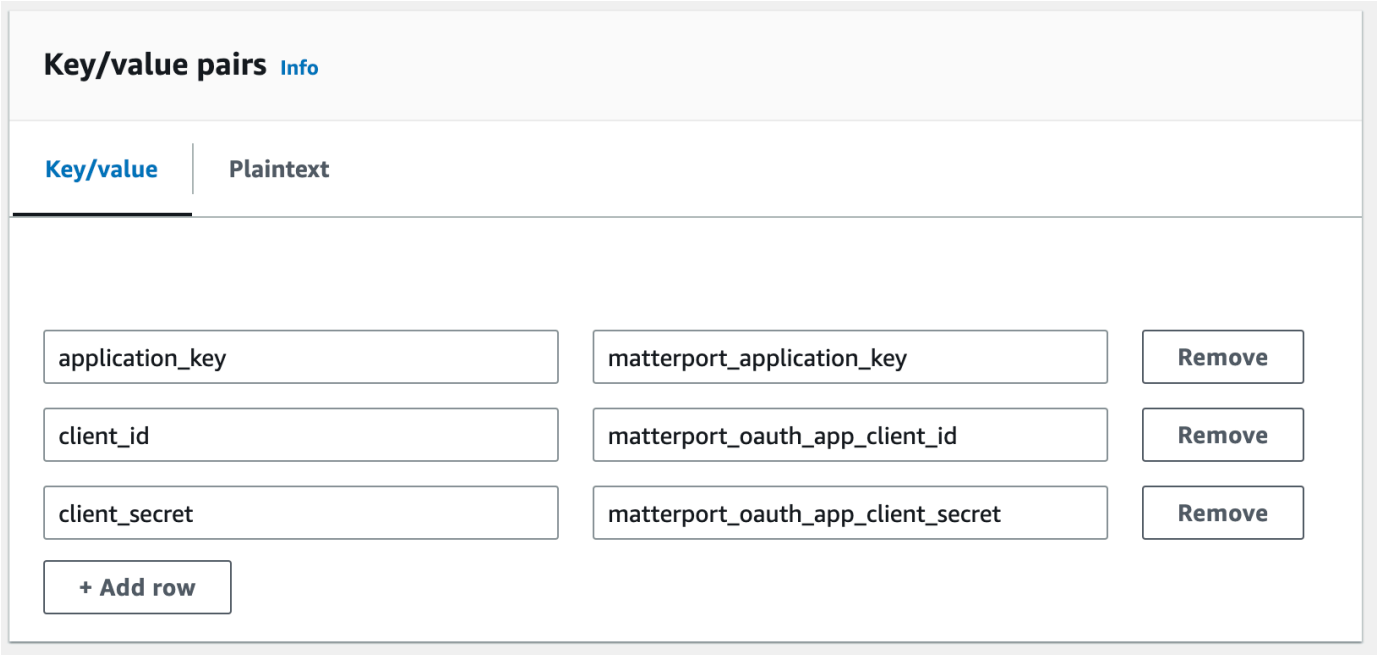
Utilice el siguiente procedimiento para almacenar sus credenciales de Matterport en. AWS Secrets Manager

> **Note**
>
> Necesita el Secreto de cliente creado a partir del procedimiento del tema Genere y registre sus credenciales de Matterport para continuar con la integración de Matterport.

1. Inicie sesión en la AWS Secrets Manager consola.
2. Vaya a la página de Secretos y seleccione Guardar un secreto nuevo.
3. En Tipo de secreto, seleccione Otro tipo de secreto.
4. En la sección de pares clave/valor, agregue los siguientes pares clave-valor, con sus credenciales de Matterport como valores:

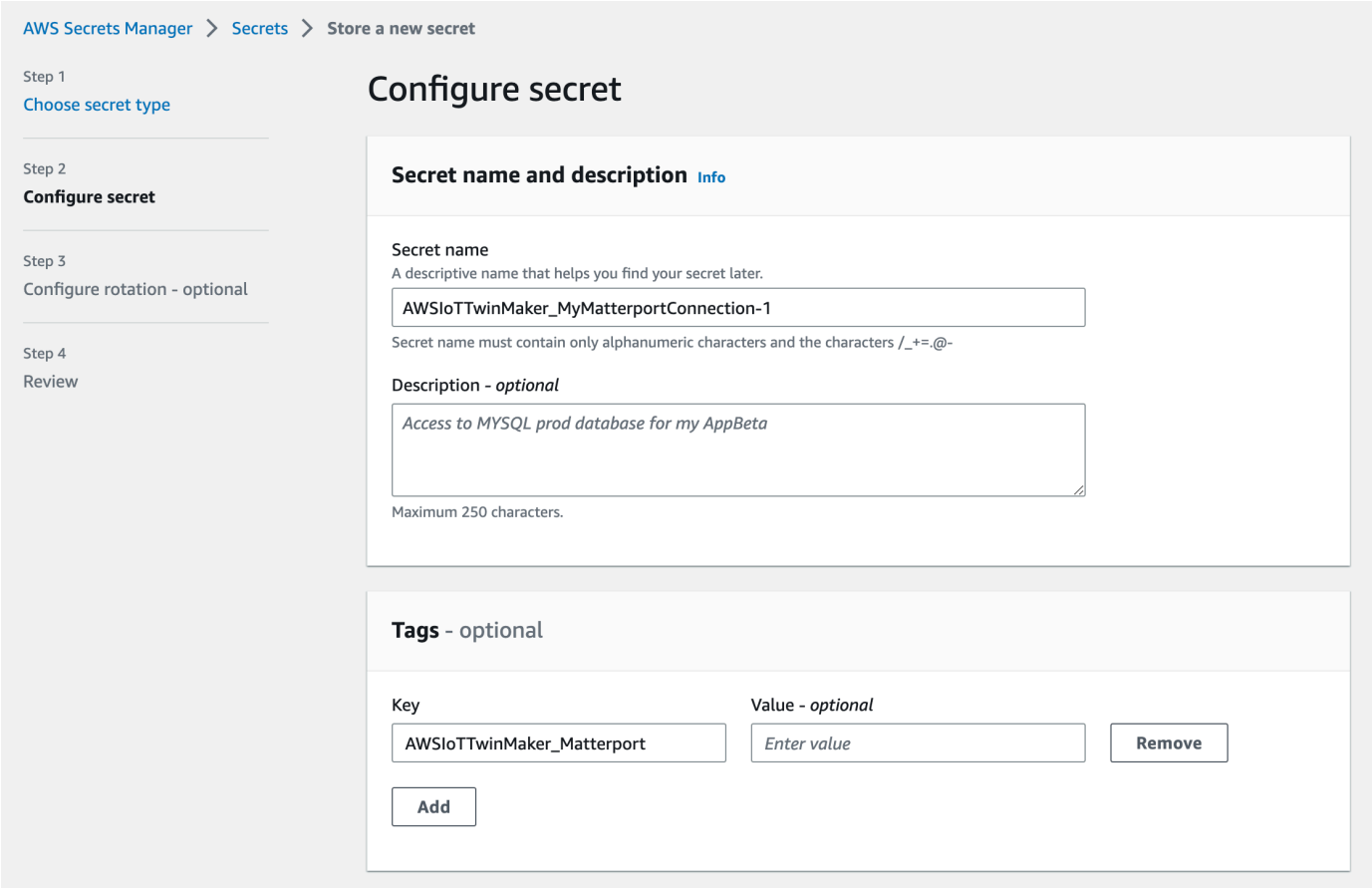
   - Cree un par clave-valor, con Clave: `application_key` y Valor: *<sus credenciales de Matterport>*.
   - Cree un par clave-valor, con Clave: `client_id` y Valor: *<sus credenciales de Matterport>*.
   - Cree un par clave-valor, con Clave: `client_secret` y Valor: *<sus credenciales de Matterport>*.

   Cuando finalice, debería disponer de una configuración similar a la del siguiente ejemplo:



5. En la Clave de cifrado, puede dejar seleccionada la clave de cifrado predeterminada `aws/secretsmanager`.
6. Seleccione Siguiente para pasar a la página de Configuración del secreto.
7. Rellene el campo correspondiente al Nombre del secreto y a la Descripción.
8. Agregue una etiqueta a este secreto en la sección Etiquetas.

   Al crear la etiqueta, asigne la clave `AWSIoTTwinMaker_Matterport` como se muestra en la siguiente captura de pantalla:



> **Note**
>
> Debe agregar una etiqueta. Las etiquetas son obligatorias cuando se agregan secretos de terceros a AWS Secrets Manager, a pesar de que las Etiquetas aparecen como opcionales.

El campo Valor es opcional. Una vez que haya proporcionado una Clave, puede seleccionar Agregar para pasar al siguiente paso.

9. Seleccione Siguiente para pasar a la página Configurar rotación automática. La configuración de una rotación secreta es opcional. Si desea terminar de agregar su secreto y no necesita una rotación, vuelva a seleccionar Siguiente. Para obtener más información sobre la rotación secreta, consulte Rotar AWS Secrets Manager secretos.
10. Confirme su configuración secreta en la página de Revisión. Cuando esté listo para agregar su secreto, seleccione Tienda.

Para obtener más información sobre su uso AWS Secrets Manager, consulte la siguiente AWS Secrets Manager documentación:

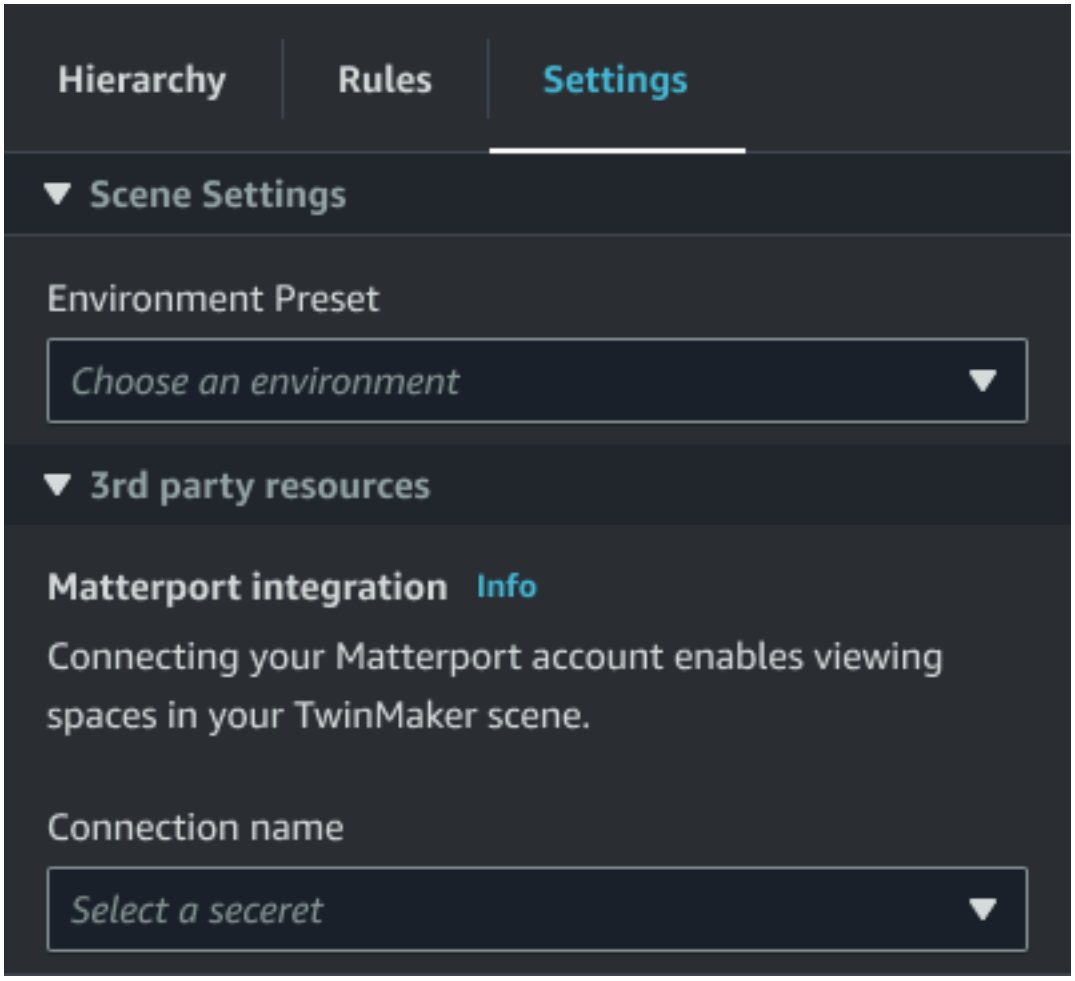
- Cree y administre secretos con AWS Secrets Manager
- ¿Qué es AWS Secrets Manager?
- Rota AWS Secrets Manager los secretos

Ahora está listo para importar sus activos de Matterport a AWS IoT TwinMaker las escenas. Consulte el procedimiento en la siguiente sección: Importe espacios de Matterport a las escenas AWS IoT TwinMaker

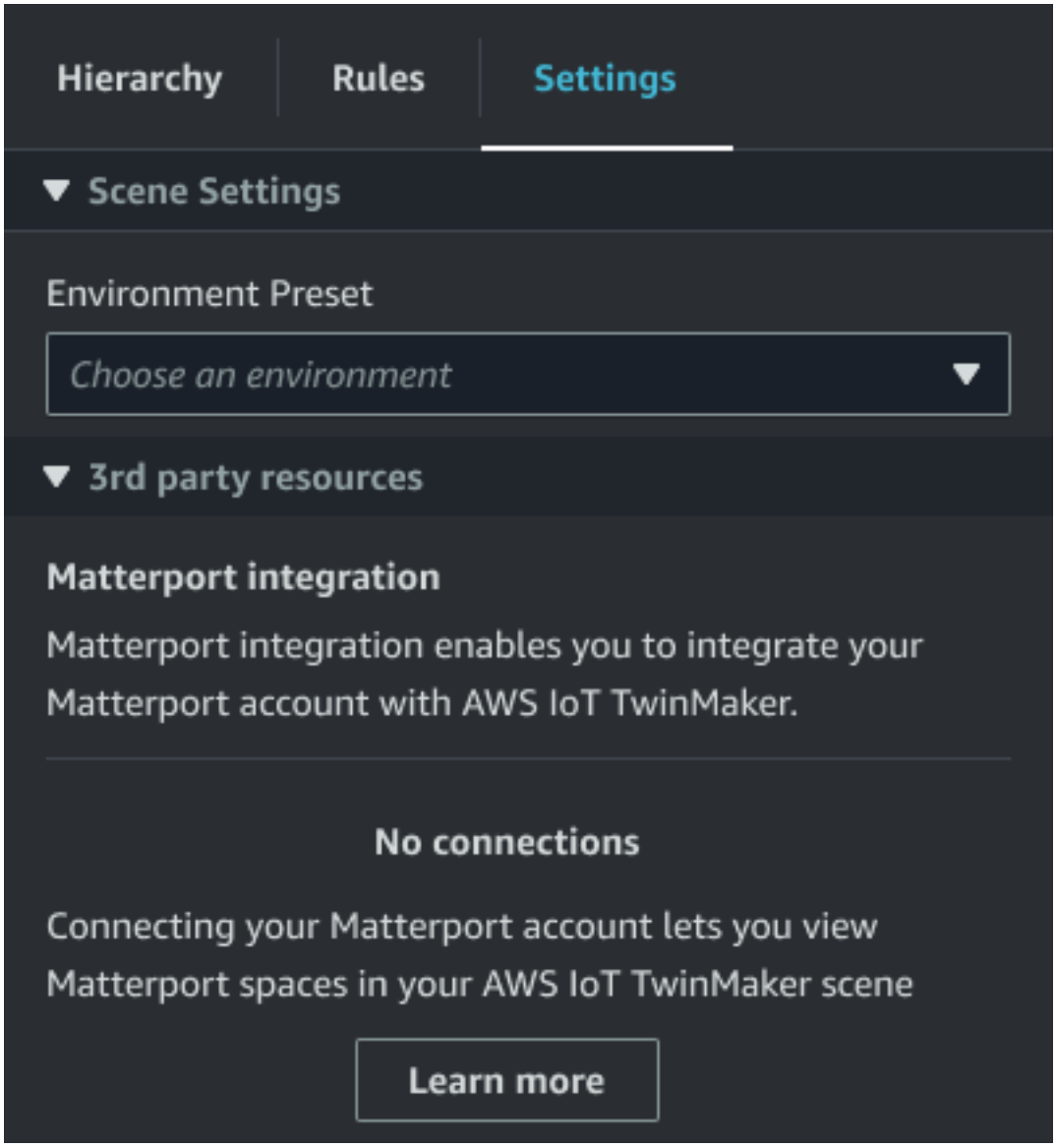
Importe espacios de Matterport a las escenas AWS IoT TwinMaker

Agregue análisis de Matterport a su escena seleccionando la cuenta de Matterport conectada desde la página de ajustes de la escena. Utilice el siguiente procedimiento para importar los análisis y las etiquetas de Matterport:

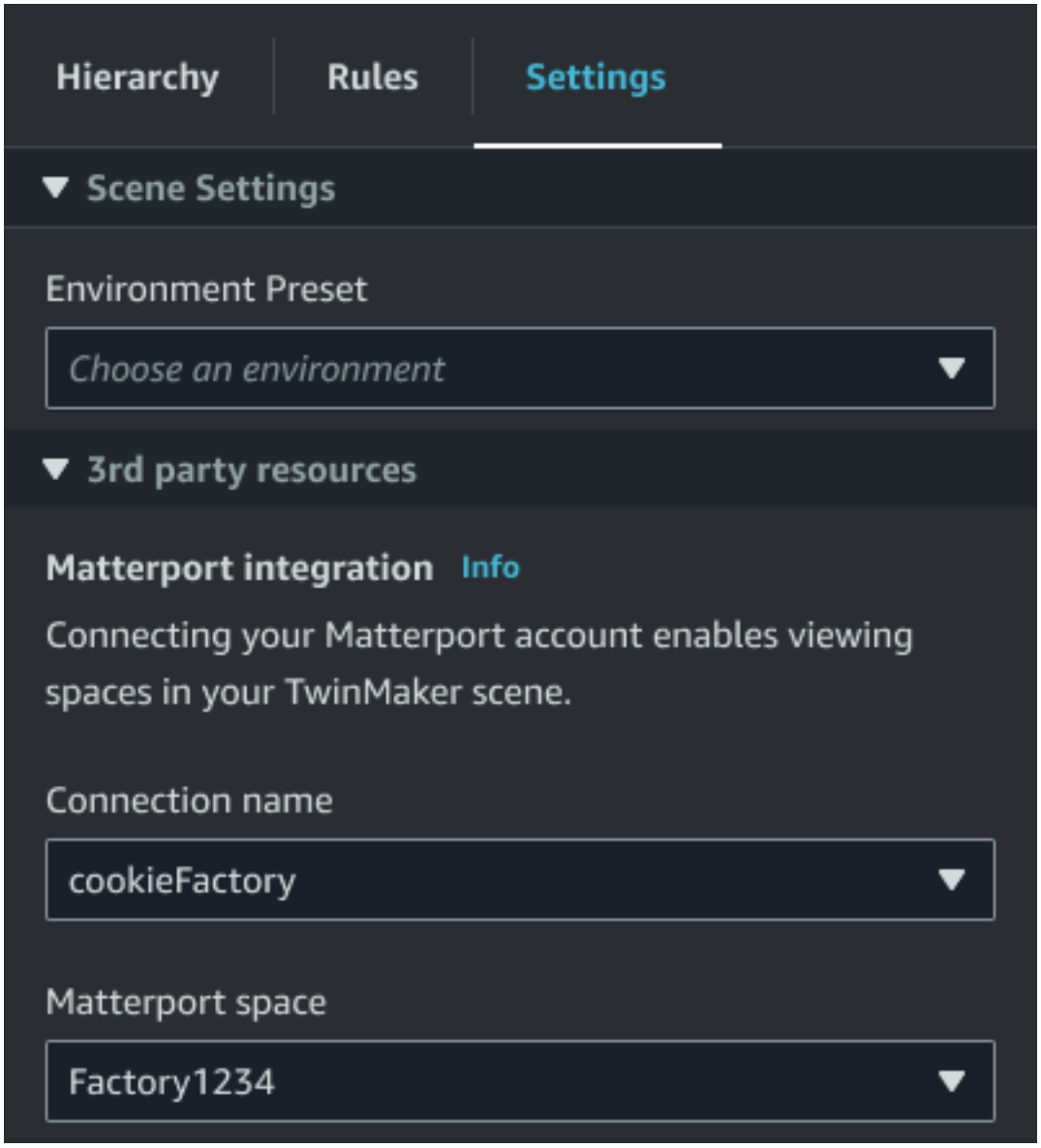
1. Inicie sesión en la consola de AWS IoT TwinMaker.
2. Cree o abra una AWS IoT TwinMaker escena existente en la que desee utilizar un espacio de Matterport.
3. Una vez que se haya abierto la escena, vaya a la pestaña Ajustes.
4. Dentro de Configuración, en Recursos de terceros, busque el Nombre de la conexión e introduzca el secreto que creó durante el procedimiento desde Guarde sus credenciales de Matterport en AWS Secrets Manager.



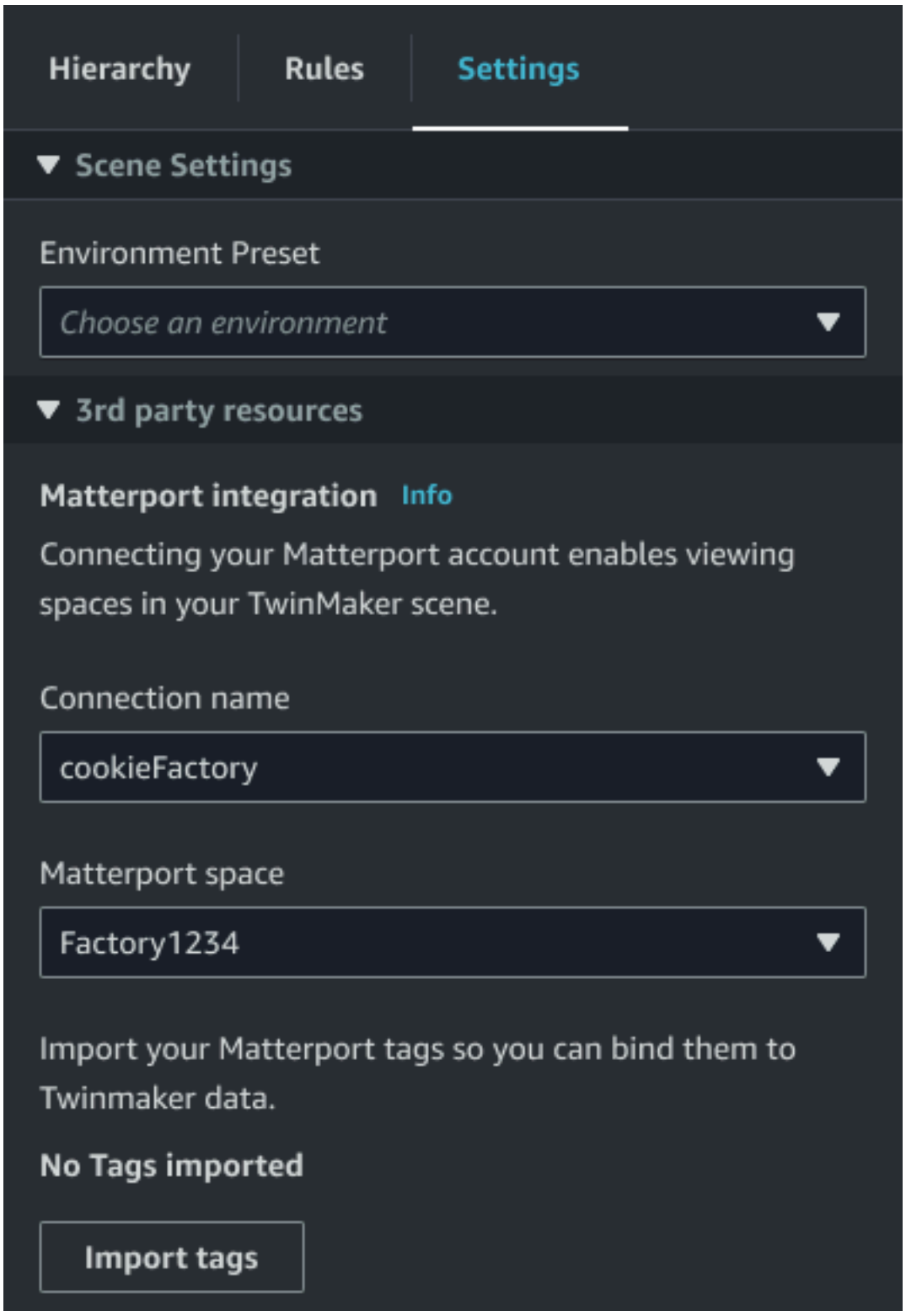
> **Note**
>
> Si ve un mensaje que indica que No hay conexiones, vaya a la página de ajustes de la consola de AWS IoT TwinMaker para iniciar el proceso de integración de Matterport.



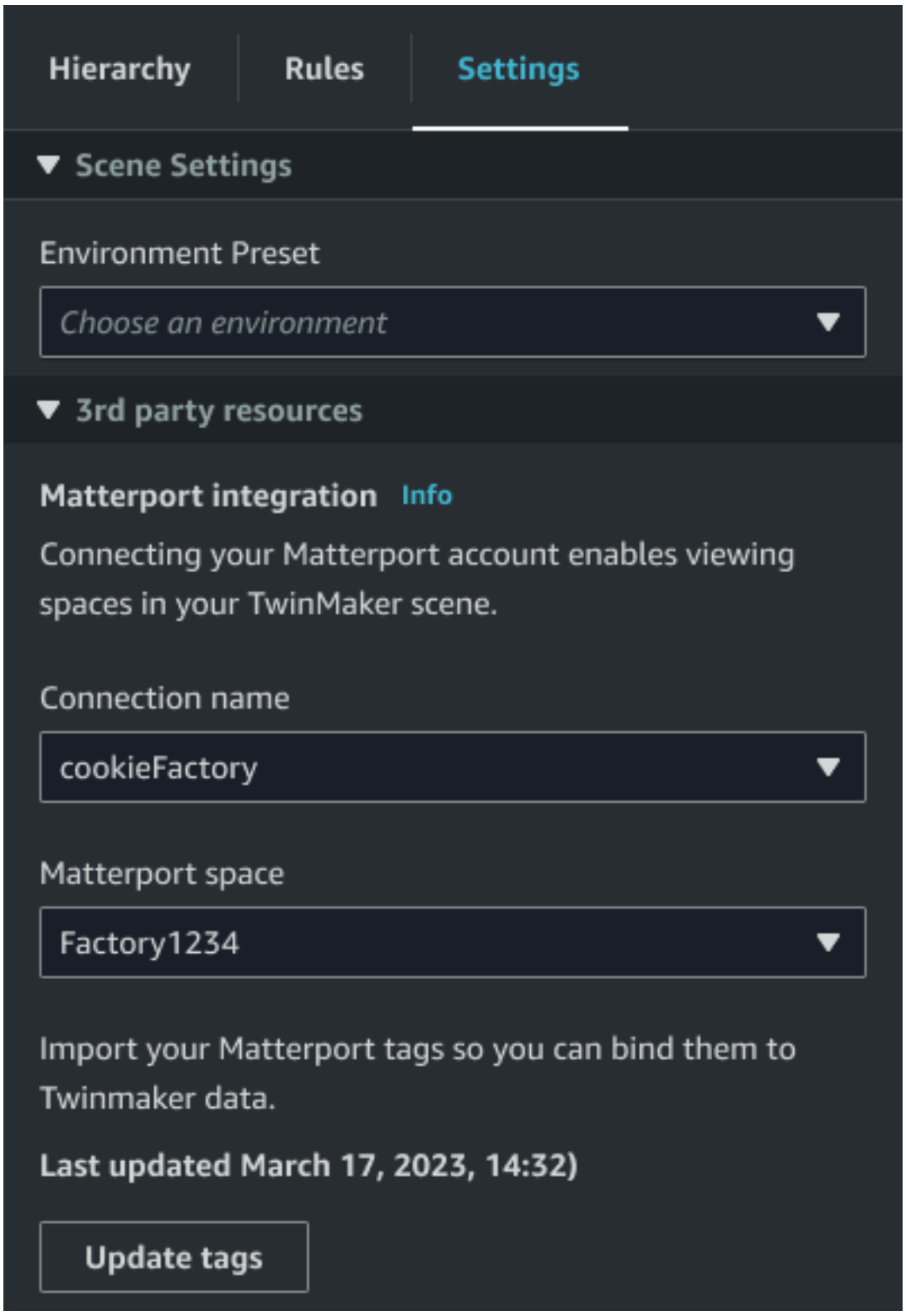
5. A continuación, elija el espacio de Matterport que desearía usar en la escena, seleccionándolo en la lista desplegable del espacio de Matterport.



6. Tras seleccionar un espacio, puede importar sus etiquetas de Matterport y convertirlas en etiquetas de AWS IoT TwinMaker escena pulsando el botón Importar etiquetas.



Una vez importadas las etiquetas de Matterport, el botón se sustituye por el botón Actualizar etiquetas. Puede actualizar continuamente sus etiquetas de Matterport AWS IoT TwinMaker para que siempre reflejen los cambios más recientes en su cuenta de Matterport.



7. Te has integrado correctamente AWS IoT TwinMaker con Matterport y ahora tu AWS IoT TwinMaker escena tiene tanto el espacio como las etiquetas de Matterport que has importado. Puede trabajar en esta escena como lo haría con cualquier otra escena. AWS IoT TwinMaker

Para obtener más información sobre cómo trabajar con AWS IoT TwinMaker escenas, consulte Creación y edición de AWS IoT TwinMaker escenas.

Usa los espacios de Matterport en tu panel de control de Grafana AWS IoT TwinMaker

Una vez que haya importado su espacio de Matterport a una AWS IoT TwinMaker escena, podrá verla con el espacio de Matterport en su panel de control de Grafana. Si ya ha configurado Grafana con AWS IoT TwinMaker, simplemente puede abrir el panel de control de Grafana para ver su escena con el espacio Matterport importado.

Si aún no se ha configurado AWS IoT TwinMaker con Grafana, complete primero el proceso de integración de Grafana. Tienes dos opciones a la hora de integrarte AWS IoT TwinMaker con Grafana. Puede usar una instancia de Grafana autoadministrada o puede usar Amazon Managed Grafana.

Consulte la siguiente documentación para obtener más información sobre las opciones de Grafana y el proceso de integración:

- AWS IoT TwinMaker Integración del panel de control de Grafana.
- Amazon Managed Grafana.
- Grafana autoadministrada.

Utilice los espacios de Matterport en su aplicación web AWS IoT TwinMaker

Una vez que haya importado su espacio de Matterport a una AWS IoT TwinMaker escena, podrá verla con el espacio de Matterport en la aplicación web del kit de aplicaciones. AWS IoT

Consulte la siguiente documentación para obtener más información sobre el uso del kit de aplicación: AWS IoT

- Para obtener más información sobre cómo usarlo AWS IoT TwinMaker con el kit de AWS IoT aplicaciones, consulteCree una aplicación web personalizada mediante componentes de AWS IoT TwinMaker interfaz de usuario.
- Para obtener más información sobre el uso del kit de AWS IoT aplicaciones, visita la página de Github del kit de AWS IoT aplicaciones.

- Para obtener instrucciones sobre cómo iniciar una nueva aplicación web con el kit de AWS IoT aplicaciones, visite la página de documentación oficial del IoT App Kit.

AWS IoT TwinMaker integración de vídeo

Las cámaras de vídeo ofrecen una buena oportunidad para la simulación de los gemelos digitales. Puede utilizar AWS IoT TwinMaker para simular la ubicación y las condiciones físicas de la cámara. Cree entidades AWS IoT TwinMaker para sus cámaras in situ y utilice componentes de vídeo para transmitir vídeo y metadatos en directo desde su sitio a su AWS IoT TwinMaker escena o a un panel de control de Grafana.

AWS IoT TwinMaker puede capturar vídeo desde dispositivos periféricos de dos maneras. Puede transmitir vídeo desde dispositivos periféricos con el conector periférico para la secuencia de vídeo de Kinesis, o puede guardar el vídeo en el dispositivo periférico e iniciar la carga del vídeo con mensajes MQTT. Utilice este componente para transmitir datos de vídeo desde sus dispositivos para utilizarlos con AWS IoT los servicios. Para generar los recursos necesarios e implementar el conector perimetral para Kinesis Video Streams, consulte Introducción al conector perimetral para la transmisión de vídeo de Kinesis en adelante. GitHub Para obtener más información sobre el AWS IoT Greengrass componente, consulte la AWS IoT Greengrass documentación sobre Edge Connector para Kinesis Video Streams.

Una vez que haya creado los AWS IoT SiteWise modelos necesarios y configurado el componente Greengrass de Kinesis Video Streams, podrá transmitir o grabar vídeo in situ a su aplicación gemela digital de la consola. AWS IoT TwinMaker También puede ver las transmisiones en directo y los metadatos de sus dispositivos en un panel de Grafana. Para obtener más información sobre la integración de Grafana y AWS IoT TwinMaker, consulte. Integración del panel de control de AWS IoT TwinMaker Grafana

Utilice el conector perimetral para la transmisión de vídeo de Kinesis para transmitir vídeo en AWS IoT TwinMaker

Con el conector perimetral para la transmisión de vídeo de Kinesis, puede transmitir vídeo y datos a una entidad de la escena. AWS IoT TwinMaker Para ello, utilice un componente de video. Para crear el componente de vídeo para usarlo en las escenas, complete el procedimiento a continuación.

Requisitos previos

Antes de crear el componente de vídeo en la AWS IoT TwinMaker escena, asegúrese de haber cumplido los siguientes requisitos previos.

- Creó los AWS IoT SiteWise modelos y activos necesarios para el conector perimetral para la transmisión de vídeo de Kinesis. Para obtener más información sobre la creación de los activos de AWS IoT SiteWise para el conector, consulte Introducción al conector periférico para la secuencia de vídeo de Kinesis.
- Implementó el conector Kinesis video Stream Edge en su AWS IoT Greengrass dispositivo. Para obtener más información sobre la implementación del componente del conector periférico de la secuencia de vídeo de Kinesis, consulte README para su implementación.

Cree componentes de vídeo para las escenas AWS IoT TwinMaker

Complete los siguientes pasos para crear el conector periférico para el componente de la secuencia de vídeo de Kinesis para su escena.

1. En la AWS IoT TwinMaker consola, abra la escena a la que desee añadir el componente de vídeo.
2. Cuando se abra la escena, elija una entidad existente o cree la entidad a la que desee agregar el componente y, a continuación, elija Agregar componente.
3. En el panel Agregar componente, introduzca un nombre para el componente y, para el Tipo, elija com.amazon.iotsitewise.connector.edgevideo.
4. Elija un modelo de activos seleccionando el nombre del modelo de AWS IoT SiteWise cámara que ha creado. Este nombre debe tener el siguiente formato: `EdgeConnectorForKVSCameraModel-0abc`, donde la cadena de letras y números al final coincide con el nombre de su propio activo.
5. En Asset, elige los recursos de AWS IoT SiteWise cámara desde los que deseas transmitir el vídeo. Aparece una pequeña ventana que muestra una vista previa de la transmisión de vídeo actual.

   > **Note**
   >
   > Para probar la transmisión de video, elija prueba. Esta prueba envía un evento MQTT para iniciar la transmisión de vídeo en directo. Espere un momento para que el vídeo aparezca en el reproductor.

6. Para agregar un componente visible a su entidad, elija Agregar un componente.

Agregue videos y metadatos de la secuencia de vídeo de Kinesis a un panel de Grafana

Una vez que hayas creado un componente de vídeo para tu entidad en la AWS IoT TwinMaker escena, puedes configurar el panel de vídeo de Grafana para ver las transmisiones en directo. Asegúrese de que se ha integrado correctamente AWS IoT TwinMaker con Grafana. Para obtener más información, consulte Integración del panel de control de AWS IoT TwinMaker Grafana.

Important

Para ver el vídeo en su panel de control de Grafana, debe asegurarse de que las fuentes de datos de Grafana tengan los permisos de IAM adecuados. Para crear la política y el rol necesarios, consulte Creación de un rol de IAM en el panel.

Complete los siguientes pasos para ver Kinesis Video Streams y los metadatos de Kinesis en el panel de Grafana.

1. AWS IoT TwinMaker Abre el panel de control.
2. Seleccione Añadir panel y, a continuación, seleccione Añadir un panel vacío.

   Note

   Para Grafana v10.4, el reproductor de AWS IoT TwinMaker vídeo se encuentra en Widget. Selecciona Añadir >> Widget.

3. En la lista de paneles, elija el panel del reproductor de vídeo de AWS IoT TwinMaker .
4. En el panel del reproductor de AWS IoT TwinMaker vídeo, introduzca el nombre de la transmisión KinesisVideoStreamName, junto con el nombre de la transmisión de vídeo de Kinesis desde la que desee transmitir el vídeo.

   Note

   Para transmitir metadatos al panel de vídeo de Grafana, primero debe haber creado una entidad con un componente de transmisión de video.

5. Opcional: para transmitir los metadatos de AWS IoT SiteWise los activos al reproductor de vídeo, en Entity, elija la AWS IoT TwinMaker entidad que creó en la AWS IoT TwinMaker escena. Para el nombre del componente, elija el componente de vídeo que creó para la entidad en su escena de AWS IoT TwinMaker .

Uso de la biblioteca Flink de AWS IoT TwinMaker

AWS IoT TwinMaker proporciona una biblioteca Flink que puede utilizar para leer y escribir datos en almacenes de datos externos que se utilizan en sus gemelos digitales.

Para utilizar la biblioteca Flink de AWS IoT TwinMaker, debe instalarla como un conector personalizado en Managed Service para Apache Flink y llevar a cabo consultas SQL de Flink en un cuaderno de Zeppelin en Managed Service para Apache Flink. El cuaderno se puede promover a una aplicación de procesamiento de flujos en constante ejecución. La biblioteca aprovecha los componentes de AWS IoT TwinMaker para recuperar datos de su espacio de trabajo.

La biblioteca Flink de AWS IoT TwinMaker requiere lo siguiente.

Requisitos previos

1. Un espacio de trabajo completamente rellenado con escenas y componentes. Utilice los tipos de componentes integrados para los datos de los servicios de AWS (AWS IoT SiteWise y de Kinesis Video Streams). Cree tipos de componentes personalizados para los datos de orígenes externos. Para obtener más información, consulte ???.
2. Comprensión de los cuadernos de Studio con Managed Service para Apache Flink. Estos cuadernos se basan en Apache Zeppelin y utilizan el marco ApacheFlink. Para obtener más información, consulte Uso del cuaderno de Studio con Managed Service para Apache Flink.

Para obtener instrucciones sobre el uso de la biblioteca, consulte la Guía del usuario de la biblioteca Flink de AWS IoT TwinMaker.

Para obtener instrucciones sobre la configuración de AWS IoT TwinMaker con el inicio rápido en las muestras de AWS IoT TwinMaker, consulte el archivo README de la aplicación de las muestras de información.

Registro y monitoreo en AWS IoT TwinMaker

El monitoreo es importante para mantener la fiabilidad, la disponibilidad y el rendimiento de AWS IoT TwinMaker y de las otras soluciones de AWS. AWS IoT TwinMaker es compatible con las siguientes herramientas de monitoreo para vigilar el servicio, informar cuando algo no va bien y tomar medidas automáticamente cuando proceda.

- Amazon CloudWatch monitorea en tiempo real los recursos de AWS y las aplicaciones que ejecuta en AWS. Puede recopilar métricas y realizar un seguimiento de las métricas, crear paneles personalizados y definir alarmas que le advierten o que toman medidas cuando una métrica determinada alcanza el umbral que se especifique. Por ejemplo, puede hacer que CloudWatch haga un seguimiento del uso de la CPU u otras métricas de las instancias de Amazon EC2 y lanzar nuevas instancias automáticamente cuando sea necesario. Para obtener más información, consulte la Guía del usuario de Amazon CloudWatch.
- Los registros de Amazon CloudWatch supervisan, almacenan y obtienen acceso a los archivos de registro de puertas de enlace de AWS IoT TwinMaker, CloudTrail u otras fuentes. CloudWatch Logs puede monitorear información en los registros y enviarle una notificación cuando se llega a determinados umbrales. También se pueden archivar los datos de los registros en un almacenamiento de larga duración. Para obtener más información, consulte la Guía del usuario de Amazon CloudWatch Logs.
- AWS CloudTrail captura llamadas a la API y eventos relacionados efectuados por su cuenta de AWS o en su nombre, y entrega los archivos de registro al bucket de Amazon S3 que se haya especificado. También pueden identificar qué usuarios y cuentas llamaron a AWS, la dirección IP de origen de las llamadas y el momento en que se hicieron. Para obtener más información, consulte la Guía del usuario de AWS CloudTrail.

Temas

- Monitoreo de AWS IoT TwinMaker con métricas de Amazon CloudWatch
- Registrar llamadas a la API de AWS IoT TwinMaker con AWS CloudTrail

Monitoreo de AWS IoT TwinMaker con métricas de Amazon CloudWatch

Puede supervisar AWS IoT TwinMaker mediante Amazon CloudWatch, que recopila y procesa los datos sin procesar y los convierte en métricas legibles y casi en tiempo real. Estas estadísticas se mantienen durante 15 meses, de forma que pueda obtener acceso a información histórica y disponer de una mejor perspectiva sobre el desempeño de su aplicación web o servicio. También puede establecer alarmas que vigilen determinados umbrales y enviar notificaciones o realizar acciones cuando se cumplan dichos umbrales. Para obtener más información, consulte la Guía del usuario de Amazon CloudWatch.

AWS IoT TwinMaker publica las métricas y dimensiones enumeradas en las secciones siguientes en el espacio de nombres `AWS/IoTTwinMaker`.

> **Tip**
>
> AWS IoT TwinMaker publica métricas en un intervalo de un minuto. Al ver estas métricas en gráficos en la consola de CloudWatch, le recomendamos que elija un período de 1 minuto para ver la resolución más alta disponible de sus datos de métricas.

Contenido

- Métricas

Métricas

AWS IoT TwinMaker publica las siguientes métricas.

Métricas

Métrica	Descripción
`ComponentTypeCreationFailure`	Esta métrica indica si la creación del tipo de componente se ha realizado correctamente. La métrica se publica cuando un tipo de componente está en el estado `CREATING`. Esto ocurre cuando se crea un tipo de componente

Métrica	Descripción
	con las propiedades necesarias en el inicializ ador del esquema y estas propiedades se instancian con los valores predeterminados. El valor de la métrica puede ser `0` si se ha realizado correctamente o 1 en caso de fallo. Dimensiones: ComponentTypeId, Workspace Id. Unidades: recuento
`ComponentTypeUpdateFailure`	Esta métrica indica si la actualización del tipo de componente se ha realizado correctamente. La métrica se publica cuando un tipo de componente está en el estado `UPDATING`. Esto ocurre cuando se actualiza un tipo de componente con las propiedades necesaria s en el inicializador del esquema y estas propiedades se instancian con los valores predeterminados. El valor de la métrica puede ser `0` si se ha realizado correctamente o 1 en caso de fallo. Dimensiones: ComponentTypeId, Workspace Id. Unidades: recuento

Métrica	Descripción
`EntityCreationFailure`	Esta métrica indica si la creación de la entidad se ha realizado correctamente. La métrica se publica cuando una entidad está en el estado `CREATING`. Esto ocurre cuando se crea una entidad con un componente. El valor de la métrica puede ser `0` si se ha realizado correctamente o 1 en caso de fallo. Dimensiones: EntityName, EntityId, WorkspaceId. Unidades: recuento
`EntityUpdateFailure`	Esta métrica indica si la actualización de la entidad se ha realizado correctamente. La métrica se publica cuando una entidad está en el estado `UPDATING`. Esto ocurre cuando se actualiza una entidad. El valor de la métrica puede ser `0` si se ha realizado correctamente o 1 en caso de fallo. Dimensiones: EntityName, EntityId, WorkspaceId. Unidades: recuento

Métrica	Descripción
`EntityDeletionFailure`	Esta métrica indica si la eliminación de la entidad se ha realizado correctamente. La métrica se publica cuando una entidad está en el estado `DELETING`. Esto ocurre cuando se elimina una entidad. El valor de la métrica puede ser `0` si se ha realizado correctamente o 1 en caso de fallo. Dimensiones: EntityName, EntityId, WorkspaceId. Unidades: recuento

> **Tip**
>
> Las métricas se publican en el espacio de nombres de `AWS/IoTTwinMaker`.

Registrar llamadas a la API de AWS IoT TwinMaker con AWS CloudTrail

AWS IoT TwinMaker 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 TwinMaker. CloudTrail captura las llamadas a la API de AWS IoT TwinMaker como eventos. Las llamadas capturadas incluyen las llamadas desde la consola de AWS IoT TwinMaker y las llamadas desde el código a las operaciones de la API de AWS IoT TwinMaker. 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 TwinMaker. 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 TwinMaker, la dirección IP desde la que se realizó, quién la realizó y cuándo, etc.

Para obtener más información acerca de CloudTrail, consulte la AWS CloudTrailGuía del usuario de .

Información de AWS IoT TwinMaker en CloudTrail

CloudTrail se habilita automáticamente cuando crea su cuenta de AWS. CloudTrail registra la actividad de evento de soporte que se produce en AWS IoT TwinMaker junto con otros eventos de servicio de AWS en 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 TwinMaker, 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 se crea un registro de seguimiento en la consola, el registro de seguimiento se aplica a todas las regiones de AWS. CloudTrail 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 los siguientes temas:

- 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 registro de CloudTrail de varias regiones y Recibir archivos de registro de CloudTrail de varias cuentas

CloudTrail registra todas la mayoría de las operaciones de AWS IoT TwinMaker y se documentan en la Referencia de la API de AWS IoT TwinMaker.

CloudTrail no registra las siguientes operaciones del plano de datos:

- GetPropertyValue
- GetPropertyValueHistory
- BatchPutPropertyValues

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 .

- 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.

Seguridad en AWS IoT TwinMaker

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

La seguridad es una responsabilidad compartida entre AWS usted 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 Nube de AWS. AWS también le proporciona servicios que puede utilizar de forma segura. Los auditores externos prueban y verifican periódicamente la eficacia de nuestra seguridad como parte de los AWS programas de de . Para obtener más información sobre los programas de cumplimiento aplicables AWS IoT TwinMaker, consulte AWS Servicios incluidos en el ámbito de aplicación por programa de conformidad y AWS servicios incluidos .
- Seguridad en la nube: su responsabilidad viene determinada por el AWS servicio que utilice. También eres responsable de otros factores, incluida la confidencialidad de los datos, los requisitos de la empresa y la legislación y la normativa aplicables.

Esta documentación le ayuda a comprender cómo aplicar el modelo de responsabilidad compartida cuando se utiliza AWS IoT TwinMaker. Los siguientes temas muestran cómo configurarlo AWS IoT TwinMaker para cumplir sus objetivos de seguridad y conformidad. También aprenderá a utilizar otros AWS servicios que le ayudan a supervisar y proteger sus AWS IoT TwinMaker recursos.

Temas

- Protección de datos en AWS IoT TwinMaker
- Identity and Access Management para AWS IoT TwinMaker
- AWS IoT TwinMaker y puntos finales de VPC de interfaz ()AWS PrivateLink
- Validación de conformidad para AWS IoT TwinMaker
- Resiliencia en AWS IoT TwinMaker
- Seguridad de la infraestructura en AWS IoT TwinMaker

Protección de datos en AWS IoT TwinMaker

El modelo de responsabilidad AWS compartida modelo se aplica a la protección de datos en AWS IoT TwinMaker. Como se describe en este modelo, AWS es responsable de proteger la infraestructura global que ejecuta todos los Nube de AWS. Eres responsable de mantener el control sobre el contenido alojado en esta infraestructura. También eres responsable de las tareas de administración y configuración de seguridad para los Servicios de AWS que utiliza. Para obtener más información sobre la privacidad de los datos, consulta las Preguntas frecuentes sobre la privacidad de datos. Para obtener información sobre la protección de datos en Europa, consulta la publicación de blog sobre el Modelo de responsabilidad compartida de AWS y GDPR en el Blog de seguridad de AWS .

Con fines de protección de datos, le recomendamos que proteja Cuenta de AWS las credenciales y configure los usuarios individuales con AWS IAM Identity Center o AWS Identity and Access Management (IAM). De esta manera, solo se otorgan a cada usuario los permisos necesarios para cumplir sus obligaciones laborales. También recomendamos proteger sus datos de la siguiente manera:

- Utiliza la autenticación multifactor (MFA) en cada cuenta.
- Se utiliza SSL/TLS para comunicarse con AWS los recursos. Se recomienda el uso de TLS 1.2 y recomendamos TLS 1.3.
- Configure la API y el registro de actividad de los usuarios con AWS CloudTrail. Para obtener información sobre el uso de CloudTrail senderos para capturar AWS actividades, consulte Cómo trabajar con CloudTrail senderos en la Guía del AWS CloudTrail usuario.
- Utilice soluciones de AWS cifrado, junto con todos los controles de seguridad predeterminados que contienen Servicios de AWS.
- Utiliza servicios de seguridad administrados avanzados, como Amazon Macie, que lo ayuden a detectar y proteger los datos confidenciales almacenados en Amazon S3.
- Si necesita módulos criptográficos validados por FIPS 140-3 para acceder a AWS través de una interfaz de línea de comandos o una API, utilice un punto final FIPS. Para obtener más información sobre los puntos de conexión de FIPS disponibles, consulta Estándar de procesamiento de la información federal (FIPS) 140-3.

Se recomienda encarecidamente no introducir nunca información confidencial o sensible, como por ejemplo, direcciones de correo electrónico de clientes, en etiquetas o campos de formato libre, tales como el campo Nombre. Esto incluye cuando trabaja con AWS IoT TwinMaker o Servicios de AWS

utiliza la consola, la API o AWS CLI AWS SDKs Cualquier dato que ingrese en etiquetas o campos de texto de formato libre utilizados para nombres se puede emplear para los registros de facturación o diagnóstico. Si proporciona una URL a un servidor externo, recomendamos encarecidamente que no incluya la información de las credenciales en la URL para validar la solicitud para ese servidor.

Cifrado en reposo

AWS IoT TwinMaker almacena la información de tu espacio de trabajo en un bucket de Amazon S3 que el servicio crea para ti, si así lo deseas. El bucket que el servicio crea para usted tiene activado el cifrado predeterminado del lado del servidor. Si decide usar su propio bucket de Amazon S3 al crear un nuevo espacio de trabajo, le recomendamos que habilite el cifrado predeterminado del lado del servidor. Para obtener más información acerca del cifrado predeterminado de Amazon S3, consulte Ajuste del comportamiento predeterminado del cifrado del servidor para para buckets de Amazon S3.

Cifrado en tránsito

Todos los datos que se envían a AWS IoT TwinMaker se envían a través de una conexión TLS mediante el protocolo HTTPS, por lo que son seguros de forma predeterminada mientras están en tránsito.

> **Note**
>
> Le recomendamos que utilice HTTPS en las direcciones de bucket de Amazon S3 como control para aplicar el cifrado en tránsito cuando AWS IoT TwinMaker interactúe con un bucket de Amazon S3. Para obtener más información acerca de los buckets de Amazon S3, consulte Creación, configuración y trabajo con buckets de Amazon S3.

Identity and Access Management para AWS IoT TwinMaker

AWS Identity and Access Management (IAM) es una herramienta Servicio de AWS 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 TwinMaker La IAM es una Servicio de AWS opción que puede utilizar sin coste adicional.

Temas

- Público
- Autenticación con identidades
- Administración de acceso mediante políticas
- ¿Cómo AWS IoT TwinMaker funciona con IAM
- Ejemplos de políticas basadas en la identidad para AWS IoT TwinMaker
- Solución de problemas de AWS IoT TwinMaker identidad y acceso
- Cómo utilizar roles vinculados a servicios de AWS IoT TwinMaker
- AWS políticas gestionadas para AWS IoT TwinMaker

Público

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

Usuario del servicio: si utiliza el AWS IoT TwinMaker servicio para realizar su trabajo, el administrador le proporcionará las credenciales y los permisos que necesita. A medida que vaya utilizando más AWS IoT TwinMaker funciones para realizar su trabajo, es posible que necesite permisos adicionales. Entender cómo se administra el acceso puede ayudarle a solicitar los permisos correctos al administrador. Si no puede acceder a una característica en AWS IoT TwinMaker, consulte Solución de problemas de AWS IoT TwinMaker identidad y acceso.

Administrador de servicios: si estás a cargo de AWS IoT TwinMaker los recursos de tu empresa, probablemente tengas acceso total a ellos AWS IoT TwinMaker. Su trabajo consiste en determinar a qué AWS IoT TwinMaker funciones y recursos deben acceder los usuarios del servicio. Luego, debe enviar solicitudes a su gestionador 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 TwinMaker, consulte¿Cómo AWS IoT TwinMaker 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 IoT TwinMaker. Para ver ejemplos de políticas AWS IoT TwinMaker basadas en la identidad que puede utilizar en IAM, consulte. Ejemplos de políticas basadas en la identidad para AWS IoT TwinMaker

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 gestionador 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 la firma de solicitudes, consulte AWS Signature Versión 4 para solicitudes 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 Autenticación multifactor AWS en IAM en 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 utiliza 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, consulta Tareas que requieren credenciales de usuario raíz en la Guía del usuario de IAM.

Identidad federada

Como práctica recomendada, exija a los usuarios humanos, incluidos los que requieren acceso de administrador, que utilicen la federación con un proveedor de identidades para acceder Servicios de AWS mediante credenciales temporales.

Una identidad federada es un usuario del directorio de usuarios de su empresa, un proveedor de identidades web AWS Directory Service, el directorio del Centro de Identidad o cualquier usuario al que acceda Servicios de AWS mediante las credenciales proporcionadas a través de una fuente de identidad. Cuando las identidades federadas acceden Cuentas de AWS, asumen funciones y las funciones proporcionan credenciales temporales.

Para una administración de acceso centralizada, le recomendamos que utiliza AWS IAM Identity Center. Puede crear usuarios y grupos en el Centro de identidades de IAM o puede conectarse y sincronizarse con un conjunto de usuarios y grupos de su propia fuente de identidad para usarlos en todas sus Cuentas de AWS aplicaciones. Para obtener más información, consulta ¿Qué es el Centro de identidades de IAM? en la Guía del usuario de AWS IAM Identity Center .

Usuarios y grupos de IAM

Un usuario de IAM es una identidad 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 puedes iniciar sesión como grupo. Puedes usar los grupos para especificar permisos para varios usuarios a la vez. Los grupos facilitan la administración de los permisos para grandes conjuntos de usuarios. Por ejemplo, puede asignar un nombre a un grupo IAMAdminsy concederle permisos 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 de larga duración permanentes; no obstante, los roles proporcionan credenciales temporales. Para obtener más información, consulte Casos de uso para usuarios de IAM 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 persona determinada. Para asumir temporalmente un rol de IAM en el AWS Management Console, puede cambiar de un rol de usuario a uno de IAM (consola). Puedes asumir un rol llamando a una operación de AWS API AWS CLI o usando una URL personalizada. Para más información sobre los métodos para el uso de roles, consulta Métodos para asumir un rol 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 de federación, consulte Crear un rol para un proveedor de identidad de terceros (federación) en la Guía de usuario de IAM. Si utiliza el 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, consulta 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, consulta Acceso a recursos entre cuentas en IAM en la Guía del usuario de IAM.
- Acceso entre servicios: algunos Servicios de AWS utilizan funciones en otros Servicios de AWS. Por ejemplo, cuando realizas una llamada en un servicio, es habitual 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 AWS ellas, se te considera principal. 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: puedes usar un rol de IAM para administrar las credenciales temporales de las aplicaciones que se ejecutan en una EC2 instancia y realizan AWS CLI solicitudes a la AWS API. Esto es preferible a almacenar las claves de acceso en la EC2 instancia. Para asignar un AWS rol a una EC2 instancia 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 que los programas que se ejecutan en la EC2 instancia obtengan credenciales temporales. Para obtener más información, consulte Usar un rol de IAM para conceder permisos a las aplicaciones que se ejecutan en EC2 instancias de Amazon 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 utiliza 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 obtener más información sobre cómo elegir una política administrada o una política insertada, consulte Elegir entre políticas administradas y políticas insertadas en la Guía del usuario de IAM.

Políticas basadas en recursos

Las políticas basadas en recursos son documentos de política JSON que se asocian a un recurso. Los ejemplos de políticas basadas en recursos son las políticas de confianza de roles de IAM y las políticas de bucket de Amazon S3. En los servicios que admiten políticas basadas en recursos, los administradores de servicios pueden utilizarlos para controlar el acceso a un recurso específico. Para el recurso al que se asocia la política, la política define qué acciones puede realizar una entidad principal especificada en ese recurso y en qué condiciones. Debe especificar una entidad principal en una política en función de recursos. Los principales pueden incluir cuentas, usuarios, roles, usuarios federados o. Servicios de AWS

Las políticas basadas en recursos son políticas insertadas que se encuentran en ese servicio. No puedes usar políticas AWS gestionadas de IAM en una política basada en recursos.

Listas de control de acceso () ACLs

Las listas de control de acceso (ACLs) controlan qué responsables (miembros de la cuenta, usuarios o roles) tienen permisos para acceder a un recurso. ACLs son similares a las políticas basadas en recursos, aunque no utilizan el formato de documento de políticas JSON.

Amazon S3 y Amazon VPC son ejemplos de servicios compatibles. AWS WAF ACLs Para obtener más información ACLs, consulte la descripción general de la lista de control de acceso (ACL) en la Guía para desarrolladores de Amazon Simple Storage Service.

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). Puedes 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 (SCPs): SCPs 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 administrar de forma centralizada varios de los Cuentas de AWS que son propiedad de su empresa. Si habilitas todas las funciones de una organización, puedes aplicar políticas de control de servicios (SCPs) a una o a todas tus 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 sobre Organizations SCPs, consulte las políticas de control de servicios en la Guía del AWS Organizations usuario.
- Políticas de control de recursos (RCPs): RCPs son políticas de JSON que puedes usar para establecer los permisos máximos disponibles para los recursos de tus cuentas sin actualizar las políticas de IAM asociadas a cada recurso que poseas. El RCP limita los permisos de los recursos en las cuentas de los miembros y puede afectar a los permisos efectivos de las identidades, incluidos los permisos Usuario raíz de la cuenta de AWS, independientemente de si pertenecen a su organización. Para obtener más información sobre Organizations e RCPs incluir una lista de

Servicios de AWS ese apoyo RCPs, consulte Políticas de control de recursos (RCPs) en la Guía del AWS Organizations usuario.

- 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 se AWS determina 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 TwinMaker funciona con IAM

Antes de utilizar IAM para gestionar el acceso AWS IoT TwinMaker, infórmese sobre las funciones de IAM disponibles para su uso. AWS IoT TwinMaker

Funciones de IAM que puede utilizar con AWS IoT TwinMaker

Característica de IAM	AWS IoT TwinMaker soporte
Políticas basadas en identidades	Sí
Políticas basadas en recursos	No
Acciones de políticas	Sí
Recursos de políticas	Sí
Claves de condición de política	Sí
ACLs	No
ABAC (etiquetas en políticas)	Parcial

Característica de IAM	AWS IoT TwinMaker soporte
Credenciales temporales	Sí
Permisos de entidades principales	Sí
Roles de servicio	Sí
Roles vinculados al servicio	No

Para obtener una visión general de cómo AWS IoT TwinMaker funcionan otros AWS servicios con la mayoría de las funciones de IAM, consulte AWS los servicios que funcionan con IAM en la Guía del AWS IAM Identity Center usuario.

Políticas basadas en la identidad para AWS IoT TwinMaker

Compatibilidad con las políticas basadas en identidad: sí

Las políticas basadas en identidad son documentos de políticas de permisos JSON que puede asociar a una identidad, como un usuario de IAM, un grupo de usuarios o un rol. Estas políticas controlan qué acciones pueden realizar los usuarios y los roles, en qué recursos y en qué condiciones. Para obtener más información sobre cómo crear una política basada en identidad, consulte Creación de políticas de IAM en la Guía del usuario de IAM.

Con las políticas basadas en identidades de IAM, puede especificar las acciones y los recursos permitidos o denegados, así como las condiciones en las que se permiten o deniegan las acciones. No es posible especificar la entidad principal en una política basada en identidad porque se aplica al usuario o rol al que está asociada. Para obtener más información sobre los elementos que puede utilizar en una política de JSON, consulte Referencia de los elementos de las políticas de JSON de IAM en la Guía del usuario de IAM.

Ejemplos de políticas basadas en la identidad para AWS IoT TwinMaker

Para ver ejemplos de políticas AWS IoT TwinMaker basadas en la identidad, consulte. Ejemplos de políticas basadas en la identidad para AWS IoT TwinMaker

Políticas basadas en recursos incluidas AWS IoT TwinMaker

Admite políticas basadas en recursos: no

Las políticas basadas en recursos son documentos de política JSON que se asocian a un recurso. Los ejemplos de políticas basadas en recursos son las políticas de confianza de roles de IAM y las políticas de bucket de Amazon S3. En los servicios que admiten políticas basadas en recursos, los administradores de servicios pueden utilizarlos para controlar el acceso a un recurso específico. Para el recurso al que se asocia la política, la política define qué acciones puede realizar una entidad principal especificada en ese recurso y en qué condiciones. Debe especificar una entidad principal en una política en función de recursos. Los principales pueden incluir cuentas, usuarios, roles, usuarios federados o. Servicios de AWS

Para habilitar el acceso entre cuentas, puede especificar toda una cuenta o entidades de IAM de otra cuenta como la entidad principal de una política en función de recursos. Añadir a una política en función de recursos una entidad principal entre cuentas es solo una parte del establecimiento de una relación de confianza. Cuando el principal y el recurso son diferentes Cuentas de AWS, el administrador de IAM de la cuenta de confianza también debe conceder a la entidad principal (usuario o rol) permiso para acceder al recurso. Para conceder el permiso, adjunte la entidad a una política basada en identidad. Sin embargo, si la política basada en recursos concede acceso a una entidad principal de la misma cuenta, no es necesaria una política basada en identidad adicional. Para obtener más información, consulte Cross account resource access in IAM en la Guía del usuario de IAM.

Acciones políticas para AWS IoT TwinMaker

Compatibilidad con las acciones de políticas: sí

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.

El elemento `Action` de una política JSON describe las acciones que puede utilizar para conceder o denegar el acceso en una política. Las acciones políticas suelen tener el mismo nombre que la operación de AWS API asociada. Hay algunas excepciones, como acciones de solo permiso que no tienen una operación de API coincidente. También hay algunas operaciones que requieren varias acciones en una política. Estas acciones adicionales se denominan acciones dependientes.

Incluya acciones en una política para conceder permisos y así llevar a cabo la operación asociada.

Para ver una lista de AWS IoT TwinMaker acciones, consulta las acciones definidas AWS IoT TwinMaker en la Referencia de autorización del servicio.

Las acciones políticas AWS IoT TwinMaker utilizan el siguiente prefijo antes de la acción:

```
iottwinmaker
```

Para especificar varias acciones en una única instrucción, sepárelas con comas.

```
"Action": [
   "iottwinmaker:action1",
   "iottwinmaker:action2"
         ]
```

Para ver ejemplos de políticas AWS IoT TwinMaker basadas en la identidad, consulte. Ejemplos de políticas basadas en la identidad para AWS IoT TwinMaker

Recursos de políticas para AWS IoT TwinMaker

Compatibilidad con los recursos de políticas: sí

Los administradores pueden usar las políticas de AWS JSON para especificar quién tiene acceso a qué. Es decir, qué entidad principal puedes realizar acciones en qué recursos y en qué condiciones.

El elemento `Resource` de la política JSON especifica el objeto u objetos a los que se aplica la acción. Las instrucciones deben contener un elemento `Resource` o `NotResource`. Como práctica recomendada, especifique un recurso utilizando el Nombre de recurso de Amazon (ARN). Puedes hacerlo para acciones que admitan un tipo de recurso específico, conocido como permisos de nivel de recurso.

Para las acciones que no admiten permisos de nivel de recurso, como las operaciones de descripción, utiliza un carácter comodín (*) para indicar que la instrucción se aplica a todos los recursos.

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

Para ver una lista de los tipos de AWS IoT TwinMaker recursos y sus tipos ARNs, consulte los recursos definidos AWS IoT TwinMaker en la Referencia de autorización de servicios. Para obtener información sobre las acciones con las que puede especificar el ARN de cada recurso, consulte Acciones definidas por AWS IoT TwinMaker.

Para ver ejemplos de políticas AWS IoT TwinMaker basadas en la identidad, consulte. Ejemplos de políticas basadas en la identidad para AWS IoT TwinMaker

Claves de condición de la política para AWS IoT TwinMaker

Compatibilidad con claves de condición de políticas específicas del servicio: sí

Los administradores pueden usar las políticas de AWS JSON para especificar quién tiene acceso a qué. Es decir, qué entidad principal puedes realizar acciones en qué recursos y en qué condiciones.

El elemento `Condition` (o bloque de `Condition`) permite especificar condiciones en las que entra en vigor una instrucción. El elemento `Condition` es opcional. Puedes crear expresiones condicionales que utilizan operadores de condición, tales como igual o menor que, para que la condición de la política coincida con los valores de la solicitud.

Si especifica varios elementos de `Condition` en una instrucción o varias claves en un único elemento de `Condition`, AWS las evalúa mediante una operación `AND` lógica. Si especifica varios valores para una única clave de condición, AWS evalúa la condición mediante una `OR` operación lógica. Se deben cumplir todas las condiciones antes de que se concedan los permisos de la instrucción.

También puedes utilizar variables de marcador de posición al especificar condiciones. Por ejemplo, puedes conceder un permiso de usuario de IAM para acceder a un recurso solo si está etiquetado con su nombre de usuario de IAM. Para más información, consulta Elementos de la política de IAM: variables y etiquetas en la Guía del usuario de IAM.

AWS admite claves de condición globales y claves de condición específicas del servicio. Para ver todas las claves de condición AWS globales, consulte las claves de contexto de condición AWS globales en la Guía del usuario de IAM.

Para ver una lista de claves de AWS IoT TwinMaker condición, consulte las claves de condición AWS IoT TwinMaker en la Referencia de autorización de servicio. Para saber con qué acciones y recursos puede utilizar una clave de condición, consulte Acciones definidas por AWS IoT TwinMaker.

Para ver ejemplos de políticas AWS IoT TwinMaker basadas en la identidad, consulte. Ejemplos de políticas basadas en la identidad para AWS IoT TwinMaker

Listas de control de acceso () ACLs en AWS IoT TwinMaker

Soporta ACLs: No

Las listas de control de acceso (ACLs) controlan qué directores (miembros de la cuenta, usuarios o roles) tienen permisos para acceder a un recurso. ACLs son similares a las políticas basadas en recursos, aunque no utilizan el formato de documento de políticas JSON.

Control de acceso basado en atributos (ABAC) con AWS IoT TwinMaker

Compatibilidad con ABAC (etiquetas en las políticas): parcial

El control de acceso basado en atributos (ABAC) es una estrategia de autorización que define permisos en función de atributos. En AWS, estos atributos se denominan etiquetas. Puede adjuntar etiquetas a las entidades de IAM (usuarios o roles) y a muchos AWS recursos. El etiquetado de entidades y recursos es el primer paso de ABAC. A continuación, designa las políticas de ABAC para permitir operaciones cuando la etiqueta de la entidad principal coincida con la etiqueta del recurso al que se intenta acceder.

ABAC es útil en entornos que crecen con rapidez y ayuda en situaciones en las que la administración de las políticas resulta engorrosa.

Para controlar el acceso en función de etiquetas, debe proporcionar información de las etiquetas en el elemento de condición de una política utilizando las claves de condición `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` o `aws:TagKeys`.

Si un servicio admite las tres claves de condición para cada tipo de recurso, el valor es Sí para el servicio. Si un servicio admite las tres claves de condición solo para algunos tipos de recursos, el valor es Parcial.

Para obtener más información sobre ABAC, consulte Definición de permisos con la autorización de ABAC en la Guía del usuario de IAM. Para ver un tutorial con los pasos para configurar ABAC, consulte Uso del control de acceso basado en atributos (ABAC) en la Guía del usuario de IAM.

Uso de credenciales temporales con AWS IoT TwinMaker

Compatibilidad con credenciales temporales: sí

Algunas Servicios de AWS no funcionan cuando inicias sesión con credenciales temporales. Para obtener información adicional, incluidas las que Servicios de AWS funcionan con credenciales temporales, consulta Cómo Servicios de AWS funcionan con IAM en la Guía del usuario de IAM.

Utiliza credenciales temporales si inicia sesión en ellas AWS Management Console mediante cualquier método excepto un nombre de usuario y una contraseña. Por ejemplo, cuando accedes AWS mediante el enlace de inicio de sesión único (SSO) de tu empresa, ese proceso crea automáticamente credenciales temporales. También crea credenciales temporales de forma automática cuando inicia sesión en la consola como usuario y luego cambia de rol. Para obtener más información sobre el cambio de roles, consulte Cambio de un usuario a un rol de IAM (consola) en la Guía del usuario de IAM.

Puedes crear credenciales temporales manualmente mediante la AWS CLI API o. AWS A continuación, puede utilizar esas credenciales temporales para acceder AWS. AWS recomienda generar credenciales temporales de forma dinámica en lugar de utilizar claves de acceso a largo plazo. Para obtener más información, consulte Credenciales de seguridad temporales en IAM.

Permisos principales entre servicios para AWS IoT TwinMaker

Admite sesiones de acceso directo (FAS): sí

Cuando utilizas un usuario o un rol de IAM para realizar acciones en él AWS, se te considera principal. Cuando utiliza algunos servicios, es posible que realice una acción que desencadene otra acción en un servicio diferente. FAS utiliza los permisos del principal que llama y los que solicita Servicio de AWS para realizar solicitudes a los servicios descendentes. Servicio de AWS 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.

Roles de servicio para AWS IoT TwinMaker

Compatibilidad con roles de servicio: sí

Un rol de servicio es un rol de IAM que asume un servicio para realizar acciones en su nombre. Un administrador de IAM puede crear, modificar y eliminar un rol de servicio desde IAM. Para obtener más información, consulte Creación de un rol para delegar permisos a un Servicio de AWS en la Guía del usuario de IAM.

> **Warning**
>
> Cambiar los permisos de un rol de servicio puede interrumpir AWS IoT TwinMaker la funcionalidad. Edite las funciones de servicio solo cuando se AWS IoT TwinMaker proporcionen instrucciones para hacerlo.

Funciones vinculadas al servicio para AWS IoT TwinMaker

Compatibilidad con roles vinculados al servicio: no

Un rol vinculado a un servicio es un tipo de rol de servicio que está vinculado a un. Servicio de AWS El servicio puede asumir el rol para realizar una acción en su nombre. Los roles vinculados al servicio

aparecen en usted Cuenta de AWS y son propiedad del servicio. Un administrador de IAM puedes ver, pero no editar, los permisos de los roles vinculados a servicios.

Para más información sobre cómo crear o administrar roles vinculados a servicios, consulta Servicios de AWS que funcionan con IAM. Busque un servicio en la tabla que incluya `Yes` en la columna Rol vinculado a un servicio. Seleccione el vínculo Sí para ver la documentación acerca del rol vinculado a servicios para ese servicio.

Ejemplos de políticas basadas en la identidad para AWS IoT TwinMaker

De forma predeterminada, los usuarios y roles no tienen permiso para crear, ver ni modificar recursos de AWS IoT TwinMaker . Tampoco pueden realizar tareas mediante la AWS Management Console, AWS Command Line Interface (AWS CLI) o AWS la API. 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 puedes añadir las políticas de IAM a roles y los usuarios puedes asumirlos.

Para obtener información acerca de cómo crear una política basada en identidades de IAM mediante el uso de estos documentos de políticas JSON de ejemplo, consulte Creación de políticas de IAM (consola) en la Guía del usuario de IAM.

Para obtener más información sobre las acciones y los tipos de recursos definidos AWS IoT TwinMaker, incluido el formato ARNs de cada uno de los tipos de recursos, consulte las claves de condición, recursos y acciones de AWS IoT TwinMaker la Referencia de autorización de servicios.

Temas

- Prácticas recomendadas sobre las políticas
- Mediante la consola de AWS IoT TwinMaker
- Cómo permitir a los usuarios consultar sus propios permisos

Prácticas recomendadas sobre las políticas

Las políticas basadas en la identidad determinan si alguien puede crear AWS IoT TwinMaker 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 con las políticas AWS administradas y avance hacia los permisos con privilegios mínimos: para empezar a conceder permisos a sus usuarios y cargas de trabajo, utilice las políticas AWS administradas que otorgan permisos para muchos casos de uso comunes. Están disponibles en su. Cuenta de AWS Le recomendamos que reduzca aún más los permisos definiendo políticas administradas por el AWS cliente que sean específicas para sus casos de uso. Con el fin de obtener más información, consulta las políticas administradas por AWS o las políticas administradas por AWS para funciones de tarea en la Guía de usuario de IAM.
- Aplique permisos de privilegio mínimo: cuando establezca permisos con políticas de IAM, conceda solo los permisos necesarios para realizar una tarea. Para ello, debe definir las acciones que se pueden llevar a cabo en determinados recursos en condiciones específicas, también conocidos como permisos de privilegios mínimos. Con el fin de obtener más información sobre el uso de IAM para aplicar permisos, consulta Políticas y permisos en IAM en la Guía del usuario de IAM.
- Utilice condiciones en las políticas de IAM para restringir aún más el acceso: puede agregar una condición a sus políticas para limitar el acceso a las acciones y los recursos. Por ejemplo, puede escribir una condición de políticas para especificar que todas las solicitudes deben enviarse utilizando SSL. También puedes usar condiciones para conceder el acceso a las acciones del servicio si se utilizan a través de una acción específica Servicio de AWS, por ejemplo AWS CloudFormation. Para obtener más información, consulta Elementos de la política de JSON de IAM: Condición en la Guía del usuario de IAM.
- Utiliza el analizador de acceso de IAM para validar las políticas de IAM con el fin de garantizar la seguridad y funcionalidad de los permisos: el analizador de acceso de IAM valida políticas nuevas y existentes para que respeten el lenguaje (JSON) de las políticas de IAM y las prácticas recomendadas de IAM. El analizador de acceso de IAM proporciona más de 100 verificaciones de políticas y recomendaciones procesables para ayudar a crear políticas seguras y funcionales. Para más información, consulte Validación de políticas con el Analizador de acceso de IAM en la Guía del usuario de IAM.
- Requerir autenticación multifactor (MFA): si tiene un escenario que requiere usuarios de IAM o un usuario raíz en Cuenta de AWS su cuenta, active la MFA para mayor seguridad. Para exigir la MFA cuando se invoquen las operaciones de la API, añada condiciones de MFA a sus políticas. Para más información, consulte Acceso seguro a la API con MFA en la Guía del usuario de IAM.

Para obtener más información sobre las prácticas recomendadas de IAM, consulte Prácticas recomendadas de seguridad en IAM en la Guía del usuario de IAM.

Mediante la consola de AWS IoT TwinMaker

Para acceder a la AWS IoT TwinMaker consola, debe tener un conjunto mínimo de permisos. Estos permisos deben permitirle enumerar y ver detalles sobre los AWS IoT TwinMaker recursos de su cuenta Cuenta de AWS. Si crea una política basada en identidades que sea más restrictiva que el mínimo de permisos necesarios, la consola no funcionará del modo esperado para las entidades (usuarios o roles) que tengan esa política.

No es necesario que concedas permisos mínimos de consola a los usuarios que solo realicen llamadas a la API AWS CLI o a la AWS API. En su lugar, permite el acceso únicamente a las acciones que coincidan con la operación de API que intentan realizar.

Para garantizar que los usuarios y los roles puedan seguir utilizando la AWS IoT TwinMaker consola, asocie también la AWS IoT TwinMaker `ConsoleAccess` política `ReadOnly` AWS gestionada a las entidades. Para obtener más información, consulte Adición de permisos a un usuario en la Guía del usuario de AWS IAM Identity Center .

Cómo permitir a los usuarios consultar sus propios permisos

En este ejemplo, se muestra cómo podría crear una política que permita a los usuarios de IAM ver las políticas gestionadas 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": "*"
        }
    ]
}
```

Solución de problemas de AWS IoT TwinMaker identidad y acceso

Utilice la siguiente información como ayuda para diagnosticar y solucionar los problemas habituales que pueden surgir al trabajar con un AWS IoT TwinMaker IAM.

Temas

- No estoy autorizado a realizar ninguna acción en AWS IoT TwinMaker
- No estoy autorizado a realizar tareas como: PassRole
- Quiero permitir que personas ajenas a mí accedan Cuenta de AWS a mis AWS IoT TwinMaker recursos

No estoy autorizado a realizar ninguna acción en AWS IoT TwinMaker

Si recibe un error que indica que no tiene autorización para realizar una acción, las políticas se deben actualizar para permitirle realizar la acción.

En el siguiente ejemplo, el error se produce cuando el usuario de IAM `mateojackson` intenta utilizar la consola para consultar los detalles acerca de un recurso ficticio *`my-example-widget`*, pero no tiene los permisos ficticios `iottwinmaker:`*`GetWidget`*.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 iottwinmaker:GetWidget on resource: my-example-widget
```

En este caso, la política del usuario `mateojackson` debe actualizarse para permitir el acceso al recurso *`my-example-widget`* mediante la acción `iottwinmaker:`*`GetWidget`*.

Si necesita ayuda, póngase en contacto con su AWS administrador. El gestionador es la persona que le proporcionó las credenciales de inicio de sesión.

No estoy autorizado a realizar tareas como: 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 TwinMaker.

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 TwinMaker. 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 gestionador 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 TwinMaker 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 respaldan las políticas basadas en recursos o las listas de control de acceso (ACLs), puedes usar esas políticas para permitir que las personas accedan a tus recursos.

Para obtener más información, consulte lo siguiente:

- Para saber si AWS IoT TwinMaker es compatible con estas funciones, consulte. ¿Cómo AWS IoT TwinMaker 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 usuario de su propiedad Cuenta de AWS 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, consulta Proporcionar acceso a usuarios autenticados externamente (identidad federada) en la Guía del usuario de IAM.
- Para conocer sobre la diferencia entre las políticas basadas en roles y en recursos para el acceso entre cuentas, consulte Acceso a recursos entre cuentas en IAM en la Guía del usuario de IAM.

Cómo utilizar roles vinculados a servicios de AWS IoT TwinMaker

AWS IoT TwinMaker usa roles vinculados al AWS Identity and Access Management servicio (IAM). Un rol vinculado a un servicio es un tipo único de rol de IAM al que se vincula directamente. AWS IoT TwinMaker Los roles vinculados al servicio están predefinidos AWS IoT TwinMaker e incluyen todos los permisos que el servicio requiere para llamar a otros AWS servicios en su nombre.

Un rol vinculado a un servicio facilita la configuración AWS IoT TwinMaker , ya que no es necesario añadir manualmente los permisos necesarios. AWS IoT TwinMaker define los permisos de sus funciones vinculadas al servicio y, a menos que se defina lo contrario, solo AWS IoT TwinMaker puede asumir sus funciones. Los permisos definidos incluyen las políticas de confianza y de permisos, y que la política de permisos no se pueda adjuntar a ninguna otra entidad de IAM.

Solo es posible eliminar un rol vinculado a un servicio después de eliminar sus recursos relacionados. Esto protege sus AWS IoT TwinMaker recursos porque no puede eliminar inadvertidamente el permiso de acceso a los recursos.

Para obtener información sobre otros servicios que admiten funciones vinculadas a servicios, consulte los AWS servicios que funcionan con IAM y busque los servicios con la palabra Sí en la columna Funciones vinculadas a servicios. Elija una opción Sí con un enlace para ver la documentación acerca del rol vinculado al servicio en cuestión.

Permisos de roles vinculados al servicio para AWS IoT TwinMaker

AWS IoT TwinMaker utiliza el rol vinculado al servicio denominado AWSServiceRoleForIoTTwinMaker: permite llamar AWS IoT TwinMaker a otros AWS servicios y sincronizar sus recursos en tu nombre.

El rol vinculado al servicio de AWSService RoleForIo TTwin Maker confía en los siguientes servicios para que asuman el rol:

- `iottwinmaker.amazonaws.com`

La política de permisos de roles denominada AWSIo TTwin MakerServiceRolePolicy permite AWS IoT TwinMaker realizar las siguientes acciones en los recursos especificados:

- Acción: `iotsitewise:DescribeAsset, iotsitewise:ListAssets, iotsitewise:DescribeAssetModel, and iotsitewise:ListAssetModels, iottwinmaker:GetEntity, iottwinmaker:CreateEntity, iottwinmaker:UpdateEntity, iottwinmaker:DeleteEntity, iottwinmaker:ListEntities, iottwinmaker:GetComponentType, iottwinmaker:CreateComponentType, iottwinmaker:UpdateComponentType, iottwinmaker:DeleteComponentType, iottwinmaker:ListComponentTypes` en `all your iotsitewise asset and asset-model resources`

Debe configurar los permisos para permitir a sus usuarios, grupos o funciones, crear, editar o eliminar la descripción de un rol vinculado al servicio. Para obtener más información, consulte Permisos de roles vinculados a servicios en la Guía del usuario de IAM.

Crear un rol vinculado a un servicio para AWS IoT TwinMaker

No necesita crear manualmente un rol vinculado a servicios. Al sincronizar AWS IoT SiteWise los activos y los modelos de activos (sincronización de activos) en la AWS Management Console, la o la AWS API AWS CLI, se AWS IoT TwinMaker crea automáticamente el rol vinculado al servicio.

Si elimina este rol vinculado a servicios y necesita crearlo de nuevo, puede utilizar el mismo proceso para volver a crear el rol en su cuenta. Al sincronizar AWS IoT SiteWise los activos y los modelos de activos (sincronización de activos), vuelve a AWS IoT TwinMaker crear automáticamente el rol vinculado al servicio.

También puede usar la consola de IAM para crear un rol vinculado a un servicio con el caso de uso «IoT TwinMaker : rol administrado». En la API AWS CLI o en la AWS API, cree una función vinculada a un servicio con el nombre del servicio. `iottwinmaker.amazonaws.com` Para obtener más información, consulte Crear un rol vinculado a un servicio en la Guía del usuario de IAM. Si elimina este rol vinculado al servicio, puede utilizar este mismo proceso para volver a crear el rol.

Edición de un rol vinculado a un servicio para AWS IoT TwinMaker

AWS IoT TwinMaker no permite editar el rol vinculado al servicio de AWSService RoleForIo TTwin Maker. Después de crear un rol vinculado al servicio, no podrá cambiar el nombre del rol, ya que varias entidades podrían hacer referencia al rol. Sin embargo, sí puede editar la descripción del rol con IAM. Para obtener más información, consulte Editar un rol vinculado a servicios en la Guía del usuario de IAM.

Eliminar un rol vinculado a un servicio para AWS IoT TwinMaker

Si ya no necesita usar una característica o servicio que requieran un rol vinculado a un servicio, le recomendamos que elimine dicho rol. Así no tendrá una entidad no utilizada que no se supervise ni mantenga de forma activa. Sin embargo, debe limpiar todos los espacios de trabajo vinculados a servicios que sigan utilizando su rol vinculado al servicio antes de poder eliminarlo manualmente.

> **Note**
>
> Si el AWS IoT TwinMaker servicio utiliza el rol al intentar eliminar los recursos, es posible que la eliminación no se realice correctamente. En tal caso, espere unos minutos e intente de nuevo la operación.

Para eliminar manualmente el rol vinculado a servicios mediante IAM

Utilice la consola de IAM AWS CLI, la o la AWS API para eliminar la función vinculada al servicio AWSService RoleForIo TTwin Maker. Para obtener más información, consulte Eliminación de un rol vinculado a servicios en la Guía del usuario de IAM.

Regiones compatibles para los roles vinculados al servicio AWS IoT TwinMaker

AWS IoT TwinMaker admite el uso de funciones vinculadas al servicio en todas las regiones en las que el servicio está disponible. Para obtener más información, consulte Puntos de conexión y Regiones de AWS.

AWS políticas gestionadas para AWS IoT TwinMaker

Para añadir permisos a usuarios, grupos y roles, es más fácil usar políticas AWS administradas que escribirlas tú mismo. Se necesita tiempo y experiencia para crear políticas administradas por el cliente de IAM que proporcionen a su equipo solo los permisos necesarios. Para empezar rápidamente, puedes usar nuestras políticas AWS gestionadas. Estas políticas cubren casos de uso comunes y están disponibles en su Cuenta de AWS. Para obtener más información sobre las políticas AWS administradas, consulte las políticas AWS administradas en la Guía del usuario de IAM.

AWS los servicios mantienen y AWS actualizan las políticas gestionadas. No puede cambiar los permisos en las políticas AWS gestionadas. En ocasiones, los servicios agregan permisos adicionales a una política administrada de AWS para admitir características nuevas. Este tipo de actualización afecta a todas las identidades (usuarios, grupos y roles) donde se asocia la política. Es más probable que los servicios actualicen una política gestionada por AWS cuando se lanza una nueva característica o cuando se ponen a disposición nuevas operaciones. Los servicios no eliminan los permisos de una política AWS administrada, por lo que las actualizaciones de la política no afectarán a los permisos existentes.

Además, AWS admite políticas administradas para funciones laborales que abarcan varios servicios. Por ejemplo, la política ReadOnlyAccess AWS gestionada proporciona acceso de solo lectura a todos los AWS servicios y recursos. Cuando un servicio lanza una nueva función, AWS agrega permisos de solo lectura para nuevas operaciones y recursos. Para obtener una lista y descripciones de las políticas de funciones de trabajo, consulte Políticas administradas de AWS para funciones de trabajo en la Guía del usuario de IAM.

AWS política gestionada: AWSIo TTwin MakerServiceRolePolicy

No puede adjuntarse AWSIo TTwin MakerServiceRolePolicy a sus entidades de IAM. Esta política está adjunta a un rol vinculado a servicios que permite a realizar acciones en su nombre. Para obtener más información, consulte Permisos de roles vinculados al servicio para AWS IoT TwinMaker.

La política de permisos de roles denominada AWSIo TTwin MakerServiceRolePolicy AWS IoT TwinMaker permite realizar las siguientes acciones en los recursos especificados:

- Acción: `iotsitewise:DescribeAsset, iotsitewise:ListAssets, iotsitewise:DescribeAssetModel, and iotsitewise:ListAssetModels, iottwinmaker:GetEntity, iottwinmaker:CreateEntity, iottwinmaker:UpdateEntity, iottwinmaker:DeleteEntity, iottwinmaker:ListEntities, iottwinmaker:GetComponentType, iottwinmaker:CreateComponentType, iottwinmaker:UpdateComponentType, iottwinmaker:DeleteComponentType, iottwinmaker:ListComponentTypes` en `all your iotsitewise asset and asset-model resources`

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [{
            "Sid": "SiteWiseAssetReadAccess",
            "Effect": "Allow",
            "Action": [
                "iotsitewise:DescribeAsset"
            ],
            "Resource": [
                "arn:aws:iotsitewise:*:*:asset/*"
            ]
        },
        {
            "Sid": "SiteWiseAssetModelReadAccess",
            "Effect": "Allow",
            "Action": [
                "iotsitewise:DescribeAssetModel"
            ],
            "Resource": [
                "arn:aws:iotsitewise:*:*:asset-model/*"
            ]
        },
        {
```

```
            "Sid": "SiteWiseAssetModelAndAssetListAccess",
            "Effect": "Allow",
            "Action": [
                "iotsitewise:ListAssets",
                "iotsitewise:ListAssetModels"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Sid": "TwinMakerAccess",
            "Effect": "Allow",
            "Action": [
                "iottwinmaker:GetEntity",
                "iottwinmaker:CreateEntity",
                "iottwinmaker:UpdateEntity",
                "iottwinmaker:DeleteEntity",
                "iottwinmaker:ListEntities",
                "iottwinmaker:GetComponentType",
                "iottwinmaker:CreateComponentType",
                "iottwinmaker:UpdateComponentType",
                "iottwinmaker:DeleteComponentType",
                "iottwinmaker:ListComponentTypes"
            ],
            "Resource": [
                "arn:aws:iottwinmaker:*:*:workspace/*"
            ],
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "iottwinmaker:linkedServices": [
                        "IOTSITEWISE"
                    ]
                }
            }
        }
    ]
}
```

AWS IoT TwinMaker actualizaciones de las políticas AWS administradas

Consulte los detalles sobre las actualizaciones de las políticas AWS administradas desde que este servicio comenzó a realizar el seguimiento de estos cambios. Para obtener alertas automáticas sobre los cambios realizados en esta página, suscríbase a la fuente RSS en la Página del historial de revisión de .

Cambio	Descripción	Fecha
AWSIoTTwinMakerServiceRolePolicy— Se agregó una política	AWS IoT TwinMaker se agregó la política de permisos de roles denominada AWSIoTTwinMakerServiceRolePolicy , que AWS IoT TwinMaker permite realizar las siguientes acciones en los recursos especificados: • Acción: `iotsitewise:DescribeAsset, iotsitewise:ListAssets, iotsitewise:DescribeAssetModel, and iotsitewise:ListAssetModels, iottwinmaker:GetEntity, iottwinmaker:CreateEntity, iottwinmaker:UpdateEntity, iottwinmaker:DeleteEntity, iottwinmaker:ListEntities, iottwinmaker:GetComponentType, iottwinmaker:CreateComponentType, iottwinmaker:Updat`	17 de noviembre de 2023

Cambio	Descripción	Fecha
	`eComponentType, iottwinmaker:Delet eComponentType, iottwinmaker:ListC omponentTypes en all your iotsitewise asset and asset-mod el resources` Para obtener más informaci ón, consulte Permisos de roles vinculados al servicio para AWS IoT TwinMaker.	
comenzó a rastrear los cambios	comenzó a realizar un seguimiento de los cambios de sus políticas AWS gestionadas.	11 de mayo de 2022

AWS IoT TwinMaker y puntos finales de VPC de interfaz ()AWS PrivateLink

Puede establecer una conexión privada entre su nube privada virtual (VPC) y la AWS IoT TwinMaker creando un punto de conexión de VPC de interfaz. Los puntos finales de la interfaz están alimentados por AWS PrivateLink, lo que puede utilizar para acceder de forma privada AWS IoT TwinMaker APIs sin una puerta de enlace a Internet, un dispositivo de traducción de direcciones de red (NAT), una conexión VPN o una conexión AWS Direct Connect. AWS IoT TwinMaker admite ambos tipos IPv4 y IPv6 (doble pila) a través de sus puntos finales de interfaz. Las instancias de su VPC no necesitan direcciones IP públicas para comunicarse con ellas. AWS IoT TwinMaker APIs El tráfico entre tu VPC y AWS IoT TwinMaker no sale de la red de Amazon.

Cada punto de conexión de la interfaz está representado por una o más interfaces de red elásticas en las subredes.

Para obtener más información, consulte Puntos de conexión de VPC de interfaz (AWS PrivateLink) en la Guía del usuario de Amazon VPC.

Consideraciones sobre los puntos AWS IoT TwinMaker finales de VPC

Antes de configurar un punto de enlace de VPC de interfaz AWS IoT TwinMaker, consulte las propiedades y limitaciones del punto de enlace de interfaz en la Guía del usuario de Amazon VPC.

AWS IoT TwinMaker admite realizar llamadas a todas sus acciones de API desde su VPC.

- Para las operaciones de la API del plano de datos, utilice el siguiente punto de conexión:

  ```
  data.iottwinmaker.region.amazonaws.com
  ```

  Las operaciones de la API del plano de datos incluyen lo siguiente:

  - GetPropertyValue
  - GetPropertyValueHistory
  - BatchPutPropertyValues
- Para las operaciones de la API del plano de control, utilice el siguiente punto de conexión:

  ```
  api.iottwinmaker.region.amazonaws.com
  ```

  Las operaciones de la API del plano de control admitidas incluyen lo siguiente:

  - CreateComponentType
  - CreateEntity
  - CreateScene
  - CreateWorkspace
  - DeleteComponentType
  - DeleteEntity
  - DeleteScene
  - DeleteWorkspace
  - GetComponentType
  - GetEntity
  - GetScene
  - GetWorkspace

- ListComponentTypes
- ListComponentTypes
- ListEntities
- ListScenes
- ListTagsForResource
- ListWorkspaces
- TagResource
- UntagResource
- UpdateComponentType
- UpdateEntity
- UpdateScene
- UpdateWorkspace

Creación de un punto de conexión de VPC de interfaz para AWS IoT TwinMaker

Puede crear un punto de enlace de VPC para el AWS IoT TwinMaker servicio mediante la consola de Amazon VPC o el (). AWS Command Line Interface AWS CLI Para obtener más información, consulte Creación de un punto de conexión de interfaz en la Guía del usuario de Amazon VPC.

Cree un punto final de VPC para el AWS IoT TwinMaker que se utilice el siguiente nombre de servicio.

- Para las operaciones de la API del plano de datos, utilice el siguiente nombre de servicio:

```
com.amazonaws.region.iottwinmaker.data
```

- Para las operaciones de la API del plano de control, utilice el siguiente nombre de servicio:

```
com.amazonaws.region.iottwinmaker.api
```

Si habilitas el DNS privado para el punto final, puedes realizar solicitudes a la API AWS IoT TwinMaker utilizando su nombre de DNS predeterminado para la región, por ejemplo. `iottwinmaker.us-east-1.amazonaws.com`

Para más información, consulte Acceso a un servicio a través de un punto de conexión de interfaz en la Guía del usuario de Amazon VPC.

AWS IoT TwinMaker PrivateLink se admite en las siguientes regiones:

- us-east-1

  El ControlPlane servicio se admite en las siguientes zonas de disponibilidad: `use1-az1use1-az2`, `yuse1-az6`.

  El DataPlane servicio se admite en las siguientes zonas de disponibilidad: `use1-az1use1-az2`, `yuse1-az4`.
- us-west-2

  Los DataPlane servicios ControlPlane y se admiten en las siguientes zonas de disponibilidad: `usw2-az1usw2-az2`, `yusw2-az3`.
- eu-west-1
- eu-central-1
- ap-southeast-1
- ap-southeast-2

Para obtener más información sobre las zonas de disponibilidad, consulte Zona de disponibilidad IDs para sus AWS recursos: AWS Resource Access Manager.

Acceso a AWS IoT TwinMaker través de un punto final de VPC de interfaz

Al crear un punto final de interfaz, AWS IoT TwinMaker genera nombres de host DNS específicos del punto final con los que puede comunicarse. AWS IoT TwinMaker La opción de DNS privado está habilitada de forma predeterminada. Para obtener más información, consulte Uso de zonas alojadas privadas en la Guía del usuario de Amazon VPC.

Si habilita el DNS privado para el punto de conexión, podrá realizar solicitudes de API a AWS IoT TwinMaker a través de uno de los siguientes puntos de conexión de VPC.

- Para las operaciones de la API del plano de datos, utilice el siguiente punto de conexión: Sustituya *`region`* por su región de AWS .

```
data.iottwinmaker.region.amazonaws.com
```

- Para las operaciones de la API del plano de control, utilice el siguiente punto de conexión: Sustituya *region* por su región de AWS .

```
api.iottwinmaker.region.amazonaws.com
```

Si inhabilita el DNS privado para el punto de conexión, debe hacer lo siguiente para acceder a AWS IoT TwinMaker través del punto de conexión:

- Especifique la URL del punto de conexión de VPC en las solicitudes de API.
  - Para las operaciones de la API del plano de datos, utilice la siguiente URL de punto de conexión: Sustituya *vpc-endpoint-id* y *region* por su ID de punto final de VPC y su región.

```
vpc-endpoint-id.data.iottwinmaker.region.vpce.amazonaws.com
```

  - Para las operaciones de la API del plano de control, utilice la siguiente URL de punto de conexión: Sustituya *vpc-endpoint-id* y *region* por su ID de punto final de VPC y su región.

```
vpc-endpoint-id.api.iottwinmaker.region.vpce.amazonaws.com
```

- Inhabilite la inyección de prefijos de host. Al llamar a AWS SDKs cada operación de API, coloque varios prefijos de host antes del punto final del servicio. AWS CLI Esto provoca que el AWS CLI y AWS SDKs no sea válido URLs para AWS IoT TwinMaker cuando se especifica un punto final de VPC.

> **Important**
>
> No puede deshabilitar la inyección de prefijos de host en AWS CLI o Herramientas de AWS para PowerShell. Esto significa que si has inhabilitado el DNS privado, no podrás usar el punto final de la VPC AWS CLI ni acceder Herramientas de AWS para PowerShell a AWS IoT TwinMaker él a través de él. Si quieres usar estas herramientas para acceder a AWS IoT TwinMaker través del punto final, habilita el DNS privado.

Para obtener más información sobre cómo deshabilitar la inyección de prefijos de host en el AWS SDKs, consulta las siguientes secciones de la documentación de cada SDK:

- AWS SDK para C++
- AWS SDK para Go

- AWS SDK para Go v2
- AWS SDK para Java
- AWS SDK for Java 2.x
- AWS SDK para JavaScript
- AWS SDK para .NET
- AWS SDK para PHP
- AWS SDK para Python (Boto3)
- AWS SDK para Ruby

Para más información, consulte Acceso a un servicio a través de un punto de conexión de interfaz en la Guía del usuario de Amazon VPC.

Crear una política de puntos de conexión de VPC para AWS IoT TwinMaker

Puede asociar una política de punto de conexión con su punto de conexión de VPC que controla el acceso a AWS IoT TwinMaker. La política especifica la siguiente información:

- La entidad principal que puede realizar acciones.
- Las acciones que se pueden realizar.
- Los recursos en los que se pueden llevar a cabo las acciones.

Para más información, consulte Control del acceso a los servicios con puntos de enlace de la VPC en la Guía del usuario de Amazon VPC.

Ejemplo: política de puntos finales de VPC para acciones AWS IoT TwinMaker

El siguiente es un ejemplo de una política de puntos finales para AWS IoT TwinMaker. Cuando se adjunta a un punto final, esta política otorga acceso a las AWS IoT TwinMaker acciones enumeradas al usuario de IAM de la AWS cuenta `iottwinmakeradmin` `123456789012` en todos los recursos.

```
{
   "Statement":[
      {
        "Principal": {
            "AWS": "arn:aws:iam::123456789012:user/role"
                },
```

```
            "Resource": "*",
            "Effect":"Allow",
            "Action":[
               "iottwinmaker:CreateEntity",
               "iottwinmaker:GetScene",
               "iottwinmaker:ListEntities"
            ]
           }
    ]
}
```

Validación de conformidad para AWS IoT TwinMaker

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:

- Cumplimiento de seguridad y gobernanza: en estas guías se explican las consideraciones de arquitectura y se proporcionan pasos para implementar las características de seguridad y cumplimiento.
- Referencia de servicios válidos de HIPAA: muestra una lista con los servicios válidos de HIPAA. No todos Servicios de AWS cumplen con los requisitos de la 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 el punto de vista 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 TwinMaker

La infraestructura AWS global se basa en zonas Regiones de AWS de disponibilidad. Regiones de AWS proporcionan varias zonas de disponibilidad aisladas y separadas físicamente, que están conectadas mediante redes de baja latencia, alto rendimiento y alta redundancia. Con las zonas de disponibilidad, puede diseñar y utilizar aplicaciones y bases de datos que realizan una conmutación por error automática entre las zonas sin interrupciones. Las zonas de disponibilidad tienen una mayor disponibilidad, tolerancia a errores y escalabilidad que las infraestructuras tradicionales de uno o varios centros de datos.

Para obtener más información sobre las zonas de disponibilidad Regiones de AWS y las zonas de disponibilidad, consulte Infraestructura global.AWS

Además de la infraestructura AWS global, AWS IoT TwinMaker ofrece varias funciones para ayudarlo a satisfacer sus necesidades de respaldo y resiliencia de datos.

Seguridad de la infraestructura en AWS IoT TwinMaker

Como servicio gestionado, AWS IoT TwinMaker está protegido por los procedimientos de seguridad de red AWS global que se describen en el documento técnico Amazon Web Services: Overview of Security Processes.

Utiliza las llamadas a la API AWS publicadas para acceder a AWS IoT TwinMaker través de la red. Los clientes deben ser compatibles con Transport Layer Security (TLS) 1.2 o una versión posterior. Recomendamos TLS 1.3 o una versión posterior. Los clientes también deben ser compatibles con conjuntos de cifrado con confidencialidad directa total (PFS) tales como Ephemeral Diffie-Hellman (DHE) o Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). 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 principal de IAM. También puedes utilizar AWS Security Token Service (AWS STS) para generar credenciales de seguridad temporales para firmar solicitudes.

Puntos de conexión y cuotas

AWS IoT TwinMaker puntos finales y cuotas

Puede encontrar información sobre los AWS IoT TwinMaker puntos finales y las cuotas en la Referencia AWS general.

- Para obtener información sobre los puntos de conexión de servicios, consulte Puntos de conexión de servicios de AWS IoT TwinMaker.
- Para obtener información sobre las cuotas, consulte Service Quotas de AWS IoT TwinMaker.
- Para obtener información sobre los límites de limitación de las API, consulte los límites de limitación de las API de AWS IoT TwinMaker.

Información adicional sobre AWS IoT TwinMaker los puntos finales

Para conectarse mediante programación AWS IoT TwinMaker, utilice un punto final. Si utiliza un cliente HTTP, debe prefijar el plano de control y el plano de datos de la siguiente manera. APIs Sin embargo, no es necesario añadir un prefijo al AWS SDK ni a los AWS Command Line Interface comandos, ya que añaden automáticamente el prefijo necesario.

- Use el `api` prefijo para el plano de control. APIs Por ejemplo, `api.iottwinmaker.us-west-1.amazonaws.com`.
- Utilice el `data` prefijo para el plano de datos. APIs Por ejemplo, `data.iottwinmaker.us-west-1.amazonaws.com`.

Historial de documentos para la guía del usuario de AWS IoT TwinMaker

En la siguiente tabla se describen las versiones de la documentación de AWS IoT TwinMaker.

Cambio	Descripción	Fecha
Nuevo rol vinculado a servicios y nueva política de IAM	AWS IoT TwinMakeragregó un nuevo rol vinculado a un servicio, llamado. AWSServiceRoleForIoTTwinMaker AWS IoT TwinMakeragregó este nuevo rol vinculado a un servicio para poder llamar AWS IoT TwinMaker a otros AWS servicios y sincronizar sus recursos en tu nombre. La nueva política de AWSIoTTwinMakerServiceRolePolicy IAM está asociada a esta función y otorga permiso para llamar AWS IoT TwinMaker a otros AWS servicios y sincronizar sus recursos en tu nombre.	17 de noviembre de 2023
Versión inicial	Versión inicial de la Guía del usuario de AWS IoT TwinMaker	30 de noviembre de 2021

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.