



Guia do usuário

# AWS IoT TwinMaker



# AWS IoT TwinMaker: Guia do usuário

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

As marcas comerciais e imagens comerciais da Amazon não podem ser usadas no contexto de nenhum produto ou serviço que não seja da Amazon, nem de qualquer maneira que possa gerar confusão entre os clientes ou que deprecie ou desprestige a Amazon. Todas as outras marcas comerciais que não pertencem à Amazon pertencem a seus respectivos proprietários, que podem ou não ser afiliados, patrocinados pela Amazon ou ter conexão com ela.

---

# Table of Contents

O que é AWS IoT TwinMaker? .....	1
Como funciona .....	1
Componentes e conceitos-chave .....	2
Espaço de trabalho .....	3
Modelo de entidade-componente .....	4
Visualização .....	6
Começando com AWS IoT TwinMaker .....	8
Crie e gerencie um perfil de serviço para o AWS IoT TwinMaker .....	9
Atribuir confiança .....	9
Permissões do Amazon S3 .....	9
Atribua permissões a um bucket específico do Amazon S3 .....	10
Permissões para conectores integrados .....	12
Permissões para um conector a uma fonte de dados externa .....	15
Modifique sua IAM função de espaço de trabalho para usar o conector de dados Athena .....	16
Para criar um workspace .....	18
Crie sua primeira entidade .....	20
Configurando uma AWS conta .....	24
Inscreva-se para um Conta da AWS .....	24
Criar um usuário com acesso administrativo .....	25
Uso e criação de tipos de componentes .....	27
Tipos de componentes integrados .....	27
Principais características dos tipos de AWS IoT TwinMaker componentes .....	28
Criação de definições de propriedade .....	30
Criar funções .....	31
Exemplo de tipos de componentes .....	31
Alarme (abstrato) .....	32
Telemetria do fluxo de tempo .....	33
Alarme (herda do alarme abstrato) .....	34
Exemplos de equipamentos .....	35
Operações em massa .....	38
Principais conceitos e terminologia .....	38
AWS IoT TwinMaker metadataTransferJob funcionalidade .....	39
Executando operações de importação e exportação em massa .....	40
metadataTransferJob pré-requisitos .....	41

permissões do IAM .....	41
Execute uma operação em massa .....	45
Gerenciamento de erros .....	48
Importar modelos de metadados .....	49
AWS IoT TwinMaker metadataTransferJob exemplos .....	53
AWS IoT TwinMaker esquema de trabalho de transferência de metadados .....	54
Conectores de dados .....	72
Conectores de dados .....	72
Conector inicializador de esquema .....	73
DataReaderByEntity .....	74
DataReaderByComponentType .....	75
DataReader .....	77
AttributePropertyValueReaderByEntity .....	78
DataWriter .....	79
Exemplos .....	80
Conector de dados tabulares do Athena .....	89
AWS IoT TwinMaker Pré-requisitos do conector de dados Athena .....	89
Como usar o conector de dados do Athena .....	90
Como usar a referência JSON do conector de dados tabulares do Athena .....	94
Como usar o conector de dados do Athena .....	95
Visualize dados tabulares do Athena no Grafana .....	96
AWS IoT TwinMaker conector de dados de série temporal .....	98
AWS IoT TwinMaker pré-requisitos do conector de dados de série temporal .....	98
Plano de fundo do conector de dados de séries temporais .....	99
Como desenvolver um conector de dados de séries temporais .....	100
Como melhorar seu conector de dados .....	109
Como testar seu conector .....	110
Segurança .....	110
Criação de AWS IoT TwinMaker recursos .....	110
Próximas etapas .....	112
AWS IoT TwinMaker conector de dados de fábrica de cookies .....	112
Criação de AWS IoT TwinMaker cenas .....	118
Antes de criar cenas .....	118
Otimize seus recursos antes de importá-los AWS IoT TwinMaker .....	118
Melhores práticas para desempenho em AWS IoT TwinMaker .....	119
Saiba mais .....	119

Fazendo upload de recursos em AWS IoT TwinMaker .....	120
Carregar arquivos para a Biblioteca de Recursos usando o console .....	120
Crie suas cenas .....	120
Use a navegação 3D AWS IoT TwinMaker em suas cenas .....	122
Adicione câmeras fixas .....	124
Edição avançada .....	124
Posicionamento direcionado dos objetos da cena .....	125
Seleção de submodelos .....	125
Editar entidades na hierarquia da cena .....	126
Adicione anotações às entidades .....	126
Adicione sobreposições às tags .....	131
Edite suas cenas .....	139
Adicione modelos .....	139
Adicione widgets .....	140
Adicionar tags .....	144
Otimize seu modelo 3D .....	144
Usando blocos 3D em sua cena .....	144
Cenas dinâmicas .....	147
Cenas estáticas versus dinâmicas .....	147
Tipos e entidades de componentes de cena .....	148
Conceitos de cena dinâmica .....	149
AWS IoT TwinMaker integração do kit de aplicativos .....	150
Alterne AWS IoT TwinMaker os modos de preços .....	151
Gráfico de conhecimento .....	153
AWS IoT TwinMaker conceitos básicos do gráfico de conhecimento .....	153
Usar o gráfico de conhecimento .....	154
Gere um gráfico de cena .....	157
AWS IoT TwinMaker pré-requisitos do gráfico de cena .....	158
Vincule nós 3D em sua cena .....	158
Crie um aplicativo web .....	161
Painel do Grafana do gráfico de conhecimento .....	163
AWS IoT TwinMaker pré-requisitos do editor de consultas .....	163
Gráfico de conhecimento e permissões do Grafana .....	163
Recursos adicionais do gráfico de conhecimento .....	168
Sincronização de ativos com o AWS IoT SiteWise .....	181
Usar a sincronização de ativos com o AWS IoT SiteWise .....	181

Usando um espaço de trabalho personalizado .....	182
Usando o IoT Site WiseDefaultWorkspace .....	187
Diferenças entre espaços de trabalho personalizados e padrão .....	188
Recursos sincronizados de AWS IoT SiteWise .....	189
Espaços de trabalho personalizados e padrão .....	189
Somente espaço de trabalho padrão .....	190
Recursos não sincronizados .....	191
Use entidades sincronizadas e tipos de componentes em AWS IoT TwinMaker .....	191
Análise o status e os erros de sincronização .....	192
Sincronizar status de tarefa .....	192
Excluir uma tarefa de sincronização .....	194
Limites de sincronização de ativos .....	196
Como configurar painéis do Grafana .....	198
Configuração de CORS .....	199
Como configurar seu ambiente do Grafana .....	200
Amazon Managed Grafana .....	200
Grafana autogerenciado .....	201
Como criar um perfil de painel .....	202
Criar uma política do IAM .....	202
Faça upload de vídeo do Edge .....	206
Adicione mais permissões .....	206
Como criar o perfil do IAM do painel do Grafana .....	208
Criação de uma política AWS IoT TwinMaker de reprodutor de vídeo .....	209
Diminua o acesso aos seus recursos .....	210
Diminua o escopo das permissões GET .....	210
AWS IoT SiteWise BatchPutAssetPropertyValue Permissão para reduzir o escopo .....	212
Conecte alarmes aos painéis do Grafana .....	214
AWS IoT SiteWise pré-requisitos de configuração de alarme .....	214
Defina a função do IAM do componente de AWS IoT SiteWise alarme .....	215
Consulte e atualize por meio da AWS IoT TwinMaker API .....	216
Configure seu painel Grafana para alarmes .....	218
Use o painel Grafana para a visualização de alarmes .....	220
Integração com o Matterport .....	222
Visão geral da integração .....	223
Pré-requisitos da integração com o Matterport .....	224
Credenciais do SDK do Matterport .....	226

Armazene as credenciais do Matterport em AWS Secrets Manager .....	227
O Matterport digitaliza em cenas AWS IoT TwinMaker .....	230
Matterport em seu painel do Grafana AWS IoT TwinMaker .....	236
Integração do Matterport com o AWS IoT kit de aplicativos .....	236
Streaming de vídeo para AWS IoT TwinMaker .....	237
Use o conector edge para o stream de vídeo do Kinesis para transmitir vídeo em AWS IoT TwinMaker .....	237
Pré-requisitos .....	237
Crie componentes de vídeo para AWS IoT TwinMaker cenas .....	238
Adicione vídeo e metadados da transmissão de vídeo do Kinesis a um painel do Grafana .....	239
Como usar a biblioteca Flink AWS IoT TwinMaker .....	241
Registro e monitoramento .....	242
Monitorar o com métricas do Amazon CloudWatch .....	242
Métricas .....	243
Registrar em log chamadas de API com o AWS CloudTrail .....	245
Informações do AWS IoT TwinMaker no CloudTrail .....	246
Segurança .....	248
Proteção de dados .....	249
Criptografia em repouso .....	250
Criptografia em trânsito .....	250
Gerenciamento de Identidade e Acesso .....	250
Público .....	251
Autenticação com identidades .....	251
Gerenciar o acesso usando políticas .....	255
Como AWS IoT TwinMaker funciona com o IAM .....	258
Exemplos de políticas baseadas em identidade .....	265
Solução de problemas .....	268
Usar perfis vinculados a serviço .....	270
AWS políticas gerenciadas .....	272
Endpoints da VPC (AWS PrivateLink) .....	277
Considerações sobre AWS IoT TwinMaker VPC endpoints .....	277
Criar um endpoint da VPC de interface para o AWS IoT TwinMaker .....	279
Acessando AWS IoT TwinMaker por meio de uma interface VPC endpoint .....	280
Criação de uma política de VPC endpoint para AWS IoT TwinMaker .....	282
Validação de conformidade .....	283
Resiliência .....	284

---

Segurança da infraestrutura .....	284
Endpoints e cotas .....	286
AWS IoT TwinMaker endpoints e cotas .....	286
Informações adicionais sobre endpoints .....	286
Histórico do documento .....	287
.....	cclxxxviii

# O que é AWS IoT TwinMaker?

AWS IoT TwinMaker é um AWS IoT serviço que você pode usar para criar gêmeos digitais operacionais de sistemas físicos e digitais. AWS IoT TwinMaker cria visualizações digitais usando medições e análises de uma variedade de sensores, câmeras e aplicativos corporativos do mundo real para ajudá-lo a acompanhar sua fábrica física, prédio ou planta industrial. Você pode usar esses dados do mundo real para monitorar operações, diagnosticar e corrigir erros, e otimizar as operações.

Um gêmeo digital é uma representação digital ao vivo de um sistema e de todos os seus componentes físicos e digitais. É atualizado dinamicamente com dados para imitar a verdadeira estrutura, estado e comportamento do sistema. Você pode usá-lo para gerar resultados comerciais.

Os usuários finais interagem com os dados do seu gêmeo digital usando um aplicativo de interface de usuário.

## Como funciona

Para atender aos requisitos mínimos para criar um gêmeo digital, é necessário fazer o seguinte.

- Modele dispositivos, equipamentos, espaços e processos em um local físico.
- Conecte esses modelos a fontes de dados que armazenam informações contextuais importantes, como feeds de câmeras de dados de sensores.
- Crie visualizações que ajudem os usuários a entender os dados e insights para tomar decisões de negócios com mais eficiência.
- Disponibilize gêmeos digitais aos usuários finais para impulsionar os resultados comerciais.

AWS IoT TwinMaker aborda esses desafios fornecendo os seguintes recursos.

- Gráfico de conhecimento do sistema de componentes da entidade: AWS IoT TwinMaker fornece ferramentas para modelar dispositivos, equipamentos, espaços e processos em um gráfico de conhecimento.

Esse gráfico de conhecimento contém metadados sobre o sistema e pode se conectar a dados em diferentes locais. AWS IoT TwinMaker fornece conectores integrados para dados armazenados no Kinesis AWS IoT SiteWise Video Streams. Você também pode criar conectores personalizados para dados armazenados em outros locais.

Juntos, o gráfico de conhecimento e os conectores fornecem uma interface única para consultar dados em locais diferentes.

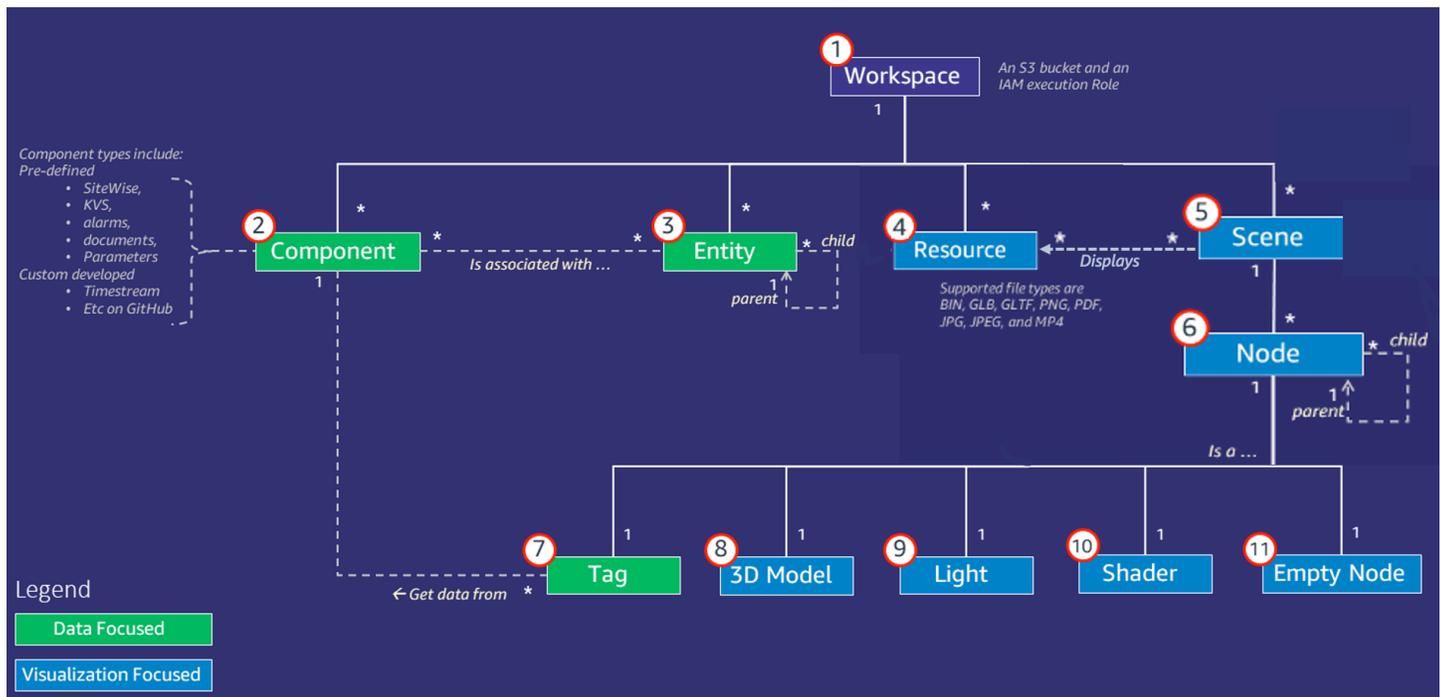
- Compositor de cenas: o AWS IoT TwinMaker console fornece uma ferramenta de composição de cenas para criar cenas em 3D. Você carrega seus modelos 3D/CAD criados anteriormente, otimizados para exibição na web e convertidos para o formato .glTF ou .glb. Em seguida, você usa o compositor de cenas para posicionar vários modelos em uma única cena, criando representações visuais de suas operações.

Você também pode sobrepor dados na cena. Por exemplo, você pode criar uma tag em um local de cena que se conecta aos dados de temperatura de um sensor. Isso associa os dados à localização.

- Aplicativos: AWS IoT TwinMaker fornece um plug-in para o Grafana e o Amazon Managed Grafana que você pode usar para criar aplicativos de painel para usuários finais.
- Ferramentas de terceiros: a Mendix faz parceria AWS IoT TwinMaker para fornecer soluções completas para IoT industrial. Veja o workshop [Lean Daily Management Application with Mendix e comece AWS IoT TwinMaker a usar a Mendix Low Code Application Development Platform \(LCAP\)](#) com AWS serviços como Kinesis AWS IoT TwinMaker Video Streams e. AWS IoT SiteWise

## Componentes e conceitos-chave

O diagrama a seguir ilustra como os principais conceitos de AWS IoT TwinMaker se encaixam.



### Note

Os asteriscos (\*) no diagrama indicam one-to-many relacionamentos. Para as cotas de cada um desses relacionamentos, consulte [endpoints e cotas do AWS IoT TwinMaker](#).

As seções a seguir descrevem os conceitos ilustrados no diagrama.

## Espaço de trabalho

Um espaço de trabalho é um contêiner de alto nível para seu aplicativo de gêmeos digitais. Você cria um conjunto lógico de entidades, componentes, ativos de cena e outros recursos para seu gêmeo digital dentro desse espaço de trabalho. Ele também serve como um limite de segurança para gerenciar o acesso ao aplicativo do gêmeo digital e aos recursos que ele contém. Cada espaço de trabalho é vinculado ao bucket do Amazon S3 onde os dados do seu espaço de trabalho são armazenados. Você usa perfis do IAM para restringir o acesso ao seu espaço de trabalho.

Um espaço de trabalho pode conter vários componentes, entidades, cenas e recursos. Um tipo de componente, entidade, cena ou recurso existe somente em um espaço de trabalho.

## Modelo de entidade-componente

AWS IoT TwinMaker fornece ferramentas que você usa para modelar seu sistema usando um gráfico de entity-component-based conhecimento. Você pode usar a arquitetura de entidade-componente para criar uma representação do seu sistema físico. Esse modelo de componente de entidade consiste em entidades, componentes e relacionamentos. Para obter mais informações sobre sistemas de entidade-componente, consulte [Sistema de componentes de entidade](#).

### Entidade

Entidades são representações digitais dos elementos em um gêmeo digital que capturam as capacidades desse elemento. Tal elemento pode ser um equipamento físico, conceito ou processo. As entidades têm componentes associados a elas. Esses componentes fornecem dados e contexto para a entidade associada.

Com AWS IoT TwinMaker, você pode organizar entidades em hierarquias personalizadas para um gerenciamento mais eficiente. A visualização padrão do sistema de entidades e componentes é hierárquica.

### Componente

Os componentes fornecem contexto e dados para entidades em uma cena. Você adiciona componentes a entidades. A vida útil de um componente está vinculada à vida útil de uma entidade.

Os componentes podem adicionar dados estáticos, como uma lista de documentos ou as coordenadas de uma localização geográfica. Eles também podem ter funções que se conectam a outros sistemas, incluindo sistemas que contêm dados de séries temporais, como AWS IoT SiteWise e outros historiadores de nuvens de séries temporais.

Os componentes são definidos por documentos JSON que descrevem a conexão entre uma fonte de dados e o AWS IoT TwinMaker. Os componentes podem descrever fontes de dados externas ou fontes de dados incorporadas AWS IoT TwinMaker. Um componente acessa uma fonte de dados externa usando uma função do Lambda especificada no documento JSON. Um espaço de trabalho pode conter muitos componentes. Os componentes fornecem dados às tags por meio de entidades associadas.

AWS IoT TwinMaker fornece vários componentes integrados que você pode adicionar do console. Você também pode criar seus próprios componentes personalizados para se conectar a fontes de dados, como telemetria de fluxo de tempo e coordenadas geoespaciais. Exemplos disso incluem

TimeStream telemetria, componentes geoespaciais e conectores para fontes de dados de terceiros, como o Snowflake.

AWS IoT TwinMaker fornece os seguintes tipos de componentes integrados para casos de uso comuns:

- Documento, como manuais do usuário ou imagens localizadas em URLs especificados.
- Séries temporais, como dados de sensor do AWS IoT SiteWise.
- Alarmes, como alarmes de séries temporais de fontes de dados externas.
- Vídeo de câmeras IP conectadas ao Kinesis Video Streams.
- Componentes personalizados para se conectar a fontes de dados adicionais. Por exemplo, você pode criar um conector personalizado para conectar suas entidades AWS IoT TwinMaker a dados de séries temporais armazenados externamente.

## Fontes de dados

Uma fonte de dados é a localização dos dados de origem do seu gêmeo digital. AWS IoT TwinMaker suporta dois tipos de fontes de dados:

- Conectores hierárquicos, que permitem sincronizar continuamente um modelo externo com o AWS IoT TwinMaker.
- Conectores de séries temporais, que permitem que você se conecte a bancos de dados de séries temporais, como o AWS IoT SiteWise.

## Propriedade

As propriedades são os valores, tanto estáticos quanto baseados em séries temporais, contidos nos componentes. Quando você adiciona componentes às entidades, as propriedades no componente descrevem detalhes sobre o estado atual da entidade.

AWS IoT TwinMaker suporta três tipos de propriedades:

- Valor único, non-time-series propriedades — essas propriedades geralmente são pares estáticos de valores-chave e são armazenadas diretamente AWS IoT TwinMaker com os metadados da entidade associada.
- Propriedades de séries temporais — AWS IoT TwinMaker armazena uma referência ao armazenamento de séries temporais dessas propriedades. O padrão é o valor mais recente.

- Propriedades de relacionamento – essas propriedades armazenam uma referência a outra entidade ou componente. Por exemplo, `seen_by` é um componente de relacionamento que pode relacionar uma entidade de câmera a outra entidade que é visualizada diretamente por essa câmera.

Você pode consultar valores de propriedades em fontes de dados heterogêneas usando a interface unificada de consulta de dados.

## Visualização

Você usa AWS IoT TwinMaker para aumentar uma representação tridimensional do seu gêmeo digital e depois visualizá-lo na Grafana. Para criar cenas, use arquivos existentes do CAD ou outros tipos de arquivo 3D. Em seguida, usamos sobreposições de dados para adicionar dados relevantes para seu gêmeo digital.

### Cenas

As cenas são representações tridimensionais que fornecem contexto visual para os dados conectados. AWS IoT TwinMaker As cenas podem ser criadas usando um único modelo gltf (GL Transmission Format) ou glb 3D para todo o ambiente, ou usando uma composição de vários modelos. As cenas também incluem tags para indicar pontos de interesse na cena.

As cenas são os contêineres de nível superior para visualizações. Uma cena consiste em um ou mais nós.

Um espaço de trabalho pode conter várias cenas. Por exemplo, um espaço de trabalho pode conter uma cena para cada andar de uma instalação.

### Recursos

As cenas exibem recursos, que são exibidos como nós no AWS IoT TwinMaker console. Uma cena pode conter muitos recursos.

Os recursos são imagens e modelos tridimensionais do glTF baseados em imagens e usados para criar uma cena. Um recurso pode representar um único equipamento ou um local completo.

Você posiciona recursos em uma cena carregando um arquivo.gltf ou .glb na biblioteca de recursos do seu espaço de trabalho e adicionando-os à sua cena.

## Interface aumentada do usuário

Com isso, AWS IoT TwinMaker você pode aumentar suas cenas com sobreposições de dados que adicionam contexto e informações importantes, como dados de sensores, aos locais na cena.

**Nós:** os nós são instâncias de tags, luzes e modelos tridimensionais. Eles também podem estar vazios para adicionar estrutura à sua hierarquia de cenas. Por exemplo, você pode agrupar vários nós em um único nó vazio.

**Tags:** uma tag é um tipo de nó que representa dados de um componente (por meio de uma entidade). Uma tag pode ser associada apenas a um componente. Uma tag é uma anotação adicionada a uma posição específica da coordenada  $x, y, z$  de uma cena. A tag conecta essa parte da cena ao gráfico de conhecimento usando uma propriedade de entidade. Você pode usar uma tag para configurar o comportamento ou a aparência visual de um item na cena, como um alarme.

**Luzes:** você pode adicionar luzes a uma cena para focar ou projetar sombras em objetos específicos para indicar sua localização física.

**Modelos tridimensionais:** um modelo tridimensional é uma representação visual de um arquivo .gltf ou .gltb importado como um recurso.

### Note

AWS IoT TwinMaker não se destina ao uso em, ou em associação com, a operação de quaisquer ambientes perigosos ou sistemas críticos que possam causar lesões corporais graves ou morte ou causar danos ambientais ou materiais.

Os dados coletados por meio do seu uso de AWS IoT TwinMaker devem ser avaliados quanto à precisão, conforme apropriado para seu caso de uso. AWS IoT TwinMaker não deve ser usado como substituto do monitoramento humano de sistemas físicos para avaliar se esses sistemas estão operando com segurança.

# Começando com AWS IoT TwinMaker

Os tópicos nesta seção descrevem como fazer o seguinte.

- Crie e configure um novo espaço de trabalho.
- Crie uma entidade e adicione componentes para ela.

Pré-requisitos:

Para criar seu primeiro espaço de trabalho e cena, você precisa dos seguintes AWS recursos.

- Uma [conta da AWS](#).
- Uma função IAM de serviço para AWS IoT TwinMaker. Essa função é gerada automaticamente por padrão, quando você cria um novo AWS IoT TwinMaker espaço de trabalho no [AWS IoT TwinMaker console](#).

Se você optar por não permitir a criação AWS IoT TwinMaker automática de uma nova função de IAM serviço, deverá especificar uma que você já tenha criado.

Para obter instruções sobre como criar e gerenciar esse perfil de serviço, consulte [???](#).

Para obter mais informações sobre funções IAM de serviço, consulte [Criação de uma função para delegar permissões a um AWS service \(Serviço da AWS\)](#).

## Important

Essa função de serviço deve ter uma política anexada que conceda permissão para que o serviço leia e grave em um bucket do Amazon S3. AWS IoT TwinMaker usa essa função para acessar outros serviços em seu nome. Você também precisará atribuir uma relação de confiança entre essa função e AWS IoT TwinMaker para que o serviço possa assumir a função. Se seu irmão gêmeo interagir com outros AWS serviços, adicione também as permissões necessárias para esses serviços.

Tópicos

- [Crie e gerencie um perfil de serviço para o AWS IoT TwinMaker](#)
- [Para criar um workspace](#)

- [Crie sua primeira entidade](#)
- [Configurando uma AWS conta](#)

## Crie e gerencie um perfil de serviço para o AWS IoT TwinMaker

AWS IoT TwinMaker exige que você use um perfil de serviço para permitir o acesso a recursos em outros serviços em seu nome. Essa função deve ter uma relação de confiança com AWS IoT TwinMaker. Ao criar um espaço de trabalho, você deve atribuir esse perfil ao espaço de trabalho. Este tópico contém exemplos de políticas que mostram como configurar as permissões para cenários comuns.

### Atribuir confiança

A política a seguir estabelece uma relação de confiança entre sua função e AWS IoT TwinMaker. Atribua essa relação de confiança ao perfil que você usa no seu espaço de trabalho.

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

### Permissões do Amazon S3

A política a seguir permite que seu perfil leia, exclua e grave em um bucket do Amazon S3. Os espaços de trabalho armazenam recursos no Amazon S3, portanto, as permissões do Amazon S3 são necessárias para todos os espaços de trabalho.

```
{
  "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

Quando você cria um espaço de trabalho, AWS IoT TwinMaker cria um arquivo em seu bucket do Amazon S3 que indica que ele está sendo usado por um espaço de trabalho. Essa política dá AWS IoT TwinMaker permissão para excluir esse arquivo quando você exclui o espaço de trabalho.

AWS IoT TwinMaker coloca outros objetos relacionados ao seu espaço de trabalho. É sua responsabilidade excluir esses objetos quando excluir um espaço de trabalho.

## Atribua permissões a um bucket específico do Amazon S3

Ao criar um espaço de trabalho no AWS IoT TwinMaker console, você pode optar por AWS IoT TwinMaker criar um bucket do Amazon S3 para você. Você pode encontrar informações sobre esse bucket usando o AWS CLI comando a seguir.

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

O exemplo a seguir mostra o formato da saída deste 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 atualizar sua política para que ela atribua permissões para um bucket específico do Amazon S3, use o valor de *bucket name*.

A política a seguir permite que seu perfil leia, exclua e grave em um bucket específico do Amazon S3.

```
{
  "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_*"
    ]
  }
]
}

```

## Permissões para conectores integrados

Se o seu espaço de trabalho interagir com outros AWS serviços usando conectores integrados, você deverá incluir permissões para esses serviços nesta política. Se usar o tipo de componente `com.amazon.iotsitewise.connector`, você deverá incluir permissões para AWS IoT SiteWise. Para obter mais informações sobre tipos de componentes, consulte [???](#).

### Note

Se você interagir com outros AWS serviços usando um tipo de componente personalizado, deverá conceder permissão à função para executar a função Lambda que implementa a função em seu tipo de componente. Para obter mais informações, consulte [???](#).

O exemplo a seguir mostra como incluir AWS IoT SiteWise em sua política.

```

{
  "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_*"
  ]
}
]
}

```

Se você usa o tipo de componente com `amazon.iotsitewise.connector` e precisa ler os dados da propriedade, você deve incluir a seguinte permissão em sua política. AWS IoT SiteWise

```

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

```

```

    "Effect": "Allow"
  },
  ...

```

Se você usa o tipo de componente com `amazon.iotsitewise.connector` e precisa gravar dados da propriedade, você deve incluir a seguinte permissão em sua política. AWS IoT SiteWise

```

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

```

Se você usa o tipo de componente com `amazon.iotsitewise.connector.edgevideo`, deverá incluir permissões para o Kinesis Video Streams e o Kinesis Video Streams. AWS IoT SiteWise O exemplo de política a seguir mostra como incluir AWS IoT SiteWise permissões do Kinesis Video Streams em sua 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"
},
...

```

## Permissões para um conector a uma fonte de dados externa

Se criar um tipo de componente que usa uma função que se conecta a uma fonte de dados externa, você deve dar permissão ao seu perfil de serviço para usar a função do Lambda que implementa a função. Para obter mais informações sobre como criar tipos de componentes e funções, consulte [???](#).

O exemplo a seguir dá permissão ao seu perfil de serviço para usar uma função do Lambda.

```

{
  "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 obter mais informações sobre como criar funções e atribuir políticas e relações de confiança a elas usando o IAM console, o e o AWS CLI IAMAPI, consulte [Criação de uma função para delegar permissões a um. AWS service \(Serviço da AWS\)](#)

## Modifique sua IAM função de espaço de trabalho para usar o conector de dados Athena

Para usar o [conector de dados tabulares do AWS IoT TwinMaker Athena](#), você deve atualizar sua AWS IoT TwinMaker função no espaço de trabalho. IAM Adicione as seguintes permissões à sua IAM função no espaço de trabalho:

### Note

Essa IAM alteração só funciona para dados tabulares do Athena armazenados com o Amazon AWS Glue S3. Para usar o Athena com outras fontes de dados, você deve

configurar uma IAM função para o Athena, consulte [Gerenciamento de identidade e acesso no Athena](#).

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

Leia o [Gerenciamento de identidade e acesso no Athena](#) para obter mais informações sobre a configuração do Athena. IAM

## Para criar um workspace

Para criar e configurar seu primeiro espaço de trabalho, use as etapas a seguir.

### Note

Este tópico mostra como criar um espaço de trabalho simples com um único recurso. Para um espaço de trabalho completo com vários recursos, experimente a configuração de amostra no repositório de [AWS IoT TwinMaker amostras](#) do Github.

1. Na página inicial do [console do AWS IoT TwinMaker](#), escolha Espaços de trabalho no painel de navegação à esquerda.
2. Na página Espaços de trabalho, escolha Criar espaço de trabalho.
3. Na página Criar um espaço de trabalho, insira um nome para seu espaço de trabalho.
4. (Opcional) Adicione uma descrição para seu espaço de trabalho.
5. Em Recurso do S3, escolha Criar um bucket do S3. Essa opção cria um bucket do Amazon S3 onde AWS IoT TwinMaker armazena informações e recursos relacionados ao espaço de trabalho. Cada espaço de trabalho requer seu próprio bucket.
6. Em Função de execução, escolha Gerar automaticamente uma nova função ou a IAM função personalizada que você criou para esse espaço de trabalho.

Se você escolher Gerar automaticamente uma nova função, AWS IoT TwinMaker anexe uma política à função que concede permissão à nova função de serviço para acessar outros AWS serviços, incluindo permissão para ler e gravar no bucket do Amazon S3 que você especificou na etapa anterior. Para obter informações sobre como atribuir permissões a este perfil, consulte [???](#).

- Escolha Criar espaço de trabalho. O banner a seguir é exibido na parte superior da página Espaços de trabalho.



- Escolha Obter JSON. Recomendamos que você adicione a IAM política que você vê à IAM função AWS IoT TwinMaker criada para usuários e contas que visualizam o painel da Grafana. O nome dessa função segue esse padrão: *workspace-name*DashboardRole, Para obter instruções sobre como criar uma política e anexá-la a uma função, consulte [Modificação de uma política de permissões de função \(console\)](#).

O exemplo a seguir contém a política a ser adicionada ao perfil do painel.

```
{
  "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",
```

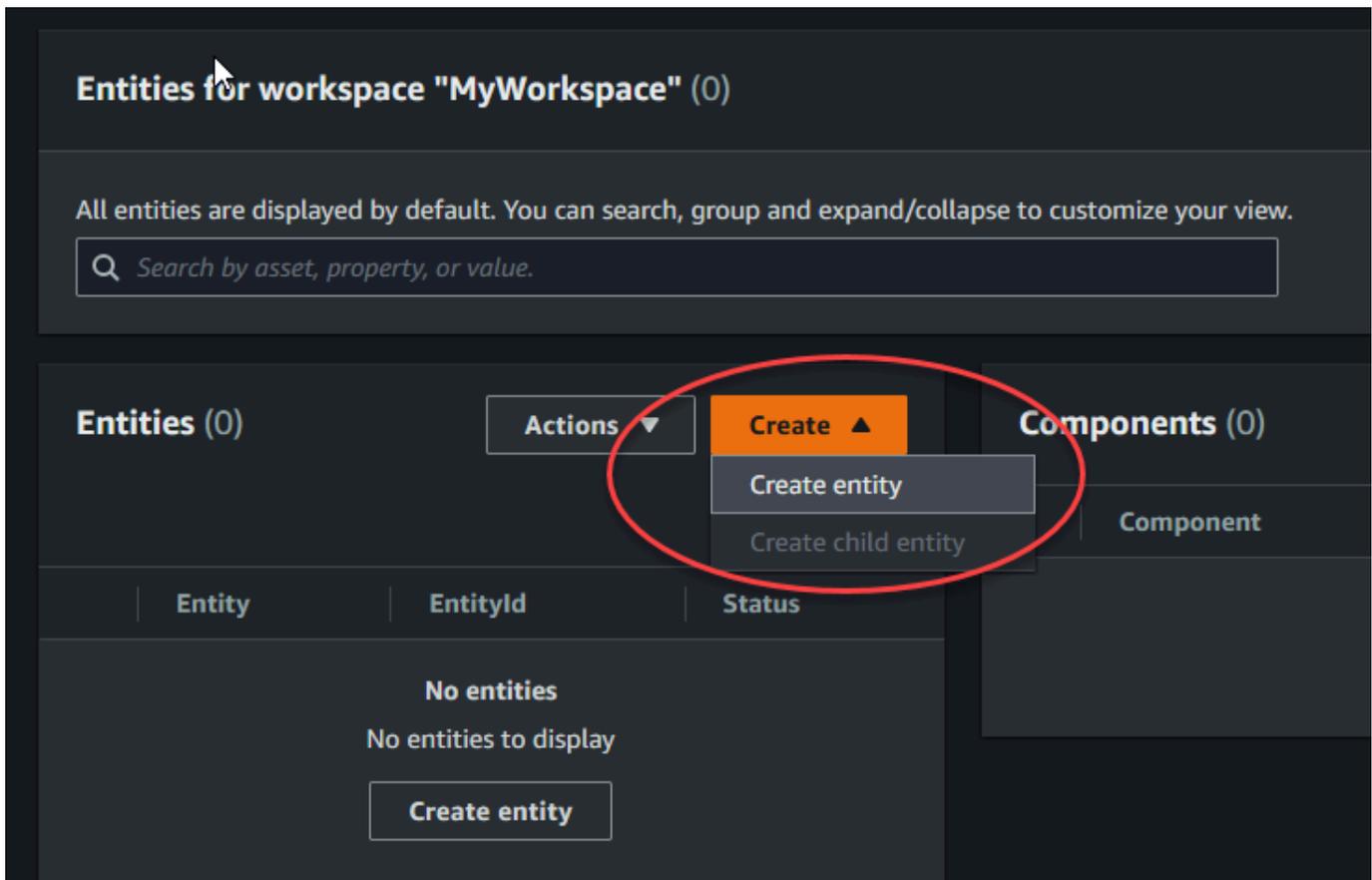
```
    "arn:aws:iottwinmaker:us-east-1:account-id:workspace/workspace-name/*"  
  ]  
},  
{  
  "Effect": "Allow",  
  "Action": "iottwinmaker:ListWorkspaces",  
  "Resource": "*"   
}  
]  
}
```

Agora você está pronto para começar a criar um modelo de dados para seu espaço de trabalho com sua primeira entidade. Para obter instruções sobre como fazer isso, consulte [Crie sua primeira entidade](#).

## Crie sua primeira entidade

Para criar sua primeira entidade, siga as etapas a seguir.

1. Na página Espaços de trabalho, escolha seu espaço de trabalho e, no painel esquerdo, escolha Entidades.
2. Na página Entidades, escolha Criar e, em seguida, Criar entidade.



3. Na janela Criar uma entidade, insira um nome para a sua entidade. Este exemplo usa uma entidade do **CookieMixer**.
4. (Opcional) Insira uma descrição para sua entidade.
5. Escolha Criar entidade.

As entidades contêm dados sobre cada item em seu espaço de trabalho. Você coloca dados em entidades adicionando componentes. AWS IoT TwinMaker fornece os seguintes tipos de componentes integrados.

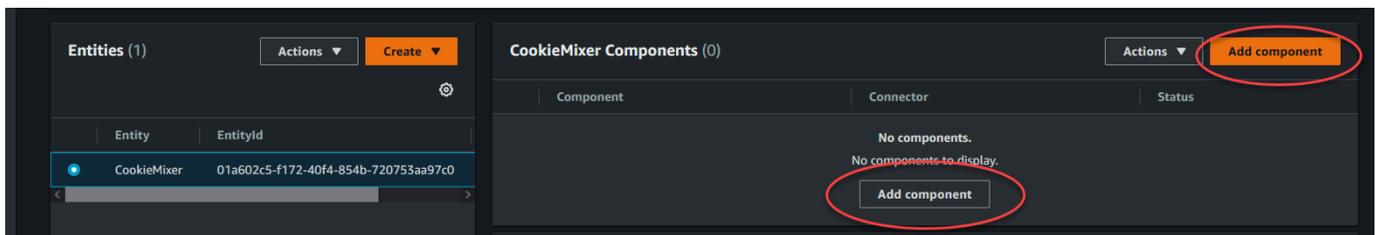
- Parâmetros: adiciona um conjunto de propriedades de chave-valor.
- Documento: adiciona um nome e um URL para um documento que contém informações sobre a entidade.
- Alarmes: conecta-se a uma fonte de dados de séries temporais de alarme.
- SiteWise conector: extrai propriedades de séries temporais definidas em um AWS IoT SiteWise ativo.

- Conector Edge para Kinesis AWS IoT Greengrass Video Streams: extrai dados de vídeo do Edge Connector para. KVS AWS IoT Greengrass Para obter mais informações, consulte [AWS IoT TwinMaker integração de vídeo](#).

Você pode ver esses tipos de componentes e suas definições escolhendo Tipos de componentes no painel esquerdo. Também é possível criar um novo tipo de componente na página Tipos de componentes. Para obter mais informações sobre a criação de tipos de componentes, consulte [Uso e criação de tipos de componentes](#).

Neste exemplo, criamos um componente de documento simples que adiciona informações descritivas sobre sua entidade.

1. Na página Entidades, escolha a entidade e, em seguida, escolha adicionar componente.



2. Na janela Adicionar componente, insira um nome para seu componente. Como esse exemplo usa uma entidade de mixer de cookies, inserimos **MixerDescription** no campo Nome.

**Add component** ✕

Name

MixerDescription

Type

Types of components include documents, time-series data, structured data, and unstructured data.

com.amazon.iottwinmaker.documents ▼

Edit form  Edit JSON

Document editor

No docs associated to the entity

**Add a doc**

▼ **Properties**

Property	Data type	is Timeseries	Storage
documents	Map ▼	False ▼	Internal ▼

Value

\_\_\_\_\_

**Add another property**

Cancel **Add component**

3. Escolha Adicionar um documento e, em seguida, insira valores para o nome do documento e o URL externo. Com o componente de documentos, você pode armazenar uma lista externa URLs que contém informações importantes sobre a entidade.

#### 4. Escolha Adicionar componente.

Agora, você está pronto para criar sua primeira cena. Para obter instruções sobre como fazer isso, consulte [Criação e edição de AWS IoT TwinMaker cenas](#).

## Configurando uma AWS conta

Se você não tiver um Conta da AWS, conclua as etapas a seguir para criar um.

Para se inscrever em um Conta da AWS

1. Abra a <https://portal.aws.amazon.com/billing/inscrição>.
2. Siga as instruções online.

Parte do procedimento de inscrição envolve receber uma chamada telefônica e inserir um código de verificação no teclado do telefone.

Quando você se inscreve em um Conta da AWS, um Usuário raiz da conta da AWS é criado. O usuário raiz tem acesso a todos os Serviços da AWS e atributos na conta. Como prática recomendada de segurança, atribua o acesso administrativo a um usuário e use somente o usuário-raiz para executar [tarefas que exigem acesso de usuário-raiz](#).

## Inscreva-se para um Conta da AWS

Se você não tiver um Conta da AWS, conclua as etapas a seguir para criar um.

Para se inscrever em um Conta da AWS

1. Abra a <https://portal.aws.amazon.com/billing/inscrição>.
2. Siga as instruções online.

Parte do procedimento de inscrição envolve receber uma chamada telefônica e inserir um código de verificação no teclado do telefone.

Quando você se inscreve em um Conta da AWS, um Usuário raiz da conta da AWS é criado. O usuário raiz tem acesso a todos os Serviços da AWS e atributos na conta. Como prática recomendada de segurança, atribua o acesso administrativo a um usuário e use somente o usuário-raiz para executar [tarefas que exigem acesso de usuário-raiz](#).

AWS envia um e-mail de confirmação após a conclusão do processo de inscrição. A qualquer momento, você pode visualizar a atividade atual da sua conta e gerenciar sua conta acessando <https://aws.amazon.com/e> escolhendo Minha conta.

## Criar um usuário com acesso administrativo

Depois de se inscrever em um Conta da AWS, proteja seu Usuário raiz da conta da AWS AWS IAM Identity Center, habilite e crie um usuário administrativo para que você não use o usuário root nas tarefas diárias.

Proteja seu Usuário raiz da conta da AWS

1. Faça login [AWS Management Console](#) como proprietário da conta escolhendo Usuário raiz e inserindo seu endereço de Conta da AWS e-mail. Na próxima página, insira sua senha.

Para obter ajuda ao fazer login usando o usuário raiz, consulte [Fazer login como usuário raiz](#) no Guia do usuário do Início de Sessão da AWS .

2. Ative a autenticação multifator (MFA) para seu usuário root.

Para obter instruções, consulte [Habilitar um MFA dispositivo virtual para seu usuário Conta da AWS root \(console\)](#) no Guia IAM do usuário.

Criar um usuário com acesso administrativo

1. Ative o IAM Identity Center.

Para obter instruções, consulte [Habilitar AWS IAM Identity Center](#) no Guia do usuário do AWS IAM Identity Center .

2. No IAM Identity Center, conceda acesso administrativo a um usuário.

Para ver um tutorial sobre como usar o Diretório do Centro de Identidade do IAM como fonte de identidade, consulte [Configurar o acesso do usuário com o padrão Diretório do Centro de Identidade do IAM](#) no Guia AWS IAM Identity Center do usuário.

Iniciar sessão como o usuário com acesso administrativo

- Para entrar com seu usuário do IAM Identity Center, use o login URL que foi enviado ao seu endereço de e-mail quando você criou o usuário do IAM Identity Center.

Para obter ajuda para fazer login usando um usuário do IAM Identity Center, consulte [Como fazer login no portal de AWS acesso](#) no Guia Início de Sessão da AWS do usuário.

### Atribuir acesso a usuários adicionais

1. No IAM Identity Center, crie um conjunto de permissões que siga as melhores práticas de aplicação de permissões com privilégios mínimos.

Para obter instruções, consulte [Create a permission set](#) no Guia do usuário do AWS IAM Identity Center .

2. Atribua usuários a um grupo e, em seguida, atribua o acesso de autenticação única ao grupo.

Para obter instruções, consulte [Add groups](#) no Guia do usuário do AWS IAM Identity Center .

# Uso e criação de tipos de componentes

Este tópico mostra os valores e as estruturas que você usa para criar um tipo de AWS IoT TwinMaker componente. Ele mostra como criar um objeto de solicitação que você pode passar para a [CreateComponentType](#) API ou usando o editor de tipo de componente no AWS IoT TwinMaker console.

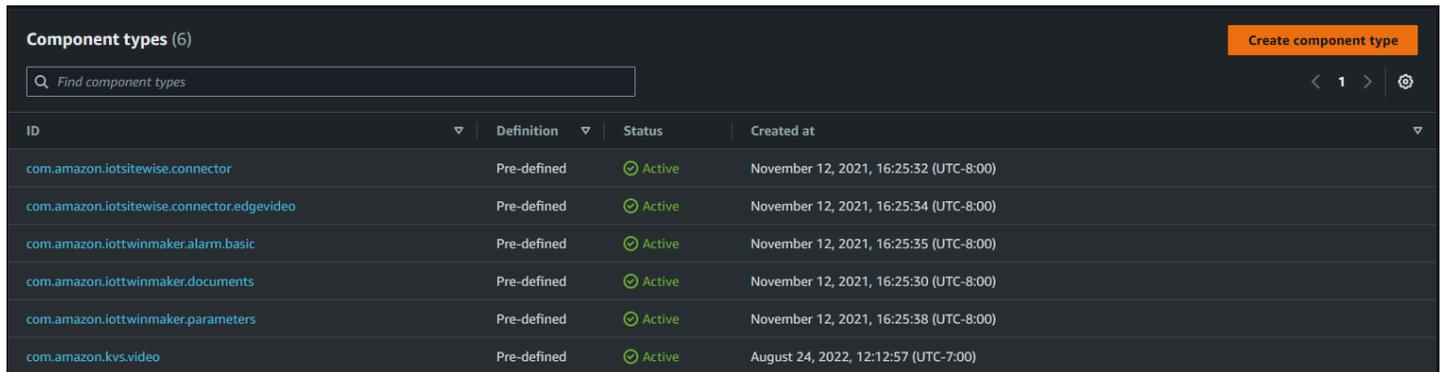
Os componentes fornecem contexto para propriedades e dados de suas entidades associadas.

## Tipos de componentes integrados

No AWS IoT TwinMaker console, quando você escolhe um espaço de trabalho e, em seguida, escolhe Tipos de componentes no painel esquerdo, você vê os seguintes tipos de componentes.

- `com.amazon.iotsitewise.resourcesync`: um tipo de componente que sincroniza automaticamente seus ativos e modelos de ativos e os converte em entidades, componentes e tipos de componentes. AWS IoT SiteWise AWS IoT TwinMaker Para obter mais informações sobre como usar a sincronização AWS IoT SiteWise de ativos, consulte [Sincronização de ativos com AWS IoT SiteWise](#).
- `com.amazon.iottwinmaker.alarm.basic`: um componente básico de alarme que extrai dados de alarme de uma fonte externa para uma entidade. Esse componente não contém uma função que se conecta a uma fonte de dados específica. Isso significa que o componente de alarme é abstrato e pode ser herdado por outro tipo de componente que especifica uma fonte de dados e uma função que lê a partir dessa fonte.
- `com.amazon.iottwinmaker.documents`: um mapeamento simples de títulos para URLs para documentos que contêm informações sobre uma entidade.
- `com.amazon.iotsitewise.connector.edgevideo`: um componente que extrai vídeo de um dispositivo de IoT usando o componente Edge Connector for Kinesis Video Streams em uma entidade. AWS IoT Greengrass O componente [Edge Connector para Kinesis AWS IoT Greengrass Video](#) Streams AWS IoT TwinMaker não é um componente, mas sim AWS IoT Greengrass um componente pré-construído que é implantado localmente em seu dispositivo de IoT.
- `com.amazon.iotsitewise.connector`: um componente que extrai dados do AWS IoT SiteWise para uma entidade.
- `com.amazon.iottwinmaker.parameters`: um componente que adiciona pares estáticos de valores-chave a uma entidade.

- `com.amazon.kvs.video`: um componente que puxa o vídeo do Kinesis Video Streams para uma entidade. AWS IoT TwinMaker



ID	Definition	Status	Created at
<code>com.amazon.iotsitewise.connector</code>	Pre-defined	Active	November 12, 2021, 16:25:32 (UTC-8:00)
<code>com.amazon.iotsitewise.connector.edgevideo</code>	Pre-defined	Active	November 12, 2021, 16:25:34 (UTC-8:00)
<code>com.amazon.iottwinmaker.alarm.basic</code>	Pre-defined	Active	November 12, 2021, 16:25:35 (UTC-8:00)
<code>com.amazon.iottwinmaker.documents</code>	Pre-defined	Active	November 12, 2021, 16:25:30 (UTC-8:00)
<code>com.amazon.iottwinmaker.parameters</code>	Pre-defined	Active	November 12, 2021, 16:25:38 (UTC-8:00)
<code>com.amazon.kvs.video</code>	Pre-defined	Active	August 24, 2022, 12:12:57 (UTC-7:00)

## Principais características dos tipos de AWS IoT TwinMaker componentes

A lista a seguir descreve os principais atributos dos tipos de componentes.

- Definições de propriedade: o [PropertyDefinitionRequest](#) objeto define uma propriedade que você pode preencher no compositor de cenas ou pode ser preenchida com dados extraídos de fontes de dados externas. As propriedades estáticas que você define são armazenadas em AWS IoT TwinMaker. As propriedades de séries temporais e outras propriedades extraídas de fontes de dados são armazenadas externamente.

Você especifica as definições de propriedades dentro de uma string no mapa `PropertyDefinitionRequest`. Cada string deve ser exclusiva do mapa.

- Funções: o [FunctionRequest](#) objeto especifica uma função Lambda que lê e potencialmente grava em uma fonte de dados externa.

Um tipo de componente que contém uma propriedade com um valor armazenado externamente, mas que não tem uma função correspondente para recuperar os valores, é um tipo de componente abstrato. Você pode estender tipos de componentes concretos a partir de um tipo de componente abstrato. Você não pode adicionar tipos de componentes abstratos a uma entidade. Eles não aparecem na cena do compositor.

Você especifica funções dentro de uma string para o mapa `FunctionRequest`. A string deve especificar um dos seguintes tipos de função predefinidos.

- `dataReader`: uma função que extrai dados de uma fonte externa.
- `dataReaderByEntity`: uma função que extrai dados de uma fonte externa.

Quando você usa esse tipo de leitor de dados, a operação da [GetPropertyValueHistory](#) API suporta somente consultas específicas da entidade para propriedades nesse tipo de componente. (Você pode solicitar apenas o histórico de valores da propriedade para `componentName + entityId`.)

- `dataReaderByComponentType`: uma função que extrai dados de uma fonte externa.

Quando você usa esse tipo de leitor de dados, a operação da [GetPropertyValueHistory](#) API suporta somente consultas entre entidades para propriedades nesse tipo de componente. (Você pode apenas solicitar o histórico de valores da propriedade para `componentTypeId`.)

- `dataWriter`: uma função que grava dados em uma fonte externa.
- `schemaInitializer`: uma função que inicializa automaticamente os valores das propriedades sempre que você cria uma entidade que contém o tipo de componente.

Um dos três tipos de funções do leitor de dados é necessário em um tipo de componente não abstrato.

Para ver um exemplo de uma função do Lambda que implementa componentes de telemetria de fluxo de tempo, incluindo alarmes, consulte o leitor de dados em [Exemplos AWS IoT TwinMaker](#).

#### Note

Como o conector de alarme herda do tipo de componente de alarme abstrato, a função do Lambda deve retornar o valor de `alarm_key`. Se você não retornar esse valor, o Grafana não o reconhecerá como um alarme. Isso é necessário para todos os componentes que retornam alarmes.

- Herança: os tipos de componentes promovem a reutilização do código por meio de herança. Um tipo de componente pode herdar até dez tipos de componentes primários.

Use o parâmetro `extendsFrom` para especificar os tipos de componentes dos quais seu tipo de componente herda propriedades e funções.

- `IsSingleton`: alguns componentes contêm propriedades, como coordenadas de localização, que não podem ser incluídas mais de uma vez em uma entidade. Defina o valor do parâmetro

`isSingleton` para `true` para indicar que seu tipo de componente pode ser incluído somente uma vez em uma entidade.

## Criação de definições de propriedade

A tabela a seguir descreve os parâmetros de um `PropertyDefinitionRequest`.

Parâmetro	Descrição
<code>isExternalId</code>	<p>Um booleano que especifica se a propriedade é um identificador exclusivo (como um ID de AWS IoT SiteWise ativo) de um valor de propriedade armazenado externamente.</p> <p>O valor padrão da propriedade é <code>false</code>.</p>
<code>isStoredExternally</code>	<p>Um booleano que especifica se o valor da propriedade é armazenado externamente.</p> <p>O valor padrão da propriedade é <code>false</code>.</p>
<code>isTimeSeries</code>	<p>Um booleano que especifica se a propriedade armazena dados de séries temporais.</p> <p>O valor padrão dessa propriedade é <code>false</code>.</p>
<code>isRequiredInEntity</code>	<p>Um booleano que especifica se a propriedade deve ter um valor em uma entidade que usa o tipo de componente.</p>
<code>dataType</code>	<p>Um <a href="#">DataType</a> objeto que especifica o tipo de dados (como string, mapa, lista e unidade de medida) da propriedade.</p>
<code>defaultValue</code>	<p>Um <a href="#">DataValue</a> objeto que especifica o valor padrão da propriedade.</p>

Parâmetro	Descrição
<code>configuration</code>	Um string-to-string mapa que especifica informações adicionais que você precisa para se conectar a uma fonte de dados externa.

## Criar funções

A tabela a seguir descreve os parâmetros de um `FunctionRequest`.

Parâmetro	Descrição
<code>implementedBy</code>	Um <a href="#">DataConnector</a> objeto que especifica a função Lambda que se conecta à fonte de dados externa.
<code>requiredProperties</code>	Uma lista de propriedades de que a função precisa para ler e gravar em uma fonte de dados externa.
<code>scope</code>	O escopo da função. Use <code>Workspace</code> para funções com um escopo que abrange um espaço de trabalho inteiro. Use <code>Entity</code> para funções com um escopo limitado à entidade que contém o componente.

Para exemplos que mostram como criar e estender tipos de componentes, consulte o [???](#).

## Exemplo de tipos de componentes

Este tópico contém exemplos que mostram como implementar conceitos-chave de tipos de componentes.

## Alarme (abstrato)

O exemplo a seguir é o tipo de componente de alarme abstrato que aparece no AWS IoT TwinMaker console. Contém uma lista de functions que consiste em um dataReader que não tem valor de 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  
  }  
}  
}
```

### Observações:

Valores para `componentTypeId` e `workspaceId` são obrigatórios. O valor de `componentTypeId` deve ser exclusivo para o seu espaço de trabalho. O valor de `alarm_key` é um identificador exclusivo que uma função pode usar para recuperar dados de alarme de uma fonte externa. O valor da chave é obrigatório e armazenado em AWS IoT TwinMaker. Os valores das séries temporais `alarm_status` são armazenados na fonte externa.

Mais exemplos estão disponíveis em [AWS IoT TwinMaker Exemplos](#).

## Telemetria do fluxo de tempo

O exemplo a seguir é um tipo de componente simples que recupera dados de telemetria sobre um tipo específico de componente (como um alarme ou um misturador de cookies) de uma fonte externa. Especifica uma função do Lambda que os tipos de componentes herdam.

```
{  
  "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
    }
}
}
}

```

## Alarme (herda do alarme abstrato)

O exemplo a seguir herda os tipos de componentes de alarme abstrato e telemetria de fluxo de tempo. Especifica sua própria função do Lambda que recupera dados de alarme.

```

{
  "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 o conector de alarme herda do tipo de componente de alarme abstrato, a função do Lambda deve retornar o valor de `alarm_key`. Se você não retornar esse valor, o Grafana não o reconhecerá como um alarme. Isso é necessário para todos os componentes que retornam alarmes.

## Exemplos de equipamentos

O exemplo nesta seção mostra como modelar equipamentos em potencial. Você pode usar esses exemplos para obter algumas ideias sobre como modelar equipamentos nos seus próprios processos.

### Mixer de cookies

O exemplo a seguir é herdado do tipo de componente de telemetria de fluxo de tempo. Especifica propriedades adicionais de séries temporais para a taxa de alternância e temperatura de um mixer 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
    }
}

```

## Tanque de água

O exemplo a seguir é herdado do tipo de componente de telemetria de fluxo de tempo. Especifica propriedades adicionais de séries temporais para o volume e a vazão de um tanque de água.

```

{
  "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
    }
  }
}

```

```
}  
}
```

## Localização do espaço

O exemplo a seguir contém propriedades, cujos valores são armazenados em AWS IoT TwinMaker. Como os valores são especificados pelos usuários e armazenados internamente, nenhuma função é necessária para recuperá-los. O exemplo também usa o tipo de dados RELATIONSHIP para especificar um relacionamento com outro tipo de componente.

Esse componente fornece um mecanismo leve para adicionar contexto a um gêmeo digital. Você pode usá-lo para adicionar metadados indicando onde algo está localizado. Você também pode usar essas informações na lógica usada para determinar quais câmeras podem ver um equipamento ou espaço, ou para saber como enviar alguém para um local.

```
{  
  "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 operações em massa

Use a `metadataTransferJob` para transferir e gerenciar seus AWS IoT TwinMaker recursos em grande escala. A `metadataTransferJob` permite que você execute operações em massa e transfira recursos entre AWS IoT TwinMaker e AWS IoT SiteWise e o Amazon S3.

Você pode usar operações em massa nos seguintes cenários:

- Migração em massa de ativos e dados entre contas, por exemplo, migração de uma conta de desenvolvimento para uma conta de produção.
- Gerenciamento de ativos em grande escala, como upload e edição de AWS IoT ativos em grande escala.
- Importação em massa de seus ativos para AWS IoT TwinMaker AWS IoT SiteWise e.
- Importação em massa de AWS IoT TwinMaker entidades de arquivos de ontologia existentes, como BIM arquivos `revit` ou.

## Tópicos

- [Principais conceitos e terminologia](#)
- [Executando operações de importação e exportação em massa](#)
- [AWS IoT TwinMaker esquema de trabalho de transferência de metadados](#)

## Principais conceitos e terminologia

AWS IoT TwinMaker as operações em massa usam os seguintes conceitos e terminologia:

- **Importar:** a ação de mover recursos para um AWS IoT TwinMaker espaço de trabalho. Por exemplo, de um arquivo local, um arquivo em um bucket do Amazon S3 ou de um AWS IoT TwinMaker espaço AWS IoT SiteWise de trabalho.
- **Exportação:** a ação de mover recursos de um AWS IoT TwinMaker espaço de trabalho para uma máquina local ou um bucket do Amazon S3.
- **Fonte:** O local inicial de onde você deseja mover os recursos.

Por exemplo, um bucket do Amazon S3 é uma fonte de importação e um AWS IoT TwinMaker espaço de trabalho é uma fonte de exportação.

- Destino: o local desejado para onde você deseja mover seus recursos.

Por exemplo, um bucket do Amazon S3 é um destino de exportação e um AWS IoT TwinMaker espaço de trabalho é um destino de importação.

- AWS IoT SiteWise Esquema: Um esquema usado para importar e exportar recursos de e para AWS IoT SiteWise
- AWS IoT TwinMaker Esquema: Um esquema usado para importar e exportar recursos de e para AWS IoT TwinMaker
- AWS IoT TwinMaker recursos de nível superior: recursos usados em recursos existentes APIs. Especificamente, uma entidade ou uma ComponentType.
- AWS IoT TwinMaker recursos de subnível: tipos de recursos aninhados usados nas definições de metadados. Especificamente, um componente.
- Metadados: informações importantes necessárias para importar ou exportar AWS IoT SiteWise AWS IoT TwinMaker recursos com sucesso.
- metadataTransferJob: o objeto criado quando você executa `CreateMetadataTransferJob`.

## AWS IoT TwinMaker metadataTransferJob funcionalidade

Este tópico explica o comportamento que AWS IoT TwinMaker ocorre quando você executa uma operação em massa — como a `metadataTransferJob` é processada. Também explica como definir um esquema com os metadados necessários para transferir seus recursos. AWS IoT TwinMaker as operações em massa oferecem suporte às seguintes funcionalidades:

- Criação ou substituição de recursos de nível superior: AWS IoT TwinMaker criará novos recursos ou substituirá todos os recursos existentes identificados exclusivamente por uma ID de recurso.

Por exemplo, se uma entidade existir no sistema, a definição da entidade será substituída pela nova definida no modelo abaixo da `Entity` chave.

- Criação ou substituição do sub-recurso:

No `EntityComponent` nível, você só pode criar ou substituir um componente. A entidade já deve existir, caso contrário, a ação produzirá uma `ValidationException`.

No nível da propriedade ou do relacionamento, você só pode criar ou substituir uma propriedade ou relacionamento, e o conteúdo já `EntityComponent` deve existir.

- Exclusão do sub-recurso:

AWS IoT TwinMaker também oferece suporte à exclusão de sub-recursos. Um sub-recurso pode ser um componente, propriedade ou relacionamento.

Se quiser excluir um componente, você deve fazer isso no nível da entidade.

Se quiser excluir uma propriedade ou relacionamento, você deve fazer isso a partir da Entidade ou do EntityComponent nível.

Para excluir um sub-recurso, você atualiza o recurso de nível superior e omite a definição do sub-recurso.

- Sem exclusão de recursos de nível superior: nunca AWS IoT TwinMaker excluirá recursos de nível superior. Um recurso de nível superior se refere a uma entidade ou ComponentType.
- Nenhuma definição de sub-recurso para o mesmo recurso de nível superior em um modelo:

Você não pode fornecer a definição completa da entidade e a definição do sub-recurso (como propriedade) da mesma entidade no mesmo modelo.

Se um EntityID for usado em Entity, você não poderá usar o mesmo ID em Entity EntityComponent, propriedade ou relacionamento.

Se uma combinação entityID ou componentName for usada em EntityComponent, você não poderá usar a mesma combinação em EntityComponent, propriedade ou relacionamento.

Se uma combinação entityID, componentName, propertyName for usada na propriedade ou no relacionamento, você não poderá usar a mesma combinação na propriedade ou no relacionamento.

- ExternalID é opcional para AWS IoT TwinMaker: O ExternalID pode ser usado para ajudar você a identificar seus recursos.

## Executando operações de importação e exportação em massa

Este tópico aborda como realizar operações de importação e exportação em massa e como lidar com erros em suas tarefas de transferência. Ele fornece exemplos de trabalhos de transferência usando comandos da CLI.

A Referência AWS IoT TwinMaker da API contém informações sobre as [CreateMetadataTransferJob](#) outras ações da API.

## Tópicos

- [metadataTransferJob pré-requisitos](#)
- [permissões do IAM](#)
- [Execute uma operação em massa](#)
- [Gerenciamento de erros](#)
- [Importar modelos de metadados](#)
- [AWS IoT TwinMaker metadataTransferJob exemplos](#)

## metadataTransferJob pré-requisitos

Preencha os seguintes pré-requisitos antes de executar um: metadataTransferJob

- Crie um AWS IoT TwinMaker espaço de trabalho. O espaço de trabalho pode ser o destino de importação ou a fonte de exportação de um metadataTransferJob. Para obter informações sobre como criar um espaço de trabalho, consulte [Para criar um workspace](#).
- Crie um bucket do Amazon S3 para armazenar recursos. Para obter mais informações sobre o uso do Amazon S3, consulte [O que é o Amazon S3?](#)

## permissões do IAM

Ao realizar operações em massa, você precisa criar uma política do IAM com permissões para permitir a troca de AWS recursos entre o Amazon S3,, AWS IoT TwinMaker AWS IoT SiteWise, e sua máquina local. Para obter mais informações sobre a criação de políticas do IAM, consulte [Como criar políticas do IAM](#).

As declarações de política do Amazon S3 AWS IoT SiteWise e do Amazon S3 estão listadas aqui:  
AWS IoT TwinMaker

- AWS IoT TwinMaker política:

```
{
  "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:

```

{
  "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 do Amazon S3:

```

{
  "Effect": "Allow",
  "Action": [
    "s3:PutObject",
    "s3:GetObject",
    "s3:GetBucketLocation",
    "s3:ListBucket",
    "s3:AbortMultipartUpload",
    "s3:ListBucketMultipartUploads",

```

```

    "s3:ListMultipartUploadParts"
  ],
  "Resource": "*"
}

```

Como alternativa, você pode definir o escopo de sua política do Amazon S3 para acessar somente um único bucket do Amazon S3, consulte a política a seguir.

### Política de escopo de bucket único do Amazon S3

```

{
  "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/*"
  ]
}

```

## Defina o controle de acesso para um metadataTransferJob

Para controlar o tipo de trabalhos que um usuário pode acessar, adicione a seguinte política do IAM à função usada para chamar AWS IoT TwinMaker.

### Note

Essa política só permite o acesso a trabalhos de AWS IoT TwinMaker importação e exportação que transferem recursos de e para o Amazon S3.

```

{
  "Effect": "Allow",

```

```

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

```

## Execute uma operação em massa

Esta seção aborda como realizar operações de importação e exportação em massa.

### Importar dados do Amazon S3 para AWS IoT TwinMaker

1. Especifique os recursos que você deseja transferir usando o AWS IoT TwinMaker `metadataTransferJob` esquema. Crie e armazene seu arquivo de esquema em seu bucket do Amazon S3.

Por exemplo, esquemas, consulte [Importar modelos de metadados](#).

2. Crie um corpo de solicitação e salve-o como um arquivo JSON. O corpo da solicitação especifica a fonte e o destino do trabalho de transferência. Certifique-se de especificar seu bucket do Amazon S3 como origem e seu AWS IoT TwinMaker espaço de trabalho como destino.

Veja a seguir um exemplo de corpo de solicitação:

```

{
  "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"
      }
    }
  }
}

```

Registre o nome do arquivo que você deu ao corpo da solicitação. Você precisará dele na próxima etapa. Neste exemplo, o corpo da solicitação é nomeado `createMetadataTransferJobImport.json`.

3. Execute o seguinte comando CLI para invocar `CreateMetadataTransferJob` (substitua o nome do arquivo `input-json` pelo nome que você deu ao corpo da solicitação):

```

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

```

Isso cria `metadataTransferJob` e inicia o processo de transferência dos recursos selecionados.

## Exportar dados AWS IoT TwinMaker para o Amazon S3

1. Crie um corpo de solicitação JSON com os filtros apropriados para escolher os recursos que você deseja exportar. Para este exemplo, usamos:

```

{
  "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"
    }
  }
}

```

A `filters` matriz permite especificar quais recursos serão exportados. Neste exemplo, filtramos `parentity`, `componentType` e.

Certifique-se de especificar seu AWS IoT TwinMaker espaço de trabalho como origem e seu bucket do Amazon S3 como destino do trabalho de transferência de metadados.

Salve o corpo da solicitação e registre o nome do arquivo. Você precisará dele na próxima etapa. Neste exemplo, nomeamos nosso corpo de solicitação `createMetadataTransferJobExport.json`.

2. Execute o seguinte comando CLI para invocar `CreateMetadataTransferJob` (substitua o nome do arquivo `input-json` pelo nome que você deu ao corpo da solicitação):

```

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

```

Isso cria `metadataTransferJob` e inicia o processo de transferência dos recursos selecionados.

Para verificar ou atualizar o status de um trabalho de transferência, use os seguintes comandos:

- Para cancelar um trabalho, use a ação [CancelMetadataTransferJob](#) da API. Quando você chama `CancelMetadataTransferJob`, a API só cancela uma execução `metadataTransferJob`, e quaisquer recursos já exportados ou importados não são afetados por essa chamada de API.

- Para recuperar informações sobre um trabalho específico, use a ação da [GetMetadataTransferJobAPI](#).

Ou você pode chamar um trabalho GetMetadataTransferJob de transferência existente com o seguinte comando da CLI:

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

Se você chamar GetMetadataTransferJob um trabalho de AWS IoT TwinMaker importação ou exportação inexistente, receberá uma resposta ResourceNotFoundException de erro.

- Para listar os trabalhos atuais, use a ação [ListMetadataTransferJobs](#) da API.

Aqui está um exemplo de CLI que chama ListMetadataTransferJobs com as as DestinationType e AWS IoT TwinMaker as SourceType: s3

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

#### Note

Você pode alterar os valores dos parâmetros SourceType e DestinationType para que correspondam à origem e ao destino do seu trabalho de importação ou exportação.

Para obter mais exemplos de comandos da CLI que invocam essas ações de API, consulte [AWS IoT TwinMaker metadataTransferJob exemplos](#)

Se você encontrar algum erro durante o trabalho de transferência, consulte [Gerenciamento de erros](#).

## Gerenciamento de erros

Depois de criar e executar um trabalho de transferência, você pode ligar GetMetadataTransferJob para diagnosticar qualquer erro que tenha ocorrido:

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

Depois de ver o estado do trabalho para o qual se COMPLETED candidata, você pode verificar os resultados do trabalho. GetMetadataTransferJob retorna um objeto chamado [MetadataTransferJobProgress](#) que contém os seguintes campos:

- FailedCount: indica o número de recursos que falharam durante o processo de transferência.
- skippedCount: indica o número de recursos que foram ignorados durante o processo de transferência.
- SucceededCount: indica o número de recursos que foram bem-sucedidos durante o processo de transferência.
- TotalCount: indica a contagem total de recursos envolvidos no processo de transferência.

Além disso, é retornado um elemento reportUrl que contém uma URL pré-assinada. Se sua tarefa de transferência tiver erros que você deseja investigar mais detalhadamente, você pode baixar um relatório completo de erros usando esse URL.

## Importar modelos de metadados

Você pode importar vários componentes, ComponentTypes ou entidades com uma única operação de importação em massa. Os exemplos desta seção mostram como fazer isso.

template: Importing entities

Use o seguinte formato de modelo para um trabalho que importa 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

Use o seguinte formato de modelo para um trabalho que importa ComponentTypes:

```

{
  "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

Use o seguinte formato de modelo para uma tarefa que importa 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 exemplos

Use os comandos a seguir para gerenciar suas transferências de metadados:

- [CreateMetadataTransferJob](#) Ação da API.

Exemplo de comando CLI:

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

- Para cancelar um trabalho, use a ação [CancelMetadataTransferJob](#) da API.

Exemplo de comando CLI:

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

Quando você liga `CancelMetadataTransferJob`, ele cancela apenas um trabalho específico de transferência de metadados, e os recursos já exportados ou importados não são afetados.

- Para recuperar informações sobre um trabalho específico, use a ação da [GetMetadataTransferJob](#) API.

Exemplo de comando CLI:

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

```

- Para listar os trabalhos atuais, use a ação [ListMetadataTransferJobs](#) da API.

Você pode filtrar os resultados retornados `ListMetadataTransferJobs` usando um arquivo JSON.

Veja o procedimento a seguir usando a CLI:

1. Crie um arquivo JSON de entrada CLI para especificar os filtros que você deseja usar:

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

```

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

```

Salve-o e registre o nome do arquivo, você precisará dele ao inserir o comando CLI.

2. Use o arquivo JSON como argumento para o seguinte comando da CLI:

```

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

```

## AWS IoT TwinMaker esquema de trabalho de transferência de metadados

metadataTransferJob esquema de importação: use esse esquema de AWS IoT TwinMaker metadados para validar seus dados ao carregá-los em um bucket do 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": "\\$R00T|^[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": "$ROOT"
  },
  "DisplayName": {
    "description": "A friendly name for the property.",
    "type": "string",
    "pattern": ".*[^\u0000-\u001F\u007F]*.*",
    "minLength": 0,
    "maxLength": 256
  },
  "Tags": {
    "description": "Metadata that you can use to manage the entity / componentType",
    "patternProperties": {
      "^[a-zA-Z0-9_./+=\\-@]*$": {
        "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
  }
}
}
```

```
}
```

Aqui está um exemplo que cria um novo `ComponentType` chamado `component.type.initial` e cria uma entidade chamada `initial`:

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

Aqui está um exemplo que atualiza 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 dados

AWS IoT TwinMaker usa uma arquitetura baseada em conectores para que você possa conectar dados do seu próprio armazenamento de dados a AWS IoT TwinMaker. Isso significa que você não precisa migrar dados antes de usar AWS IoT TwinMaker. Atualmente, AWS IoT TwinMaker suporta conectores primários para AWS IoT SiteWise. Se você armazenar dados de modelagem e propriedade AWS IoT SiteWise, não precisará implementar seus próprios conectores. Se você armazenar seus dados de modelagem ou propriedade em outros armazenamentos de dados, como Timestream, DynamoDB ou Snowflake, deverá implementar AWS Lambda conectores com a interface do conector de AWS IoT TwinMaker dados para que possa invocar seu conector quando necessário. AWS IoT TwinMaker

## Tópicos

- [AWS IoT TwinMaker conectores de dados](#)
- [AWS IoT TwinMaker Conector de dados tabulares Athena](#)
- [Desenvolvendo AWS IoT TwinMaker conectores de dados de séries temporais](#)

# AWS IoT TwinMaker conectores de dados

Os conectores precisam acessar seu armazenamento de dados subjacente para resolver as consultas enviadas e retornar resultados ou um erro.

Para saber mais sobre os conectores disponíveis, suas interfaces de solicitação e suas interfaces de resposta, consulte os tópicos a seguir.

Para obter mais informações sobre as propriedades usadas nas interfaces do conector, consulte a ação da [GetPropertyValueHistory](#) API.

### Note

Alguns conectores têm dois campos de registro de data e hora nas interfaces de solicitação e resposta para propriedades de hora de início e hora de término. Ambos `startDateTime` e `endDateTime` usam um número longo para representar o segundo epoch, que não é mais suportado. Para manter a compatibilidade com versões anteriores, ainda enviamos um valor do registro de data e hora para esse campo, mas recomendamos usar os campos `startTime` e `endTime` que sejam consistentes com o formato de data e hora da nossa API.

## Tópicos

- [Conector inicializador de esquema](#)
- [DataReaderByEntity](#)
- [DataReaderByComponentType](#)
- [DataReader](#)
- [AttributePropertyValueReaderByEntity](#)
- [DataWriter](#)
- [Exemplos](#)

## Conector inicializador de esquema

Você pode usar o inicializador de esquema no tipo de componente ou no ciclo de vida da entidade para buscar o tipo de componente ou as propriedades do componente na fonte de dados subjacente. O inicializador de esquema importa automaticamente o tipo de componente ou as propriedades do componente sem chamar explicitamente uma ação de API para configurar `properties`.

### SchemaInitializer interface de solicitação

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

#### Note

O mapa de propriedades nessa interface de solicitação é um `PropertyRequest`. Para obter mais informações, consulte [PropertyRequest](#).

## SchemaInitializer interface de resposta

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

### Note

O mapa de propriedades nessa interface de solicitação é um `PropertyResponse`. Para obter mais informações, consulte [PropertyResponse](#).

## DataReaderByEntity

`DataReaderByEntity` é um conector de plano de dados usado para obter os valores de séries temporais das propriedades em um único componente.

Para obter informações sobre os tipos de propriedade, a sintaxe e o formato desse conector, consulte a ação da [GetPropertyValueHistoryAPI](#).

## DataReaderByEntityinterface de solicitação

```
{
  "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"
}

```

## DataReaderByEntityinterface de resposta

```

{
  "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 obter os valores de séries temporais de propriedades comuns que vêm do mesmo tipo de componente, use o conector `DataReaderByEntity` do plano de dados. Por exemplo, caso defina propriedades de séries temporais no tipo de componente e tiver vários componentes usando esse mesmo tipo, poderá consultar tais propriedades em todos os componentes em um determinado intervalo de tempo. Um caso de uso comum sobre isso é quando você deseja consultar o status do alarme de vários componentes para obter uma visão global de suas entidades.

Para obter informações sobre os tipos de propriedade, a sintaxe e o formato desse conector, consulte a ação da [GetPropertyValueHistory](#) API.

## DataReaderByComponentType interface de solicitação

```
{
  "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 interface de resposta

```
{
  "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` é um conector de plano de dados que pode lidar com o caso de `DataReaderByEntity` `DataReaderByComponentType` e.

Para obter informações sobre os tipos de propriedade, a sintaxe e o formato desse conector, consulte a ação da [GetPropertyValueHistoryAPI](#).

### `DataReader` interface de solicitação

Os `EntityId` e `componentName` são opcionais.

```
{
  "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` interface de resposta

```
{
  "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` é um conector de plano de dados que você pode usar para buscar o valor das propriedades estáticas em uma única entidade.

Para obter informações sobre os tipos de propriedade, a sintaxe e o formato desse conector, consulte a ação da [GetPropertyValueAPI](#).

### AttributePropertyValueReaderByEntity interface de solicitação

```

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

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

```

### AttributePropertyValueReaderByEntity interface de resposta

```

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

```

```

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

```

## DataWriter

DataWriter é um conector de plano de dados que você pode usar para gravar pontos de dados de séries temporais no armazenamento de dados subjacente para propriedades em um único componente.

Para obter informações sobre os tipos de propriedade, a sintaxe e o formato desse conector, consulte a ação da [BatchPutPropertyValuesAPI](#).

### DataWriter interface de solicitação

```

{
  "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 interface de resposta

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

## Exemplos

Os exemplos de JSON a seguir são exemplos de sintaxe de resposta e solicitação para vários conectores.

- SchemaInitializer:

Os exemplos a seguir mostram o inicializador de esquema em um ciclo de vida do tipo componente.

Solicitação:

```
{
  "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"
    }
  }
}

```

**Resposta:**

```

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

```

```

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

```

- O inicializador de esquema no ciclo de vida da entidade:

Solicitação:

```

{
  "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"
    }
}
}

```

Resposta:

```

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

```

- **DataReaderByEntity e DataReader:**

Solicitação:

```

{
  "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
    }
  }
}
```

Resposta:

```
{
  "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:**

Solicitação:

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

```
    }
  }
}
```

Resposta:

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

• **DataWriter:**

Solicitação:

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

Resposta:

```
{
  "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 dados tabulares Athena

Com o conector de dados tabulares do Athena, você pode acessar e usar seus armazenamentos de dados do Athena em AWS IoT TwinMaker. Você pode usar seus dados do Athena para criar gêmeos digitais sem um esforço intensivo de migração de dados. Você pode usar o conector pré-construído ou criar um conector Athena personalizado para acessar dados de suas fontes de dados do Athena.

## AWS IoT TwinMaker Pré-requisitos do conector de dados Athena

Antes de usar o conector de dados tabulares do Athena, preencha os seguintes pré-requisitos:

- Crie tabelas gerenciadas do Athena e seus recursos associados do Amazon S3. Para obter informações sobre o uso do Athena, consulte [a documentação do Athena](#).
- Crie um AWS IoT TwinMaker espaço de trabalho. É possível criar um espaço de trabalho no [console do AWS IoT TwinMaker](#).
- Atualize sua função do IAM do workspace com as permissões do Athena. Para ter mais informações, consulte [Modifique sua IAM função de espaço de trabalho para usar o conector de dados Athena](#).

- Familiarize-se com AWS IoT TwinMaker o sistema de componentes de entidade e como criar entidades. Para ter mais informações, consulte [Crie sua primeira entidade](#).
- Familiarize-se com os conectores AWS IoT TwinMaker de dados da. Para ter mais informações, consulte [AWS IoT TwinMaker conectores de dados](#).

## Como usar o conector de dados do Athena

Para usar o conector de dados do Athena, você deve criar um componente usando o conector do Athena como o tipo de componente. Em seguida, você anexa o componente a uma entidade em sua cena para uso em AWS IoT TwinMaker.

Crie um tipo de componente com o conector de dados do Athena

Use este procedimento para criar um tipo de AWS IoT TwinMaker componente com o conector de dados tabulares Athena:

1. Navegue até o [console do AWS IoT TwinMaker](#).
2. Abra um espaço de trabalho existente ou [crie um novo](#).
3. No menu de navegação do lado esquerdo, escolha Tipos de componentes e selecione Criar tipo de componente para abrir a página de criação do tipo de componente.
4. Na página Criar tipo de componente, preencha o campo ID com um ID que corresponda ao seu caso de uso.

**Component type information**

ID

com.test.athena.connector.example

Description

Example athena connector child component type

Must be less than 2048 characters

Base Type

Choose a pre-defined Component Type or create your own

com.amazon.athena.connector

5. Escolha o tipo de base. Na lista suspensa, selecione o conector de dados tabulares do Athena, rotulado como `com.amazon.athena.connector`.
6. Configure a fonte de dados do tipo de componente escolhendo recursos do Athena para os seguintes campos:
  - Escolha uma fonte de dados do Athena.
  - Escolha um banco de dados Athena.
  - Escolha um nome da tabela.
  - Escolha um grupo de trabalho do Athena.
7. Depois de escolher os recursos do Athena que deseja usar como fonte de dados, escolha quais colunas da tabela deseja incluir.
8. Selecione um nome de coluna de ID externo. Selecione uma coluna da etapa anterior para servir como coluna de ID externo. O ID externo é o ID usado para representar um ativo do Athena e mapeá-lo para uma AWS IoT TwinMaker entidade.

## 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

<input checked="" type="checkbox"/>	Table name	Data type
<input checked="" type="checkbox"/>	recordid	bigint
<input type="checkbox"/>	assetid	string
<input checked="" type="checkbox"/>	description	string
<input checked="" type="checkbox"/>	dateperformed	string
<input checked="" type="checkbox"/>	performedby	string
<input checked="" type="checkbox"/>	datevalidated	string
<input checked="" type="checkbox"/>	validatedby	string
<input checked="" type="checkbox"/>	comments	string
<input checked="" type="checkbox"/>	nextservicedate	string
<input checked="" type="checkbox"/>	servicerecordurl	string

### External ID Column

assetid

### Athena workgroup

Select an Athena workgroup

TestWorkgroup

9. (Opcional) Adicione AWS tags a esses recursos para que você possa agrupá-los e organizá-los.
10. Escolha Criar tipo de componente para concluir a criação do tipo de componente.

Crie um componente com o tipo de conector de dados do Athena e anexe-o a uma entidade

Use este procedimento para criar um AWS IoT TwinMaker componente com o conector de dados tabulares Athena e anexá-lo a uma entidade:

 Note

Você deve ter um tipo de componente existente que use o conector de dados tabulares do Athena como fonte de dados para concluir esse procedimento. Consulte o procedimento anterior Crie um tipo de componente com o conector de dados Athena antes de iniciar este passo a passo.

1. Navegue até o [console do AWS IoT TwinMaker](#).
2. Abra um espaço de trabalho existente ou [crie um novo](#).
3. No menu de navegação do lado esquerdo, escolha Entidades e selecione a entidade à qual você deseja adicionar o componente ou criar uma nova entidade.
4. [Crie uma nova entidade](#).
5. Em seguida, selecione Adicionar componente e preencha o campo Nome do componente com um nome que corresponda ao seu caso de uso.
6. No menu suspenso Tipo de componente, selecione o ID do tipo de componente que você criou no procedimento anterior.
7. Insira as informações do componente, um nome do componente e selecione o filho ComponentType criado anteriormente. Isso é o ComponentType que você criou com o conector de dados Athena.
8. Na seção Propriedades, insira o athenaComponentExternalID do componente.

Property	Data type	isTimeSeries	Storage	isRequired	Value
athenaComponentExt	String	False	Internal	True	A0001

Add another property

- Escolha Adicionar componente para concluir a criação do componente.

Agora você criou com sucesso um componente com o conector de dados do Athena como o tipo de componente e o associou a uma entidade.

## Como usar a referência JSON do conector de dados tabulares do Athena

O exemplo a seguir é a referência JSON completa para o conector de dados tabulares do Athena. Use isso como um recurso para criar conectores de dados e 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
      }
    }
  }
}
```

## Como usar o conector de dados do Athena

Você pode exibir suas entidades que estão usando tabelas do Athena no Grafana. Para obter mais informações, consulte [Integração do painel do Grafana AWS IoT TwinMaker](#).

Leia a [documentação do Athena](#) para obter informações sobre como criar e usar tabelas do Athena para armazenar dados.

## Solução de problemas do conector de dados do Athena

Este tópico aborda problemas comuns que você pode encontrar ao configurar o conector de dados Athena.

Localização do grupo de trabalho do Athena:

Ao criar o ComponentType do conector do Athena, um grupo de trabalho dele precisa ter a localização de saída configurada. Consulte [Como funcionam os grupos de trabalho](#).

## Permissões de perfil do IAM ausentes:

A função de espaço de trabalho AWS IoT TwinMaker; pode não ter a permissão de acesso à API Athena ao criar um ComponentType, adicionar um componente Ca a uma entidade ou executar a API. GetPropertyValue Para atualizar as permissões do IAM, consulte [Criar e gerenciar uma função de serviço para AWS IoT TwinMaker](#).

## Visualize dados tabulares do Athena no Grafana

Um plug-in Grafana também está disponível para visualizar seus dados tabulares no Grafana, um painel de controle com recursos adicionais, como classificação e filtragem com base nas propriedades selecionadas, sem fazer chamadas de API ou interações com o Athena. AWS IoT TwinMaker Este tópico mostra como configurar o Grafana para visualizar dados tabulares do Athena.

### Pré-requisitos

Antes de configurar um painel do Grafana para visualizar os dados tabulares do Athena, revise os seguintes pré-requisitos:

- Você configurou um ambiente no Grafana. Para mais informações, consulte [AWS IoT TwinMaker Integração ao Grafana](#).
- Você pode configurar uma fonte de dados do Grafana. Para obter mais informações, consulte [Grafana AWS IoT TwinMaker](#).
- Você está familiarizado com a criação de um novo painel e a adição de um novo painel.

## Visualize dados tabulares do Athena no Grafana

Este procedimento mostra como configurar um painel do Grafana para visualizar dados tabulares do Athena.

1. Abra seu painel da AWS IoT TwinMaker Grafana.
2. Selecione o painel Tabela nas configurações do painel.
3. Selecione sua fonte de dados na configuração da consulta.
4. Selecione a consulta Obter valor da propriedade.
5. Selecione uma entidade.

6. Selecione um componente que tenha um ComponentType que estenda o tipo de componente básico do Athena.
7. Selecione o grupo de propriedades da sua tabela do Athena.
8. Selecione qualquer número de propriedades do grupo de propriedades.
9. Configure as condições tabulares por meio de uma lista de filtros e ordens de propriedades. Com as seguintes opções:
  - Filtro: defina uma expressão para o valor de uma propriedade para filtrar seus dados.
  - OrderBy: especifique se os dados devem ser retornados em ordem crescente ou decrescente para uma propriedade.

The screenshot displays a query configuration interface in the AWS IoT TwinMaker console. At the top, a table titled "Panel Title" shows query results with columns: crit, description, equipment\_type, status, total, and won. Below the table, the query configuration is shown with the following settings:

- Query type:** Get Property value
- Entity:** TabularEntity1
- Component Name:** TabularComponent
- Property Group:** tabularPropertyGroup (TABULAR)
- Selected Properties:** won (INTEGER), status (STRING), total (INTEGER), crit (INTEGER), description (STRING), equipment\_type (STRING)
- Filter:** crit (INTEGER) = 5
- OrderBy:** total (INTEGER) DESC

# Desenvolvendo AWS IoT TwinMaker conectores de dados de séries temporais

Esta seção explica como desenvolver um conector de dados de série temporal em um step-by-step processo. Além disso, apresentamos um exemplo de conector de dados de série temporal baseado em todo o exemplo da fábrica de cookies, que inclui modelos 3D, entidades, componentes, alarmes e conectores. A fonte de amostra da fábrica de cookies está disponível no [GitHub repositório de AWS IoT TwinMaker amostras](#).

## Tópicos

- [AWS IoT TwinMaker pré-requisitos do conector de dados de série temporal](#)
- [Plano de fundo do conector de dados de séries temporais](#)
- [Como desenvolver um conector de dados de séries temporais](#)
- [Como melhorar seu conector de dados](#)
- [Como testar seu conector](#)
- [Segurança](#)
- [Criação de AWS IoT TwinMaker recursos](#)
- [Próximas etapas](#)
- [Exemplo de conector AWS IoT TwinMaker de série temporal de fábrica de cookies](#)

## AWS IoT TwinMaker pré-requisitos do conector de dados de série temporal

Antes de desenvolver seu conector de dados de séries temporais, recomendamos que você conclua as seguintes tarefas:

- Crie um [espaço de trabalho AWS IoT TwinMaker](#).
- Crie [tipos de componentes AWS IoT TwinMaker](#).
- Crie [entidades AWS IoT TwinMaker](#).
- (Opcional) Leia [Uso e criação de tipos de componentes](#).
- (Opcional) Leia a [AWS IoT TwinMaker interface do conector de dados](#) para obter uma compreensão geral dos conectores de dados AWS IoT TwinMaker .

**Note**

Para ver um exemplo de um conector totalmente implementado, consulte nosso exemplo de implementação na fábrica de cookies.

## Plano de fundo do conector de dados de séries temporais

Imagine que você está trabalhando com uma fábrica que tem um conjunto de mixers de cookies e um tanque de água. Você gostaria de criar gêmeos AWS IoT TwinMaker digitais dessas entidades físicas para poder monitorar seus estados operacionais verificando várias métricas de séries temporais.

Você tem sensores no local configurados e já está transmitindo dados de medição para um banco de dados Timestream. Você quer ser capaz de visualizar e organizar os dados de medição em AWS IoT TwinMaker com o mínimo de sobrecarga. É possível realizar essa tarefa usando um conector de dados de séries temporais. A imagem a seguir exibe um exemplo de tabela de telemetria, que é preenchida por meio do uso de um conector de série temporal.

TelemetryAssetId	TelemetryAssetType	measure_name	time	measure_value:varchar	measure_value:double
Mixer_22_680b5b8e-1afe-4a77-87ab-834f8e5ba01e	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-834f8e5ba01e	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_c142effc-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-d6f3-4804-9fe1-7d2abca8d2d0	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	-	60.20658770752
Mixer_6_b66db3d3-c144-47b5-afb9-3a0150c53456	Mixer	RPM	2022-04-19 00:28:00.241000000	-	60.20658770752

[Os conjuntos de dados e a tabela Timestream usados nesta captura de tela estão disponíveis no repositório de amostras.AWS IoT TwinMaker GitHub](#) Consulte também o [exemplo de conector de implementação da fábrica de cookies](#), que produz o resultado mostrado na captura de tela anterior.

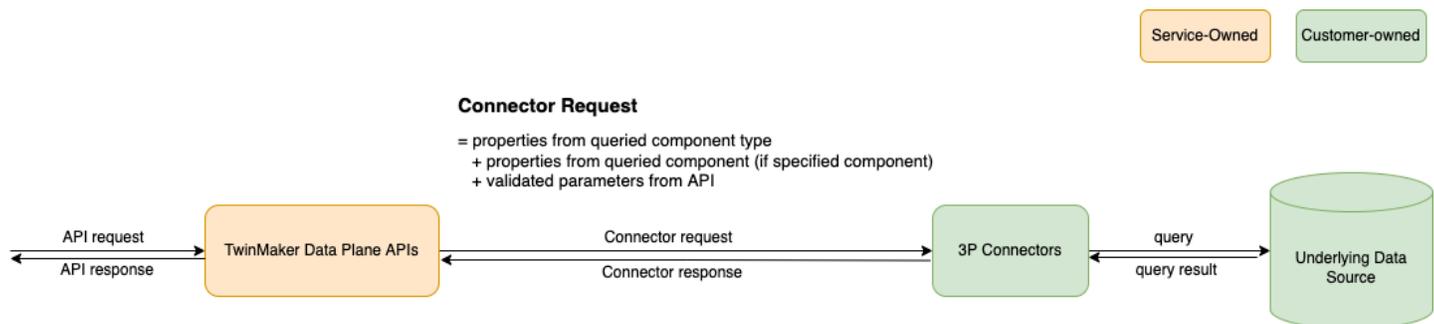
## Fluxo de dados do conector de dados de séries temporais

Para consultas de plano de dados, AWS IoT TwinMaker busca as propriedades correspondentes dos componentes e dos tipos de componentes a partir das definições de componentes e tipos de componentes. AWS IoT TwinMaker encaminha propriedades para AWS Lambda funções junto com qualquer parâmetro de consulta de API na consulta.

AWS IoT TwinMaker usa funções Lambda para acessar e resolver consultas de fontes de dados e retornar os resultados dessas consultas. As funções do Lambda usam as propriedades do componente e do tipo de componente do plano de dados para resolver a solicitação inicial.

Os resultados da consulta do Lambda são mapeados para uma resposta da API e retornados para você.

AWS IoT TwinMaker define a interface do conector de dados e a usa para interagir com as funções do Lambda. Usando conectores de dados, você pode consultar sua fonte de dados a partir da API AWS IoT TwinMaker sem nenhum esforço de migração de dados. A imagem a seguir descreve o fluxo de dados básico descrito nos parágrafos anteriores.



## Como desenvolver um conector de dados de séries temporais

O procedimento a seguir descreve um modelo de desenvolvimento que se transforma incrementalmente em um conector funcional de dados de séries temporais. As etapas básicas são:

### 1. Crie um tipo de componente básico válido

Em um tipo de componente, você define propriedades comuns que são compartilhadas entre seus componentes. Para saber mais sobre como definir tipos de componentes, consulte [Uso e criação de tipos de componentes](#).

AWS IoT TwinMaker usa um [padrão de modelagem entidade-componente](#) para que cada componente seja anexado a uma entidade. Recomendamos que você modele cada item físico como uma entidade e modele fontes de dados diferentes com seus próprios tipos de componentes.

O exemplo a seguir mostra um modelo de componente do Timestream do tipo de componente com uma propriedade:

```
{"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
  }
}
}
```

Os principais elementos do tipo de componente são os seguintes:

- A `telemetryId` propriedade identifica a chave exclusiva do item físico na fonte de dados correspondente. O conector de dados usa essa propriedade como condição de filtro para consultar somente valores associados a um determinado item. Além disso, se você incluir o valor da propriedade `telemetryId` na resposta da API do plano de dados, o lado do cliente obterá o ID e poderá realizar uma pesquisa inversa, se necessário.

- O campo `lambdaArn` identifica a função do Lambda com a qual o tipo de componente se envolve.
- O sinalizador `isRequiredInEntity` impõe a criação do ID. Esse sinalizador é necessário para que, quando o componente for criado, o ID do item também seja instanciado.
- O `TelemetryId` é adicionado ao tipo de componente como uma identificação externa para que o item possa ser identificado na tabela Timestream.

## 2. Crie um componente com o tipo de componente

Para usar o tipo de componente que você criou, você deve criar um componente e anexá-lo à entidade da qual deseja recuperar dados. As etapas a seguir detalham o processo de criação desse componente:

- a. Navegue até o [console do AWS IoT TwinMaker](#).
- b. Selecione e abra o mesmo espaço de trabalho no qual você criou os tipos de componentes.
- c. Navegue até a página da entidade.
- d. Crie uma nova entidade ou selecione uma entidade existente na tabela.
- e. Depois de escolher a entidade que você deseja usar, selecione Adicionar componente para abrir a página Adicionar componente.
- f. Dê um nome ao componente e, para o Tipo, selecione o tipo de componente que você criou com o modelo em 1. Crie um tipo de componente básico válido.

## 3. Faça com que seu tipo de componente chame um conector Lambda

O conector Lambda precisa acessar a fonte de dados e gerar a instrução de consulta com base na entrada e encaminhá-la para a fonte de dados. O exemplo a seguir mostra um modelo de solicitação JSON que faz isso.

```
{
  "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
    }
  }
}
```

Os principais elementos da solicitação:

- `selectedProperties` é uma lista que você preenche com as propriedades para as quais deseja medições de Timestream.
- Os campos `startDateTime`, `startTime`, `endDateTime` e `endTime` especificam um intervalo de tempo para a solicitação. Isso determina o intervalo de amostras para as medições retornadas.
- `entityId` é o nome da entidade da qual você está consultando dados.
- `componentName` é o nome do componente do qual você está consultando dados.
- Use o campo `orderByTime` para organizar a ordem na qual os resultados são exibidos.

Na solicitação de exemplo anterior, esperaríamos obter uma série de amostras das propriedades selecionadas durante a janela de tempo determinada para o item em questão, com a ordem de tempo selecionada. A declaração de resposta pode ser resumida da seguinte forma:

```
{
  "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. Atualize seu tipo de componente para ter duas propriedades

O modelo JSON a seguir mostra um tipo de componente válido com duas propriedades:

```
{
  "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. Atualize o conector Lambda para lidar com a segunda propriedade

A API do plano de AWS IoT TwinMaker dados suporta a consulta de várias propriedades em uma única solicitação e AWS IoT TwinMaker segue uma única solicitação a um conector fornecendo uma lista `deselectedProperties`.

A solicitação JSON a seguir mostra um modelo modificado que agora suporta uma solicitação para duas propriedades.

```

{
  "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
    }
  }
}
```

```
}
```

Da mesma forma, a resposta correspondente também é atualizada, conforme mostrado no exemplo a seguir:

```
{
  "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

Em termos de paginação nesse caso, o tamanho da página na solicitação aplica-se a todas as propriedades. Isso significa que, com 5 propriedades na consulta e um tamanho de página de 100, se houver pontos de dados suficientes na fonte, você deve esperar ver 100 pontos de dados por propriedade, com 500 pontos de dados no total.

Para ver um exemplo de implementação, consulte o [exemplo do conector Snowflake](#) em GitHub

## Como melhorar seu conector de dados

### Tratamento de exceções

É seguro para o conector Lambda lançar exceções. Na chamada da API do plano de dados, o AWS IoT TwinMaker serviço espera que a função Lambda retorne uma resposta. Se a implementação do conector gerar uma exceção, AWS IoT TwinMaker traduz o tipo de exceção como

sendoConnectorFailure, fazendo com que o cliente da API saiba que ocorreu um problema dentro do conector.

## Tratamento de paginação

No exemplo, o Timestream fornece uma [função do utilitário](#) que pode ajudar a suportar a paginação de forma nativa. No entanto, para algumas outras interfaces de consulta, como SQL, talvez seja necessário um esforço extra para implementar um algoritmo de paginação eficiente. Há um exemplo de conector [Snowflake](#) que manipula a paginação em uma interface SQL.

Quando o novo token é devolvido AWS IoT TwinMaker por meio da interface de resposta do conector, o token é criptografado antes de ser devolvido ao cliente da API. Quando o token é incluído em outra solicitação, ele é AWS IoT TwinMaker descriptografado antes de encaminhá-lo para o conector Lambda. Recomendamos evitar adicionar informações confidenciais ao token.

## Como testar seu conector

Embora você ainda possa atualizar a implementação depois de vincular o conector ao tipo de componente, é altamente recomendável verificar o conector Lambda antes de fazer a integração com AWS IoT TwinMaker.

Há várias maneiras de testar seu conector Lambda: você pode testar o conector Lambda no console Lambda ou localmente no AWS CDK.

[Para obter mais informações sobre como testar suas funções do Lambda, consulte Testando funções do Lambda e Testando aplicativos localmente. AWS CDK](#)

## Segurança

Para obter documentação sobre as melhores práticas de segurança com o Timestream, consulte [Segurança no Timestream](#).

Para ver um exemplo de prevenção de injeção de SQL, consulte o [script Python](#) a seguir no AWS IoT TwinMaker Samples GitHub Repository.

## Criação de AWS IoT TwinMaker recursos

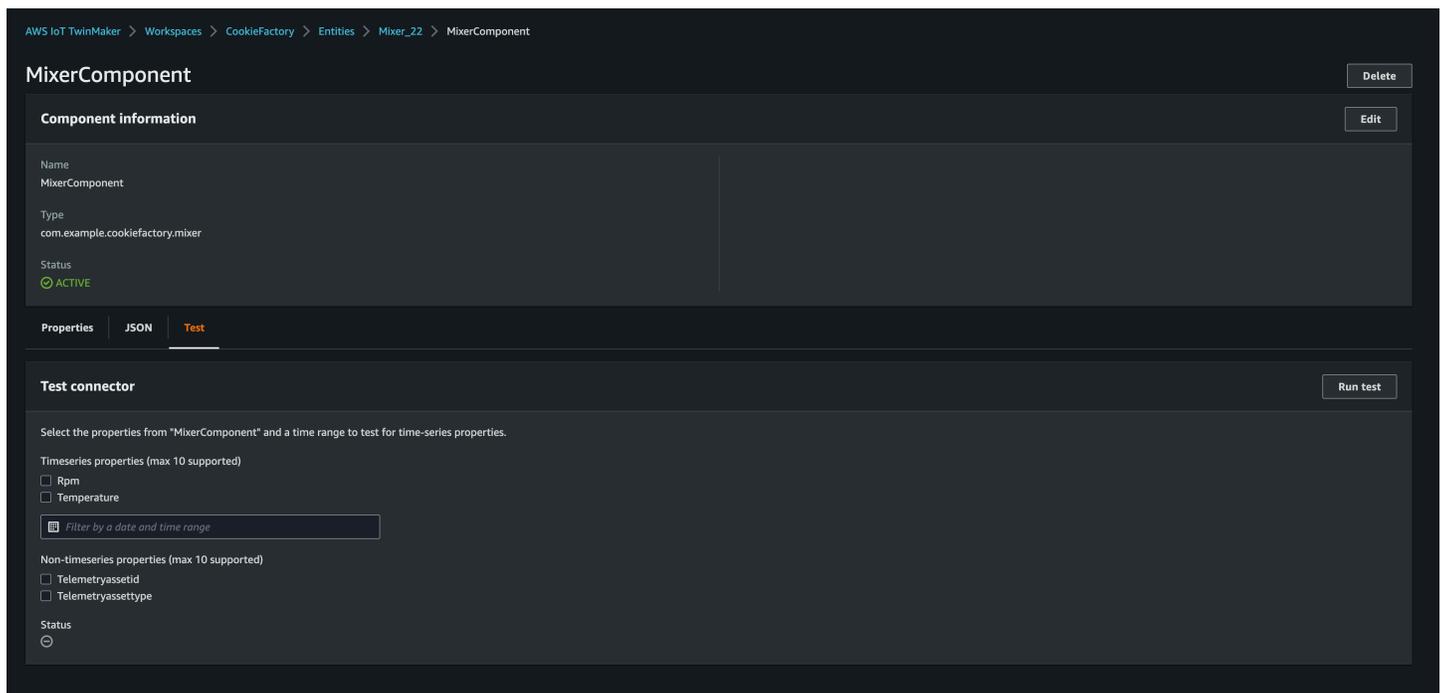
Depois de implementar a função Lambda, você pode criar AWS IoT TwinMaker recursos como tipos de componentes, entidades e componentes por meio do [AWS IoT TwinMaker console ou da API](#).

### Note

Se você seguir as instruções de configuração na GitHub amostra, todos os AWS IoT TwinMaker recursos estarão disponíveis automaticamente. Você pode verificar as definições do tipo de componente na [AWS IoT TwinMaker GitHub amostra](#). Depois que o tipo de componente é usado por qualquer componente, as definições de propriedades e funções do tipo de componente não podem ser atualizadas.

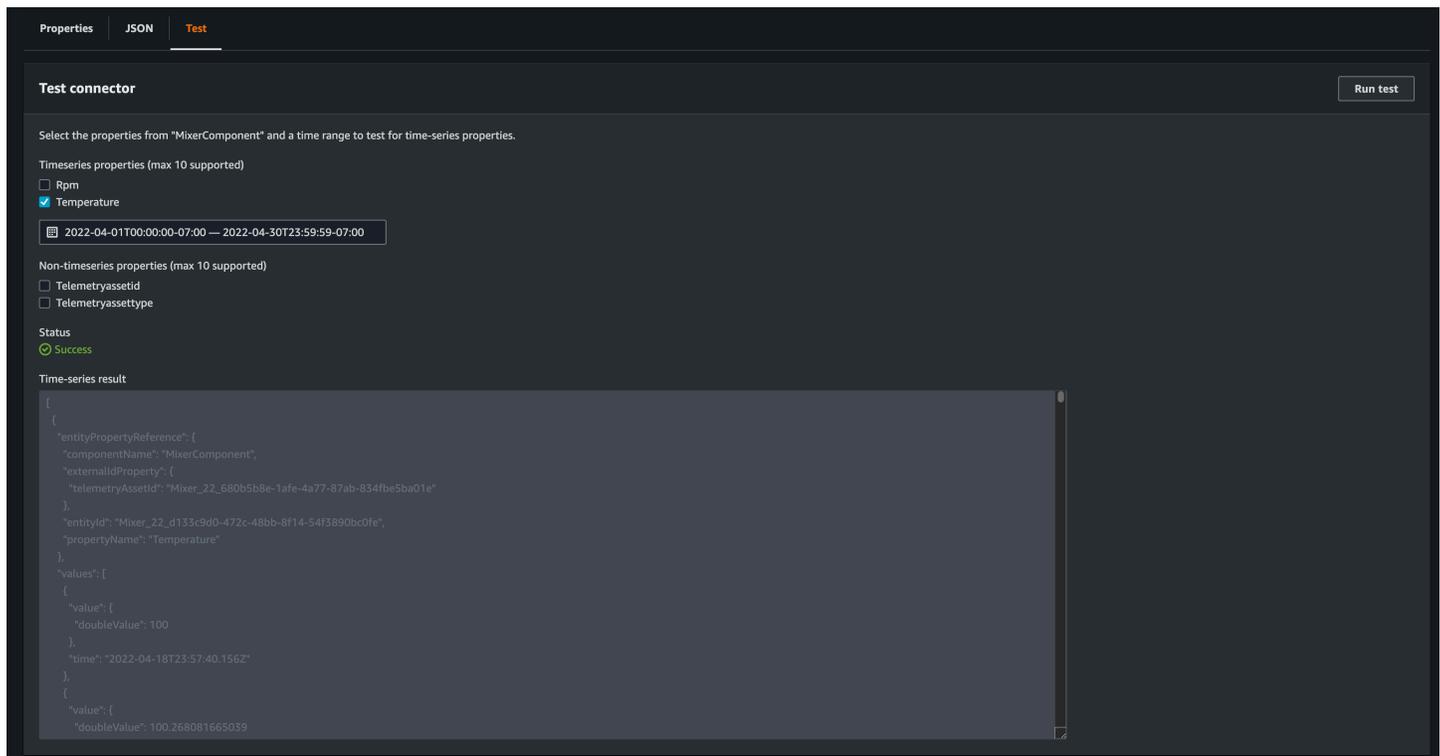
## Teste de integração

Recomendamos fazer um teste integrado AWS IoT TwinMaker para verificar se a consulta do plano de dados funciona end-to-end. Você pode fazer isso por meio [GetPropertyValueHistory](#) da API ou facilmente no [AWS IoT TwinMaker console](#).



The screenshot shows the AWS IoT TwinMaker console interface for a component named 'MixerComponent'. The breadcrumb trail at the top reads: 'AWS IoT TwinMaker > Workspaces > CookieFactory > Entities > Mixer\_22 > MixerComponent'. The main header is 'MixerComponent' with 'Delete' and 'Edit' buttons. Below this is the 'Component information' section, which includes: Name: MixerComponent; Type: com.example.cookiefactory.mixer; Status: ACTIVE (indicated by a green circle). There are tabs for 'Properties', 'JSON', and 'Test', with 'Test' being the active tab. The 'Test connector' section has a 'Run test' button and instructions: 'Select the properties from "MixerComponent" and a time range to test for time-series properties.' It lists 'Timeseries properties (max 10 supported)' with checkboxes for 'Rpm' and 'Temperature', and a 'Filter by a date and time range' input field. Below that, it lists 'Non-timeseries properties (max 10 supported)' with checkboxes for 'Telemetryassetid' and 'Telemetryassettype', and a 'Status' dropdown menu.

No AWS IoT TwinMaker console, acesse os detalhes do componente e, em Teste, você verá que todas as propriedades do componente estão listadas lá. A área de teste do console permite testar propriedades e propriedades de séries temporais. Para propriedades de séries temporais, você também pode usar a [GetPropertyValueHistory](#) API e, para non-time-series propriedades, usar a [GetPropertyValue](#) API. Se o seu conector Lambda suportar várias consultas de propriedades, você poderá escolher mais de uma propriedade.



## Próximas etapas

Agora você pode configurar um [painel do Grafana AWS IoT TwinMaker](#) para visualizar as métricas. Você também pode explorar outras amostras de conectores de dados no [GitHub repositório de AWS IoT TwinMaker amostras](#) para ver se elas se adequam ao seu caso de uso.

## Exemplo de conector AWS IoT TwinMaker de série temporal de fábrica de cookies

O [código completo da função Lambda da fábrica de cookies](#) está disponível em GitHub. Embora você ainda possa atualizar a implementação depois de vincular o conector ao tipo de componente, é altamente recomendável verificar o conector Lambda antes de fazer a integração com AWS IoT TwinMaker. Você pode testar sua função do Lambda no console do Lambda ou localmente no AWS CDK. [Para obter mais informações sobre como testar suas funções do Lambda, consulte Testando funções do Lambda e Testando aplicativos localmente. AWS CDK](#)

## Exemplos de tipos de componentes de fábrica de cookies

Em um tipo de componente, definimos propriedades comuns que são compartilhadas entre os componentes. Para o exemplo da fábrica de cookies, componentes físicos do mesmo tipo

compartilham as mesmas medidas, para que possamos definir o esquema de medidas no tipo de componente. Como exemplo, o tipo de mixer é definido no exemplo a seguir.

```
{
  "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 exemplo, um componente físico pode ter medições em um banco de dados Timestream, registros de manutenção em um banco de dados SQL ou dados de alarme em sistemas de alarme. Criar vários componentes e associá-los a uma entidade vincula diferentes fontes de dados à entidade e preenche o gráfico de componente da entidade. Nesse contexto, cada componente precisa de uma `telemetryId` propriedade para identificar a chave exclusiva do componente na fonte de dados correspondente. Especificar a `telemetryId` propriedade tem dois benefícios: a propriedade pode ser usada no conector de dados como uma condição de filtro para consultar somente os valores do componente especificado e, se você incluir o valor da `telemetryId` propriedade na resposta da API do plano de dados, o lado do cliente obterá o ID e poderá realizar uma pesquisa inversa, se necessário.

Se você adicionar o `TelemetryId` ao tipo de componente como uma identificação externa, ele identificará o componente na `TimeStream` tabela.

```
{
  "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
    }
}
}
}

```

Da mesma forma, temos o tipo de componente do WaterTank, conforme mostrado no exemplo JSON a seguir.

```

{
  "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
}
}
}
```

O `TelemetryType` é uma propriedade opcional no tipo de componente se for destinado a consultar valores de propriedades no escopo da entidade. Para ver um exemplo, consulte os tipos de componentes definidos no [GitHub repositório de AWS IoT TwinMaker amostras](#). Existem tipos de alarme também incorporados na mesma tabela, portanto, `TelemetryType` é definido e você extrai propriedades comuns como `TelemetryId` e `TelemetryType` para um tipo de componente primário para outros tipos secundários compartilharem.

## Exemplo de Lambda

O conector Lambda precisa acessar a fonte de dados e gerar a instrução de consulta com base na entrada e encaminhá-la para a fonte de dados. Um exemplo de solicitação enviada para o Lambda é mostrado no exemplo de JSON a seguir.

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

```
'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'  
}
```

O objetivo da função Lambda é consultar dados históricos de medição de uma determinada entidade. AWS IoT TwinMaker fornece um mapa de propriedades do componente e você deve especificar um valor instanciado para o ID do componente. Por exemplo, para lidar com a consulta em nível de tipo de componente (que é comum em casos de uso de alarmes) e retornar o status do alarme de todos os componentes no espaço de trabalho, o mapa de propriedades tem definições de propriedades do tipo de componente.

Para o caso mais simples, como na solicitação anterior, queremos uma série de amostras de temperatura durante a janela de tempo determinada para o componente em questão, em ordem crescente. A declaração de consulta pode ser resumida da seguinte forma:

```
...  
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}  
...
```

# Criação e edição de AWS IoT TwinMaker cenas

As cenas são visualizações tridimensionais do seu gêmeo digital. São a principal forma de você editar seu gêmeo digital. Aprenda a adicionar alarmes, dados de séries temporais, sobreposições de cores, tags e regras visuais à sua cena para alinhar as visualizações de seu gêmeo digital com seu caso de uso no mundo real.

Esta seção abrange os seguintes tópicos:

- [Antes de criar sua primeira cena](#)
- [Carregar recursos para a Biblioteca AWS IoT TwinMaker de Recursos](#)
- [Crie suas cenas](#)
- [Adicione câmeras fixas às entidades](#)
- [Edição avançada do cenário](#)
- [Edite suas cenas](#)
- [Formato de modelo 3D Tiles](#)
- [Cenas dinâmicas](#)

## Antes de criar sua primeira cena

As cenas dependem de recursos para representar seu gêmeo digital. Esses recursos são compostos por modelos 3D, dados ou arquivos de textura. O tamanho e a complexidade dos recursos, dos elementos da cena, como iluminação, e do hardware do computador afetam o desempenho das cenas AWS IoT TwinMaker. Use as informações deste tópico para reduzir o atraso, os tempos de carregamento e melhorar a taxa de quadros de suas cenas.

## Otimize seus recursos antes de importá-los AWS IoT TwinMaker

Você pode usar AWS IoT TwinMaker para interagir com seu gêmeo digital em tempo real. Para obter a melhor experiência com suas cenas, recomendamos otimizar seus recursos para uso em um ambiente em tempo real.

Seus modelos 3D podem ter um impacto significativo na performance. A geometria e as malhas complexas do modelo podem reduzir o desempenho. Por exemplo, modelos CAD industriais têm um alto nível de detalhes. Recomendamos compactar as malhas desses modelos e reduzir sua

contagem de polígonos antes de usá-las em cenas. AWS IoT TwinMaker Se você estiver criando novos modelos 3D para AWS IoT TwinMaker, deve estabelecer um nível de detalhe e mantê-lo em todos os seus modelos. Remova detalhes dos modelos que não afetam a visualização ou a interpretação do seu caso de uso.

Para compactar modelos e reduzir o tamanho do arquivo, use ferramentas de compactação de malha de código aberto, como a [compactação de dados 3D DRACO](#).

Texturas não otimizadas também podem afetar o desempenho. Caso não precise de transparência em suas texturas, considere escolher o formato de imagem PEG em vez do formato PNG. É possível compactar seus arquivos de textura usando ferramentas de compactação de textura de código aberto, como a [compactação de textura Basis Universal](#).

## Melhores práticas para desempenho em AWS IoT TwinMaker

Para obter o melhor desempenho com AWS IoT TwinMaker, observe as seguintes limitações e práticas recomendadas.

- AWS IoT TwinMaker o desempenho da renderização de cena depende do hardware. O desempenho varia nas diferentes configurações de hardware do computador.
- Recomendamos uma contagem total de polígonos de menos de 1 milhão em todos os objetos do seu AWS IoT TwinMaker.
- Recomendamos um total de 200 objetos por cena. Aumentar o número de objetos em uma cena além de 200 pode diminuir a taxa de quadros da cena.
- Recomendamos que o tamanho total de todos os ativos 3D exclusivos em sua cena não exceda 100 megabytes. Caso contrário, você poderá encontrar tempos de carregamento lentos ou desempenho reduzido, dependendo do navegador e do hardware.
- As cenas têm iluminação ambiente por padrão. Você pode adicionar luzes extras em uma cena para focar certos objetos ou projetar sombras nos objetos. Recomendamos usar uma luz por cena. Use luzes quando necessário e evite replicar luzes do mundo real em uma cena.

## Saiba mais

Use esses recursos para saber mais sobre técnicas de otimização que podem ser usadas para melhorar o desempenho em suas cenas.

- [Como converter e compactar modelos OBJ em GLTF para uso com AWS IoT TwinMaker](#)

- [Otimize seus modelos 3D para conteúdo da web](#)
- [Como otimizar cenas para um melhor desempenho do WebGL](#)

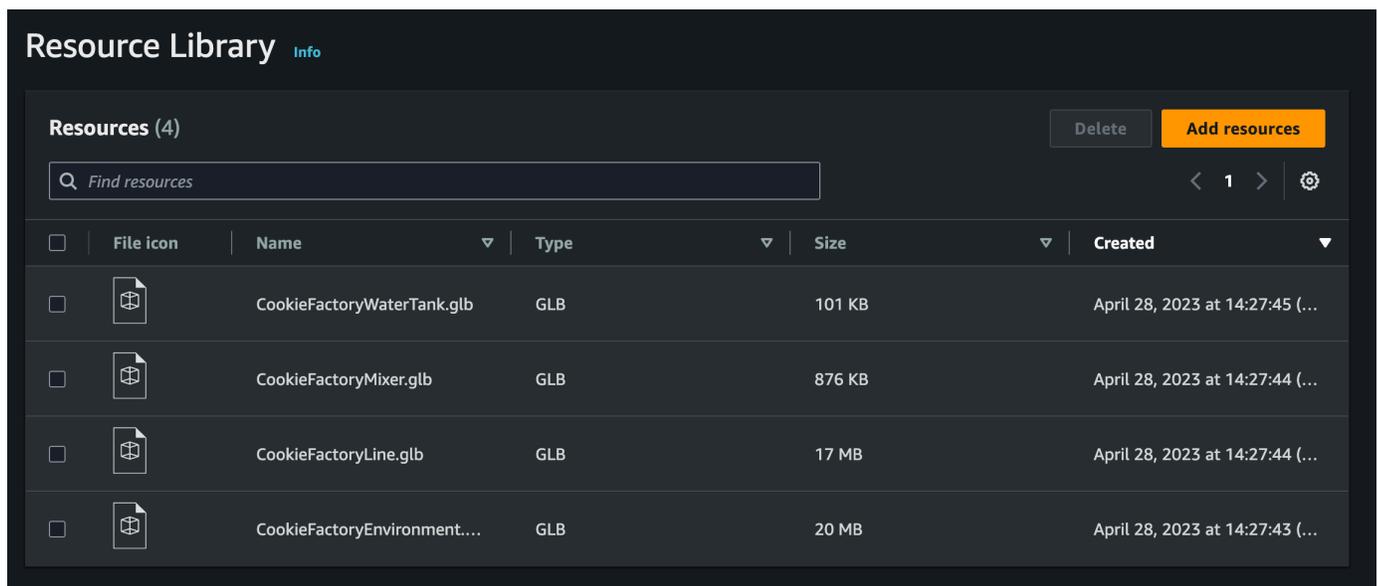
## Carregar recursos para a Biblioteca AWS IoT TwinMaker de Recursos

Você pode usar a biblioteca de recursos para controlar e gerenciar qualquer recurso que você queira colocar em cenas para sua aplicação de gêmeo digital. Para AWS IoT TwinMaker conhecer os recursos, carregue-os usando a página do console da Biblioteca de Recursos.

### Carregar arquivos para a Biblioteca de Recursos usando o console

Siga estas etapas para adicionar arquivos à Biblioteca de Recursos usando o AWS IoT TwinMaker console.

1. No menu de navegação à esquerda, em Espaços de trabalho, selecione Biblioteca de recursos.
2. Selecione Adicionar recursos e escolha os arquivos que você deseja carregar.



## Crie suas cenas

Nesta seção, você configurará uma cena para poder editar seu gêmeo digital. Você pode importar um modelo 3D que foi carregado para a [biblioteca de recursos](#) e, em seguida, adicionar widgets e vincular dados de propriedades a objetos para completar seu gêmeo digital. Os objetos de cena

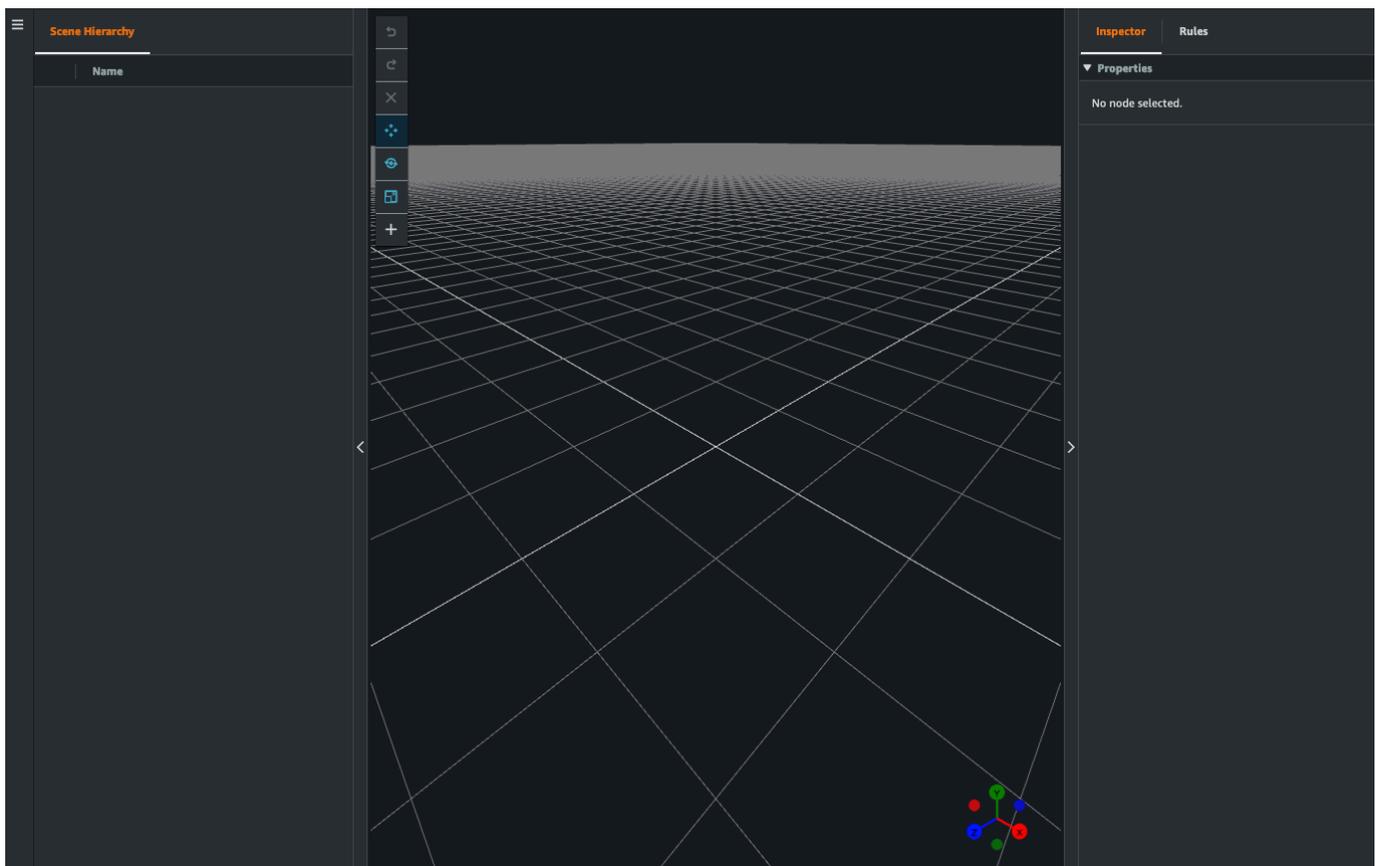
podem incluir um edifício ou espaço inteiro, ou peças individuais de equipamento posicionadas em sua localização física.

**Note**

Antes de criar uma cena, você deve criar um espaço de trabalho.

Use o procedimento a seguir para criar sua cena em AWS IoT TwinMaker.

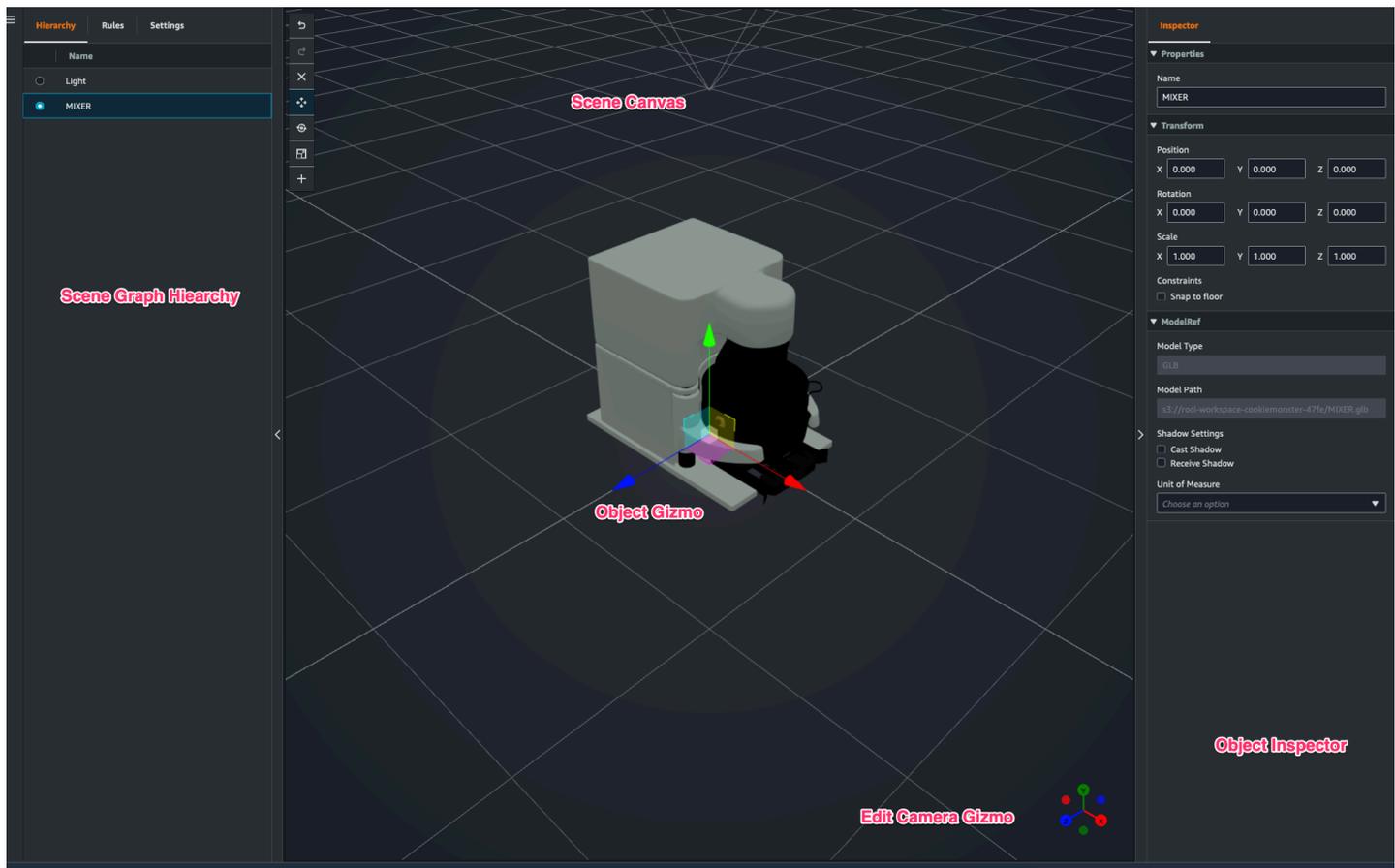
1. Para abrir o painel de cena, na navegação à esquerda do seu espaço de trabalho, escolha Cenas.
2. Escolha Create scene (Criar cena). O novo painel de criação de cena é aberto.
3. No painel de criação de cena, insira um nome e uma descrição para a nova cena. Se você tiver um plano de preços de pacote padrão ou em camadas, poderá selecionar o tipo de cena. É recomendável usar uma [cena dinâmica](#).
4. Quando estiver pronto para criar a cena, selecione Criar cena. A nova cena será exibida e está pronta para edição.



## Use a navegação 3D AWS IoT TwinMaker em suas cenas

A AWS IoT TwinMaker cena tem um conjunto de controles de navegação que você pode usar para navegar com eficiência pelo espaço 3D da cena. Para interagir com o espaço 3D e os objetos representados pela sua cena, você usa os widgets e opções de menu a seguir.

- **Inspetor:** use a janela Inspetor para visualizar e editar propriedades e configurações de uma entidade ou componente selecionado em sua hierarquia.
- **Scene Canvas:** o Scene Canvas é o espaço 3D no qual você pode posicionar e orientar quaisquer recursos 3D que desejar usar.
- **Hierarquia do gráfico de cena:** é possível usar este painel para ver todas as entidades presentes em sua cena. Ele aparece no lado esquerdo da janela.
- **Dispositivo de objeto:** use este dispositivo para mover objetos pela tela. Ele aparece no centro de um objeto 3D selecionado em Scene Canvas.
- **Editar dispositivo da câmera:** use o dispositivo Editar câmera para visualizar rapidamente a orientação atual da câmera de visualização da cena e modificar o ângulo de visão. Você pode encontrar esse dispositivo no canto inferior direito da visualização da cena.
- **Controles de zoom:** para navegar em Scene Canvas, clique com o botão direito do mouse e arraste na direção que você deseja mover. Para girar, clique com o botão esquerdo e arraste para girar. Para ampliar, use a roda de rolagem do mouse ou aperte e afaste os dedos no touchpad do laptop.



Os botões de cena no painel de hierarquia têm as seguintes funções listadas, na ordem do layout dos botões:

- Desfazer: desfaz sua última alteração na cena.
- Refazer: refaz sua última alteração na cena.
- Mais (+): usar este botão para obter acesso às seguintes ações: Adicionar nó vazio, Adicionar modelo 3D, Adicionar tag, Adicionar luz e Adicionar sombreador de modelo.
- Altere o método de navegação: tenha acesso às opções de navegação da câmera de cena, Orbit e Pan.
- Lixeira (excluir): use este botão para excluir um objeto selecionado em sua cena.
- Ferramentas de manipulação de objetos: use este botão para traduzir, girar e escalar o objeto selecionado.

## Adicione câmeras fixas às entidades

Você pode anexar visualizações de câmeras fixas às suas entidades dentro de suas AWS IoT TwinMaker cenas. Essas câmeras fornecem uma perspectiva fixa em um modelo 3D, permitindo que você mude rápida e facilmente sua perspectiva em uma cena para uma entidade-alvo.

1. Navegue até sua cena no [console do AWS IoT TwinMaker](#).
2. No menu de hierarquia da cena, selecione a entidade à qual você deseja conectar a câmera.
3. Pressione o botão + e, nas opções suspensas, selecione Adicionar câmera da visualização atual. Para aplicar uma câmera com a perspectiva atual à entidade.
4. No inspetor, você pode configurar sua câmera e ajustar as seguintes configurações:
  - Nome de uma câmera
  - A posição e rotação da câmera
  - A distância focal da câmera
  - O nível de zoom
  - Planos de recorte próximos e distantes
5. Para acessar sua câmera depois de posicioná-la. Selecione a entidade à qual você adicionou a câmera na hierarquia. Procure o nome da câmera listado abaixo da entidade.
6. Depois de selecionar a câmera posicionada na sua entidade, a visualização da câmera da cena se ajustará à perspectiva definida da câmera posicionada.

## Edição avançada do cenário

AWS IoT TwinMaker as cenas apresentam um conjunto de ferramentas para aprimoramento, edição e manipulação dos recursos presentes em sua cena.

Os tópicos a seguir ensinam como usar os recursos de edição aprimorados em suas AWS IoT TwinMaker cenas.

- [Posicionamento direcionado dos objetos da cena](#)
- [Seleção de submodelos](#)
- [Editar entidades na hierarquia da cena](#)

## Posicionamento direcionado dos objetos da cena

AWS IoT TwinMaker permite que você posicione e adicione objetos com precisão à sua cena. Este atributo de edição aprimorado oferece maior controle de onde você posiciona tags, luzes de entidades e modelos em sua cena.

1. Navegue até sua cena no [console do AWS IoT TwinMaker](#).
2. Pressione o botão + e selecione uma das opções nas opções suspensas. Pode ser um modelo, uma luz, uma tag ou qualquer item do menu +.

Ao mover o cursor no espaço 3D da cena, você deve ver um alvo ao redor do cursor.

3. Use o alvo para posicionar com precisão os elementos em sua cena.

## Seleção de submodelos

AWS IoT TwinMaker permite selecionar submodelos de modelos 3D em cenas e aplicar propriedades padrão a elas, como etiquetas, luzes ou regras.

Os formatos de arquivo do modelo 3D contêm metadados que podem especificar subáreas do modelo como submodelos dentro do modelo maior. Por exemplo, um modelo pode ser um sistema de filtragem; partes individuais do sistema, como tanques, canos ou um motor, são marcadas como submodelos do modelo 3D da filtragem.

Formatos de arquivo 3D compatíveis em cenas: GLB e GLTF.

1. Navegue até sua cena no [console do AWS IoT TwinMaker](#).
2. Se você não tiver modelos em sua cena, certifique-se de adicionar um selecionando a opção no menu +.
3. Selecione o modelo listado em sua hierarquia de cena. Uma vez selecionado, a hierarquia deve exibir todos os submodelos abaixo do modelo.

### Note

Caso não encontre nenhum submodelo listado, é provável que o modelo não tenha sido configurado para ter um submodelo.

4. Para alternar a visibilidade de um submodelo, pressione o ícone de olho, localizado à direita do nome do submodelo na hierarquia.

5. Para editar dados do submodelo, como nome ou posição, o inspetor de cena será exibido quando um submodelo for selecionado. Use o menu do inspetor para atualizar ou alterar os dados do submodelo.
6. Para adicionar tags, luzes, regras ou outras propriedades aos submodelos, pressione o botão + enquanto o submodelo é selecionado na hierarquia.

## Editar entidades na hierarquia da cena

AWS IoT TwinMaker as cenas permitem que você edite diretamente as propriedades das entidades dentro da tabela hierárquica. O procedimento a seguir mostra quais ações você pode executar em uma entidade por meio do menu de hierarquia.

1. Navegue até sua cena no [console do AWS IoT TwinMaker](#).
2. Abra a hierarquia da cena e selecione um subelemento de uma entidade que você deseja manipular.
3. Depois que o elemento for selecionado, pressione o botão + e, no menu suspenso, selecione uma das opções:
  - Adicionar nó vazio
  - Adicionar modelo 3D
  - Adicionar luz
  - Adicionar câmera da visualização atual
  - Adicionar tag
  - Adicionar sombreador de modelo
  - Adicionar indicador de movimento
4. Depois de selecionar uma das opções no menu suspenso, a seleção será aplicada à cena como elemento secundário selecionado na etapa 2.
5. Você pode reordenar elementos secundários e elementos primários selecionando um elemento secundário e arrastando na hierarquia até um novo elemento primário.

## Adicione anotações às entidades

O compositor de AWS IoT TwinMaker cenas permite que você anote qualquer elemento em sua hierarquia de cenas. A anotação é criada em markdown.

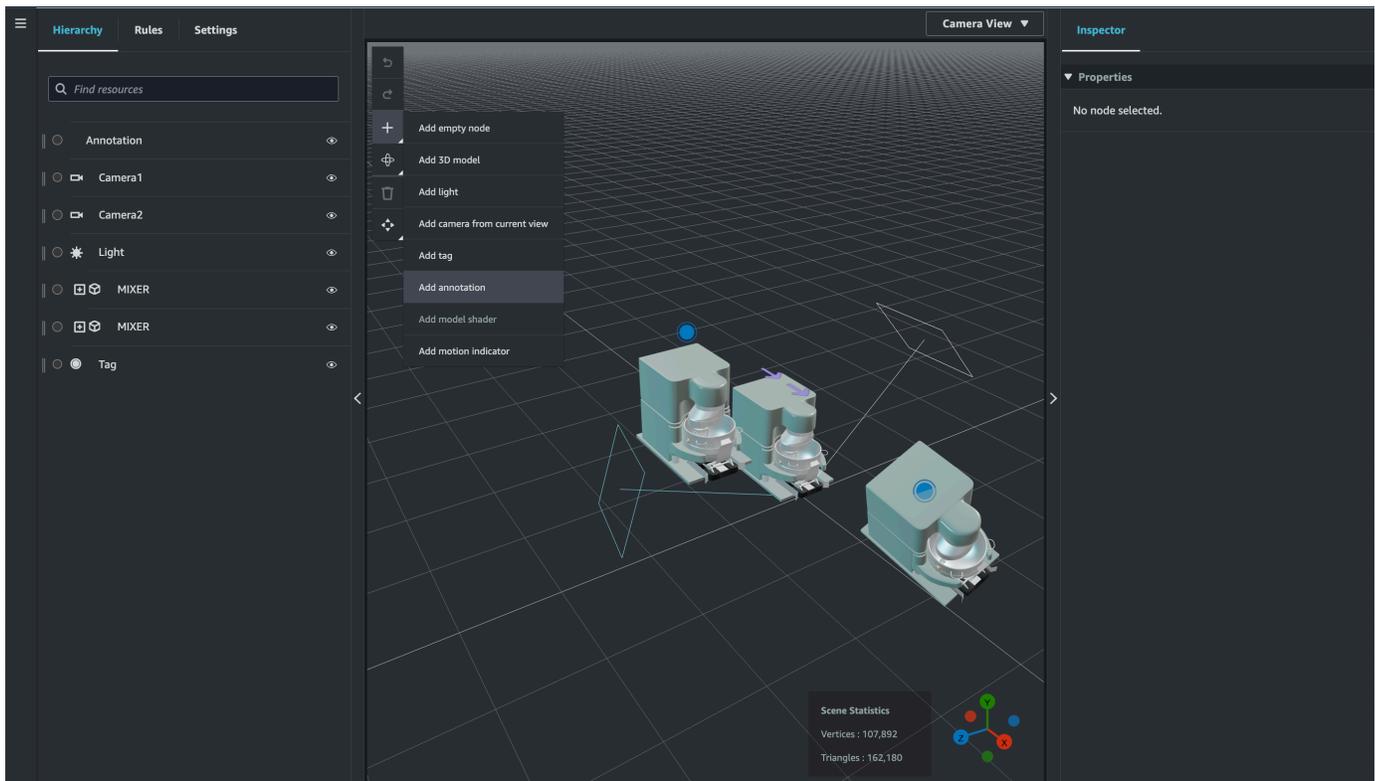
Para obter mais informações sobre como escrever em markdown, consulte a documentação oficial sobre sintaxe de markdown, [Sintaxe básica](#).

### Note

AWS IoT TwinMaker anotações e sobreposição apenas da sintaxe Markdown e não HTML.

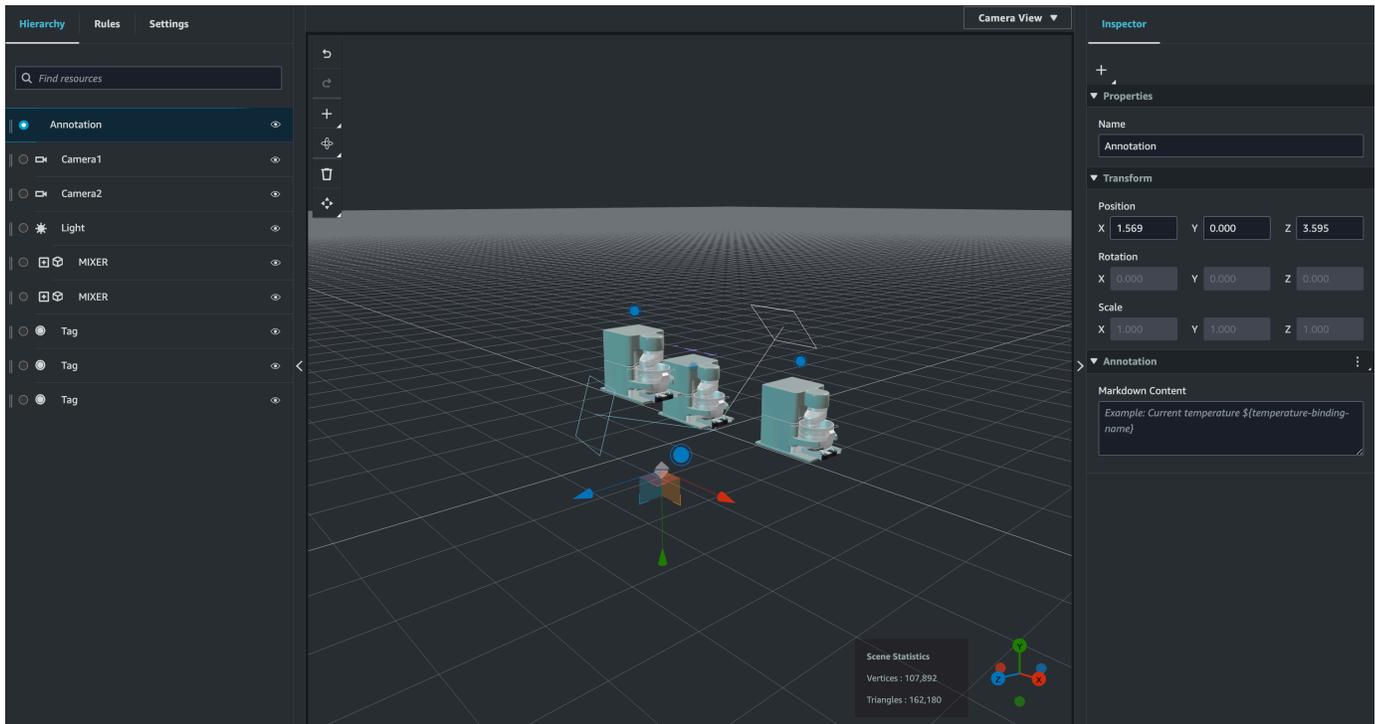
## Adicionar uma anotação a uma entidade

1. Navegue até sua cena no [console do AWS IoT TwinMaker](#).
2. Selecione um elemento da hierarquia da cena que você deseja anotar. Se nenhum elemento na hierarquia for selecionado, você poderá adicionar uma anotação à raiz.
3. Pressione o botão de adição + e escolha a opção Adicionar anotação.



4. Na janela Inspetor, à esquerda, role para baixo até a seção Anotações. Usando a sintaxe markdown, escreva o texto que você deseja que seja exibido na sua anotação.

Para obter mais informações sobre como escrever em markdown, consulte a documentação oficial sobre sintaxe de markdown, [Sintaxe básica](#).



5. Para vincular os dados da AWS IoT TwinMaker cena a uma anotação, escolha Adicionar associação de dados, adicione o ID da entidade e selecione o nome do componente e o nome da propriedade da entidade da qual você deseja exibir os dados. Você pode atualizar o nome da associação para usá-lo como uma variável do markdown e exibir os dados na anotação.

# 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

Add data binding

*Example: Current temperature `#{temperature-binding-name}`*

## Inspector



### ▼ Properties

#### Name

### ▼ Transform

#### Position

X

Y

Z

#### Rotation

X

Y

Z

#### Scale

X

Y

Z

### ▼ Annotation ⋮

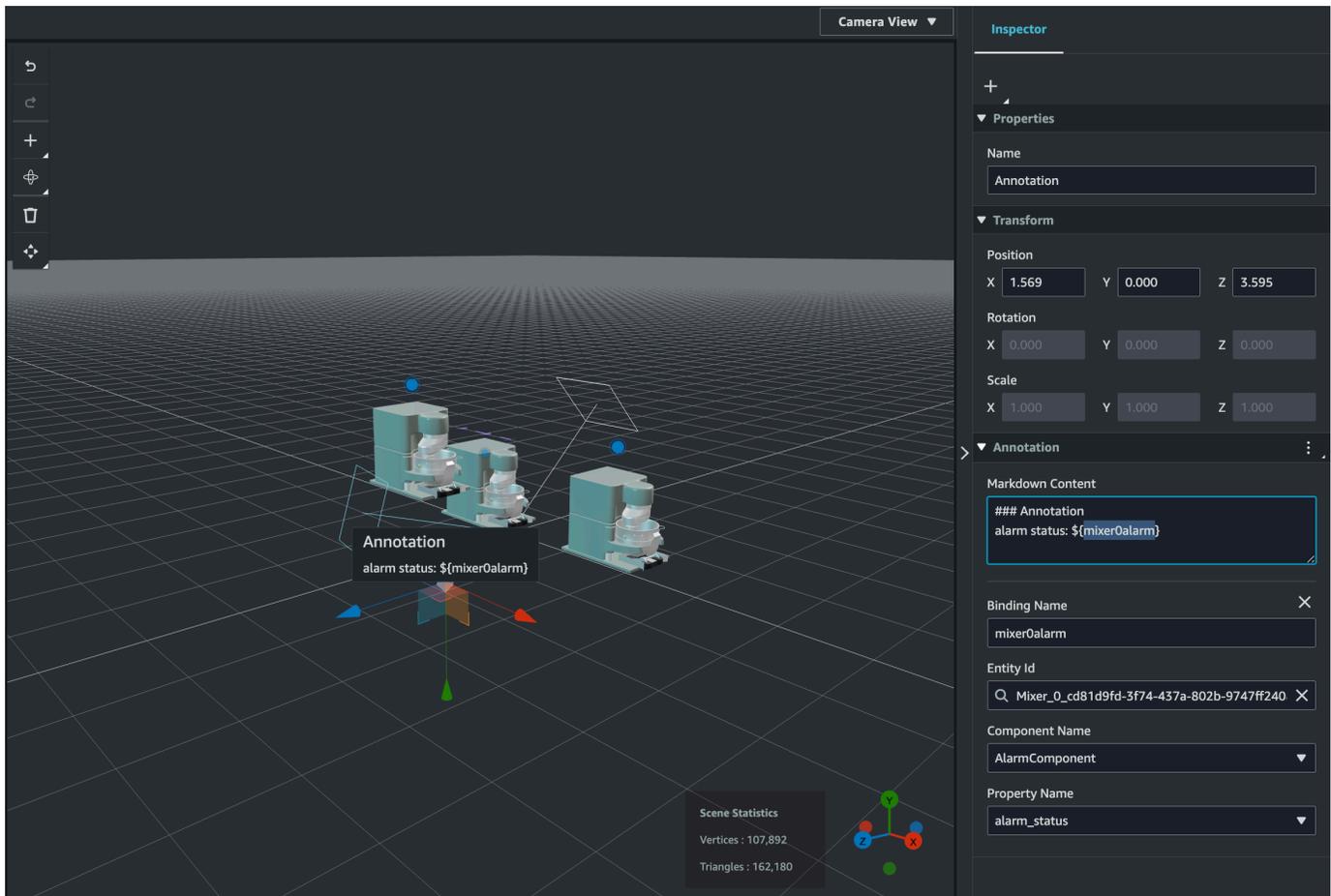
#### Markdown Content

```
Example: Current temperature  $\{temperature-binding-name\}$ 
```

## 6. O Nome da vinculação é usado para representar a variável da anotação.

Insira um nome de vinculação para mostrar o valor histórico mais recente da série temporal de uma entidade na anotação por meio AWS IoT TwinMaker da sintaxe variável: `${variable-name}`

Como exemplo, essa sobreposição exibe o valor do `mixer0alarm` na anotação com a sintaxe `${mixer0alarm}`.



## Adicione sobreposições às tags

Você pode criar sobreposições para suas AWS IoT TwinMaker cenas. As sobreposições de cena estão associadas a tags e podem ser usadas para revelar dados críticos associados às suas entidades de cena. A sobreposição é criada e renderizada no markdown.

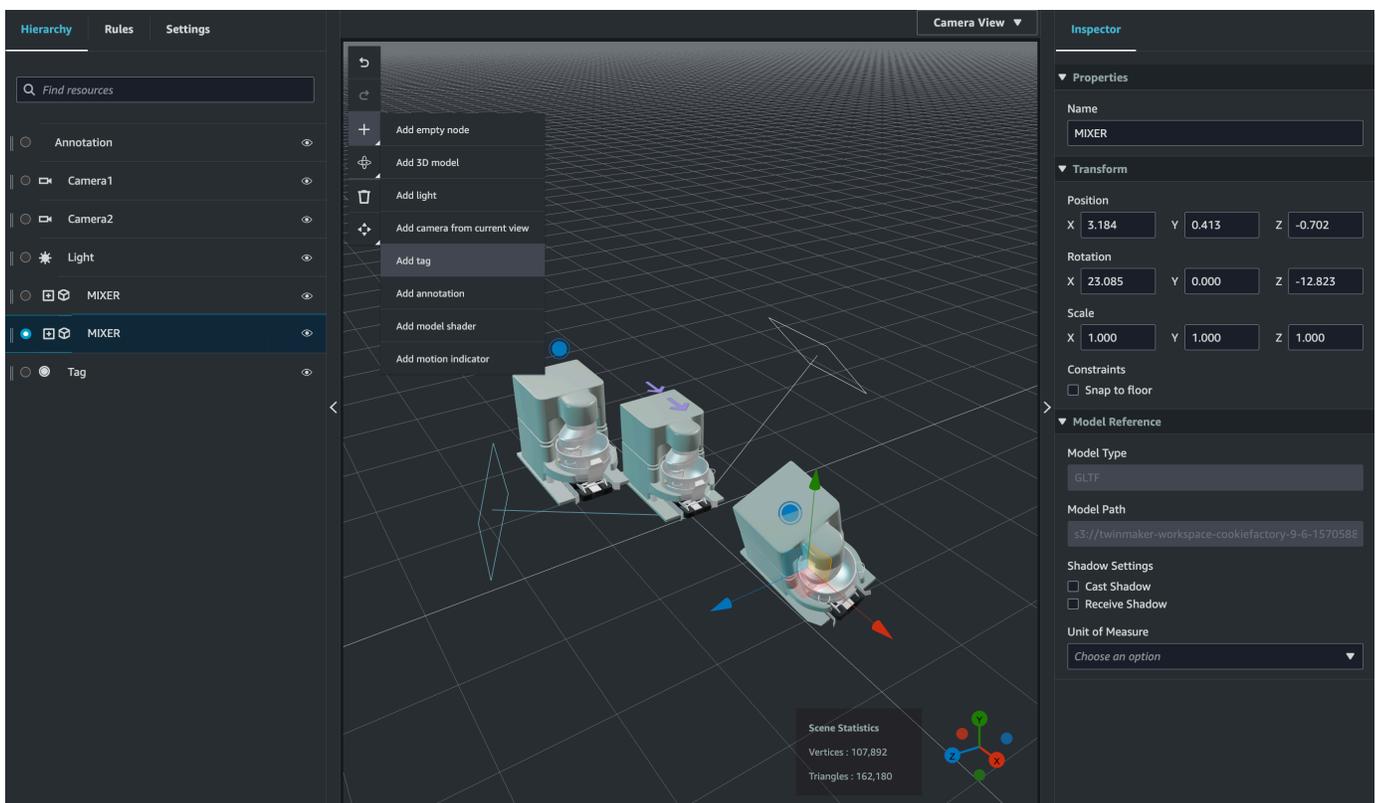
Para obter mais informações sobre como escrever em markdown, consulte a documentação oficial sobre sintaxe de markdown, [Sintaxe básica](#).

**Note**

Por padrão, uma sobreposição é visível em uma cena somente quando a tag associada a ela é selecionada. Você pode alternar isso nas configurações da cena para que todas as sobreposições fiquem visíveis ao mesmo tempo.

1. Navegue até sua cena no [console do AWS IoT TwinMaker](#).
2. A AWS IoT TwinMaker sobreposição está associada a uma cena de tag, você pode atualizar uma tag existente ou adicionar uma nova.

Pressione o botão de adição + e escolha a opção Adicionar tag.



3. No painel Inspector à direita, selecione o botão + (símbolo de adição) e selecione Adicionar sobreposição.

## 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*

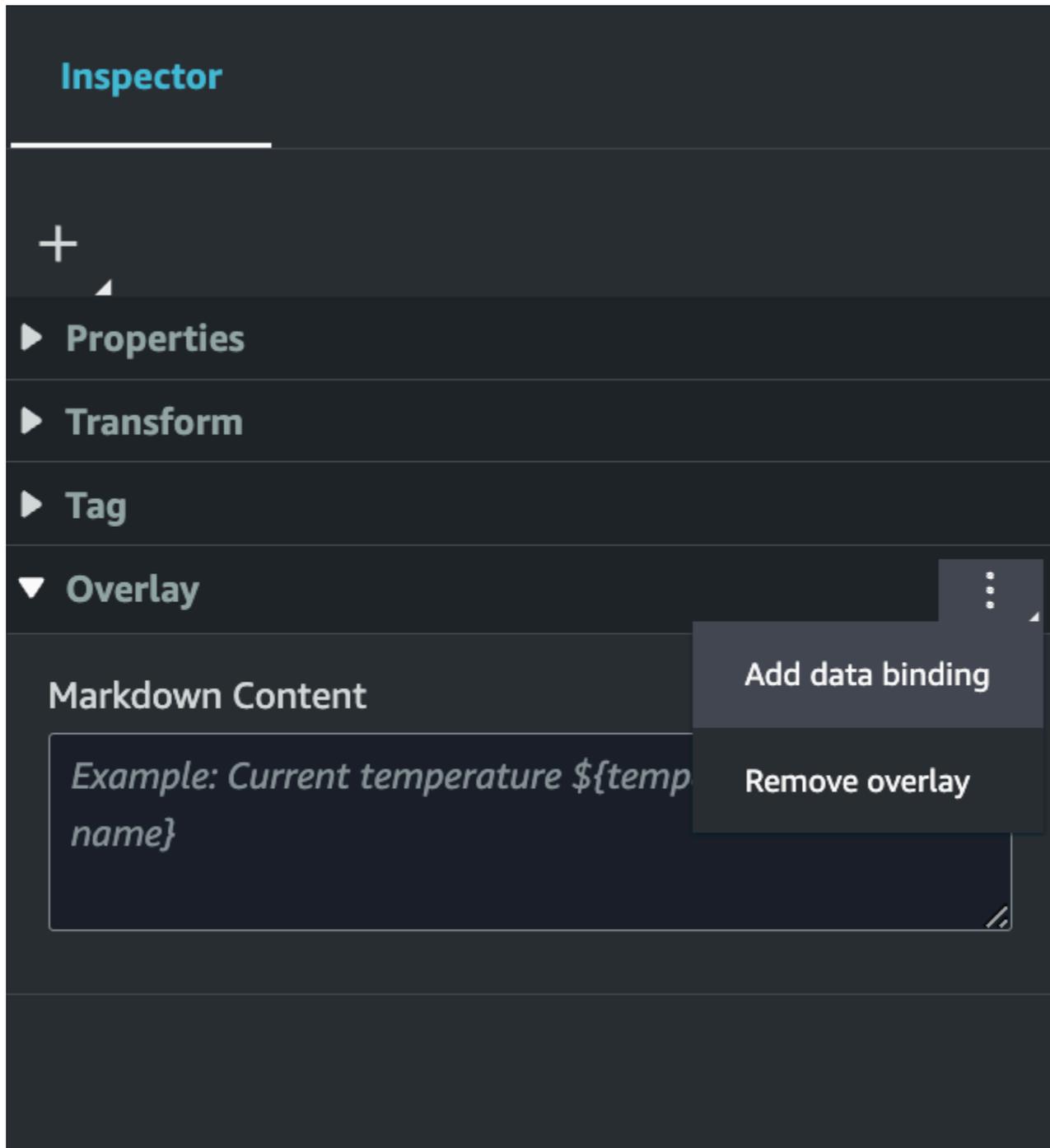


### Link Target

4. Na sintaxe do markdown, escreva o texto que você deseja que seja exibido na sua sobreposição.

Para obter mais informações sobre como escrever em markdown, consulte a documentação oficial sobre sintaxe de markdown, [Sintaxe básica](#).

5. Para vincular os dados da AWS IoT TwinMaker a uma sobreposição, selecione Adicionar vinculação de dados.



Adicione o nome da vinculação e o ID da entidade e, em seguida, selecione o nome do componente e o nome da propriedade da entidade da qual você deseja exibir dados.

6. Você pode destacar o valor histórico mais recente dos dados da série temporal de uma entidade na sobreposição por meio AWS IoT TwinMaker da sintaxe variável: `${variable-name}`

Como exemplo, essa sobreposição exibe o valor do `mixer0alarm`, na sobreposição com a sintaxe `${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 ativar a visibilidade da sobreposição, abra a guia Configurações no canto superior esquerdo e verifique se o botão de sobreposição está ativado para que todas as sobreposições fiquem visíveis ao mesmo tempo.

 Note

Por padrão, uma sobreposição é visível em uma cena somente quando a tag associada a ela é selecionada.

**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 suas cenas

Depois de criar uma cena, você pode adicionar entidades, componentes e configurar widgets aumentados na sua cena. Use componentes e widgets de entidades para modelar seu gêmeo digital e fornecer funcionalidades que correspondam ao seu caso de uso.

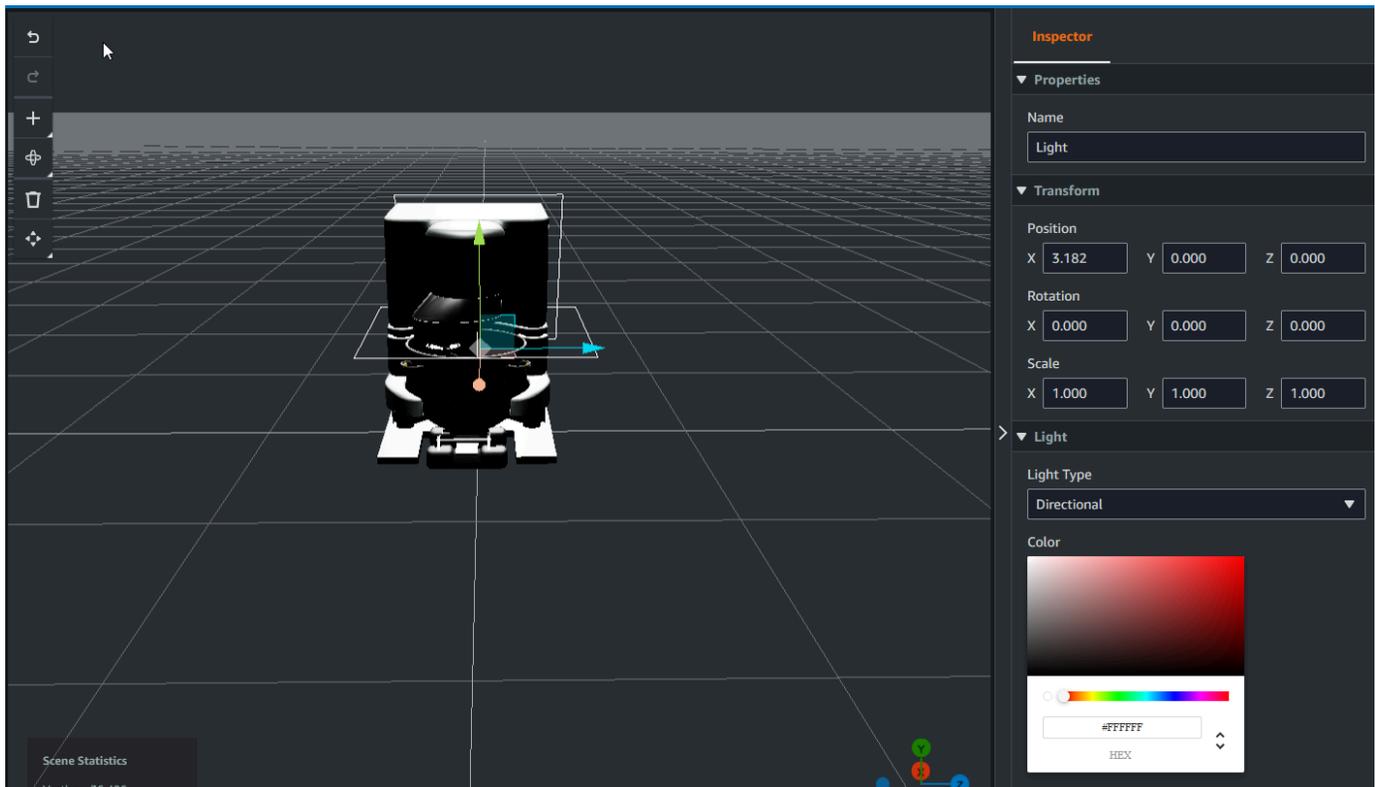
## Adicione modelos às suas cenas

Para adicionar modelos à sua cena, use o procedimento a seguir.

### Note

Para adicionar modelos à sua cena, você deve primeiro carregar os modelos na biblioteca de recursos do AWS IoT TwinMaker . Para obter mais informações, consulte [Carregar recursos para a Biblioteca AWS IoT TwinMaker de Recursos](#).

1. Na página do compositor de cenas, escolha o sinal de adição (+) e, em seguida, escolha Adicionar modelo 3D.
2. Na janela Adicionar recurso da biblioteca de recursos, escolha o CookieFactorMixerarquivo.glb e escolha Adicionar. O compositor de cenas é aberto.
3. Opcional: escolha o sinal de adição (+) e Adicionar luz.
4. Escolha cada opção de luz para ver como elas afetam a cena.



### Note

As cenas têm iluminação ambiente padrão. Para evitar a perda da taxa de quadros, considere limitar o número de luzes adicionais posicionadas em sua cena.

## Adicione widgets de interface de usuário aumentada com sombreador de modelo à sua cena

Os widgets de sombreamento do modelo podem alterar a cor de um objeto sob condições definidas por você. Por exemplo, você pode criar um widget de cores que altera a cor de um mixer de cookies em sua cena com base nos dados de temperatura do mixer.

Use o procedimento a seguir para adicionar widgets de sombreamento de modelo a um objeto selecionado.

1. Selecione um objeto na hierarquia ao qual você deseja adicionar um widget. Pressione o botão + e escolha Model Shader.

2. Para adicionar um novo grupo de regras visuais, primeiro siga as instruções abaixo para criar o `ColorRule`, em seguida, no painel Inspector do objeto da ID da Regra, escolha `ColorRule`
3. Selecione o `EntityID`, `ComponentName`, e `PropertyName` você deseja vincular o sombreador do modelo.

## Crie regras visuais as para as suas cenas

Você pode usar mapas de regras visuais para especificar as condições orientadas por dados que alteram a aparência visual de um widget de interface de usuário aumentada, como uma tag ou um sombreador de modelo. Exemplos de regras estão disponíveis, mas você também pode criar suas próprias. O exemplo a seguir mostra uma regra visual.

The screenshot displays the AWS IoT TwinMaker console interface for managing rules. It features a dark-themed sidebar on the left with a hamburger menu icon. The main content area shows a list of three rules, each with an 'Expression' field and a 'Target' section. The 'Target' section includes a dropdown menu for the icon type and a dropdown for the severity level, accompanied by a status icon. Below each rule is a 'Remove statement' button. At the bottom of the rule list are 'Add new statement' and 'Remove Rule' buttons. A rule named 'sampleTimeSeriesColorRule' is partially visible at the bottom, with a 'Rule Id' field below it.

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

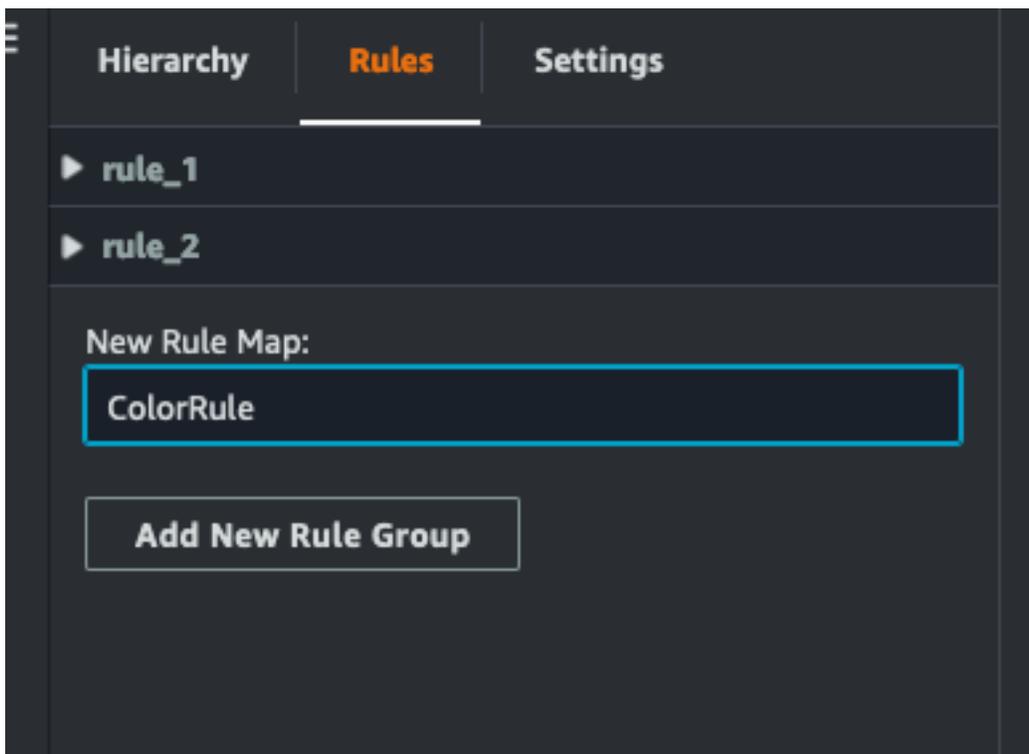
▶ sampleTimeSeriesColorRule

Rule Id

A imagem acima mostra uma regra para quando uma propriedade de dados previamente definida com ID 'temperature' é verificada em relação a um determinado valor. Por exemplo, se a "temperatura" for maior ou igual a 40, o estado mudará a aparência da etiqueta para um círculo vermelho. O destino, quando escolhido no painel do Grafana, preenche um painel de detalhes configurado para usar a mesma fonte de dados.

O procedimento a seguir mostra como adicionar um novo grupo de regras visuais para a camada de UI aumentada de colorização de malha.

1. Na guia de regras no console, insira um nome, como ColorRule no campo de texto, e escolha Adicionar novo grupo de regras.



2. Defina uma nova regra para seu caso de uso. Por exemplo, você pode criar um com base na propriedade de dados 'temperature', em que o valor relatado é menor que 20. Use a seguinte sintaxe para expressões de regras: Menor que é <, maior que é >, menor que ou igual é <=, maior que ou igual é >= e igual é ==. (Para obter mais informações, consulte a sintaxe [JEXL do Apache Commons](#).)
3. Defina o alvo para uma cor. Para definir uma cor, como #fcb03, use valores hexadecimais. (Para obter mais informações sobre valores [hexadecimais](#), consulte [Hexadecimal](#).)

## Criar tags para suas cenas

Uma tag é uma anotação adicionada a uma posição específica da coordenada  $x, y, z$  de uma cena. A tag usa uma propriedade de entidade para conectar uma parte da cena ao gráfico de conhecimento. Você pode usar uma tag para configurar o comportamento ou a aparência visual de um item na cena, como um alarme.

### Note

Para adicionar funcionalidade às tags, você aplica regras visuais a elas.

Use o procedimento a seguir para adicionar tags à sua cena.

1. Selecione um objeto na hierarquia, escolha o botão + e, em seguida, escolha Adicionar tag.
2. Dê um nome à tag. Em seguida, para aplicar uma regra visual, selecione um ID de grupo visual.
3. Nas listas suspensas, escolha o entityID, e. ComponentName PropertyName
4. Para preencher o campo Caminho de dados, escolha Criar DataFrameLabel.

## Formato de modelo 3D Tiles

### Usando blocos 3D em sua cena

Se você tiver longos tempos de espera ao carregar cenas 3D AWS IoT TwinMaker ou apresentar baixo desempenho de renderização ao navegar em um modelo 3D complexo, convém converter seus modelos em blocos 3D. Esta seção descreve o formato de mosaicos 3D e as ferramentas de terceiros disponíveis. Continue lendo para decidir se os blocos 3D são adequados para seu caso de uso e para obter ajuda para começar.

### Caso de uso de modelo complexo

Um modelo 3D em sua AWS IoT TwinMaker cena pode causar problemas de desempenho, como tempos de carregamento lentos e navegação lenta, se o modelo for:

- Grande: o tamanho do arquivo é maior que 100 MB.
- Denso: é composto por centenas ou milhares de malhas distintas.
- Complexo: a geometria da malha tem milhões de triângulos para formar formas complexas.

## Formato de telhas 3D

O [formato 3D Tiles](#) é uma solução para transmitir a geometria do modelo e melhorar o desempenho da renderização 3D. Ele permite o carregamento instantâneo de modelos 3D em uma AWS IoT TwinMaker cena e otimiza as interações 3D carregando partes de um modelo com base no que é visível na visualização da câmera.

O formato 3D Tiles foi criado pela [Cesium](#). O Cesium tem um serviço gerenciado para converter modelos 3D em blocos 3D chamado [Cesium Ion](#). Atualmente, essa é a melhor solução para criar blocos 3D, e recomendamos isso para seus modelos complexos nos [formatos suportados](#). Você pode registrar o Cesium e escolher o plano de assinatura apropriado com base nos requisitos de sua empresa na página de [preços do Cesium](#).

Para preparar um modelo de blocos 3D que você pode adicionar a uma AWS IoT TwinMaker cena, siga as instruções documentadas pelo Cesium Ion:

- [Importar um modelo para íon de céσιο](#)

## Carregue arquivos 3D de céσιο para AWS

Depois que seu modelo for convertido em blocos 3D, baixe os arquivos do modelo e, em seguida, carregue-os no bucket Amazon S3 do seu AWS IoT TwinMaker espaço de trabalho:

1. [Crie e baixe seu arquivo de modelos 3D Tiles](#).
2. Descompacte o arquivo em uma pasta.
3. Faça o upload de toda a pasta 3D Tiles no bucket do Amazon S3 associado ao seu AWS IoT TwinMaker espaço de trabalho. (Consulte [Carregamento de objetos](#) no Guia do usuário do Amazon S3.)
4. Se seu modelo de blocos 3D tiver sido carregado com sucesso, você verá um caminho de pasta do Amazon S3 na sua [biblioteca de AWS IoT TwinMaker recursos](#) com o tipo. Tiles3D

### Note

A Biblioteca AWS IoT TwinMaker de Recursos não suporta o upload direto de modelos de blocos 3D.

## Usando ladrilhos 3D em AWS IoT TwinMaker

AWS IoT TwinMaker está ciente de qualquer modelo de blocos 3D carregado no bucket do S3 do seu espaço de trabalho. O modelo deve ter um `tileset.json` e todos os arquivos dependentes (`.gltf`, `.b3dm`, `.i3dm`, `.cmpt`, `.pnts`) disponíveis no mesmo diretório do Amazon S3. O caminho do diretório Amazon S3 aparecerá na Biblioteca de Recursos com o tipo `Tiles3D`.

Para adicionar o modelo 3D Tiles à sua cena, siga estas etapas:

1. Na página do compositor de cenas, escolha o sinal de adição (+) e, em seguida, escolha Adicionar modelo 3D.
2. Na janela Adicionar recurso da biblioteca de recursos, escolha o caminho para seu modelo de blocos 3D com o tipo `Tiles3D`, em seguida, escolha Adicionar.
3. Clique na tela para colocar o modelo em sua cena.

### Diferenças de azulejos 3D

Atualmente, o 3D Tiles não oferece suporte a metadados geométricos e semânticos, o que significa que a hierarquia de malha do modelo original não está disponível para o recurso de seleção de submodelos. Você ainda pode adicionar widgets ao seu modelo de blocos 3D, mas não pode usar recursos ajustados aos submodelos: sombreador de modelo, transformações 3D separadas ou vinculação de entidades para uma malha de submodelo.

É recomendável usar a conversão de blocos 3D para grandes ativos que servem como contexto para o plano de fundo de uma cena. Se você quiser que um submodelo seja mais detalhado e anotado, ele deve ser extraído como um ativo glTF/GLB separado e adicionado diretamente à cena. Isso pode ser feito com ferramentas 3D gratuitas e comuns, como o [Blender](#).

Exemplo de caso de uso:

- Você tem um modelo de 1 GB de uma fábrica com salas e pisos de máquinas detalhados, caixas elétricas e canos de encanamento. As caixas elétricas e os canos precisam brilhar em vermelho quando os dados da propriedade associada ultrapassam um limite.
- Você isola as malhas de caixa e tubulação no modelo e as exporta para um glTF separado usando o Blender.
- Você converte a fábrica sem elementos elétricos e hidráulicos em um modelo 3D Tiles e o carrega no S3.

- Você adiciona o modelo 3D Tiles e o modelo glTF a uma AWS IoT TwinMaker cena na origem (0,0,0).
- Você adiciona componentes do sombreador do modelo aos submodelos de caixa elétrica e tubulação do glTF para tornar as malhas vermelhas com base nas regras de propriedade.

## Cenas dinâmicas

AWS IoT TwinMaker as cenas liberam o poder do [gráfico de conhecimento](#) ao armazenar nós e configurações da cena em um componente de entidade. Use o AWS IoT TwinMaker console para criar cenas dinâmicas para gerenciar, criar e renderizar cenas 3D com mais facilidade.

Características principais:

- Todos os objetos, configurações e vinculações de dados do nó de cena 3D são renderizados “dinamicamente” com base nas consultas do gráfico de conhecimento.
- Se você usar o Visualizador de Cena somente para leitura em um aplicativo Grafana ou personalizado, você pode obter atualizações para suas cenas em um intervalo de 30 segundos.

## Cenas estáticas versus dinâmicas

As cenas estáticas são compostas por um arquivo JSON de cena armazenado no S3 que tem detalhes de todos os nós e configurações da cena. Qualquer alteração na cena deve ser feita no documento JSON e salva no S3. Uma cena estática é a única opção se você tiver um [plano de preços básico](#).

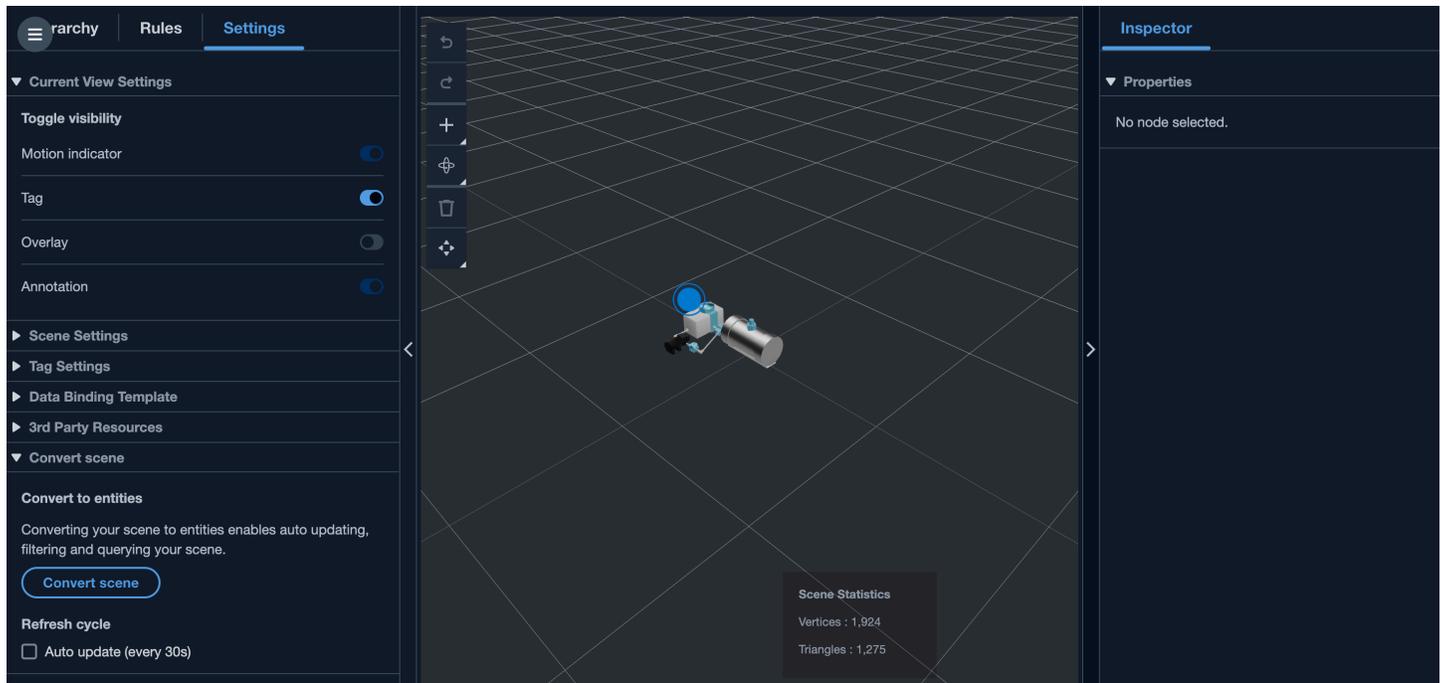
As cenas dinâmicas são compostas por um arquivo JSON de cena que tem configurações globais para a cena, enquanto todos os outros nós de cena e configurações de nós são armazenados como componentes de entidade no gráfico de conhecimento. As cenas dinâmicas são suportadas somente nos planos de preços de pacotes padrão e em camadas. Consulte [Alterne AWS IoT TwinMaker os modos de preços](#) para obter informações sobre como atualizar seu plano de preços).

Você pode converter uma cena estática existente em uma cena dinâmica seguindo estas etapas:

- Navegue até sua cena no [console do AWS IoT TwinMaker](#).
- No painel esquerdo, clique na guia Configurações.
- Expanda a seção Converter cena na parte inferior do painel.
- Clique no botão Converter cena e, em seguida, clique em Confirmar.

**Warning**

A conversão de uma cena estática para dinâmica é irreversível.



## Tipos e entidades de componentes de cena

Para criar componentes de entidade específicos da cena, os seguintes tipos de componentes 1P são suportados:

- `com.amazon.iottwinmaker.3d.component.camera` Um tipo de componente que armazena as configurações de um widget [de câmera](#).
- `com.amazon.iottwinmaker.3d.component.dataoverlay` Um tipo de componente que armazena as configurações de uma sobreposição [de um widget de anotação ou tag](#).
- `com.amazon.iottwinmaker.3d.component.light` Um tipo de componente que armazena as configurações de um widget light.
- `com.amazon.iottwinmaker.3d.component.modelref` Um tipo de componente que armazena as configurações e a localização no S3 de um modelo 3D usado em uma cena.
- [com.amazon.iottwinmaker.3d.component.modelshader](#) Um tipo de componente que armazena as configurações de um sombreador de modelo em um modelo 3D.

- `com.amazon.iottwinmaker.3d.component.motionindicator` Um tipo de componente que armazena as configurações de um widget indicador de movimento.
- `com.amazon.iottwinmaker.3d.component.submodelref` Um tipo de componente que armazena as configurações de um submodelo [de um modelo 3D](#).
- `com.amazon.iottwinmaker.3d.component.tag` Um tipo de componente que armazena as configurações de um widget [de tag](#).
- `com.amazon.iottwinmaker.3d.node` Um tipo de componente que armazena as configurações básicas de um nó de cena, como sua transformação 3D, nome e propriedades genéricas.

## Conceitos de cena dinâmica

As entidades dinâmicas da cena são armazenadas em uma entidade global \$SCENES rotulada. Cada cena é composta por uma entidade raiz e uma hierarquia de entidades filhas que correspondem à hierarquia dos nós da cena. Cada nó de cena abaixo da raiz tem um componente `com.amazon.iottwinmaker.3d.node` e um componente para o tipo de nó (modelo 3D, widget etc.).

### Warning

Não exclua manualmente nenhuma entidade da cena ou sua cena pode estar em um estado quebrado. Se você quiser excluir parcial ou totalmente uma cena, use a página do compositor de cenas para adicionar e excluir nós de cena e use a página de cenas para selecionar e excluir uma cena.

# Crie um aplicativo web personalizado usando componentes de AWS IoT TwinMaker interface do usuário

AWS IoT TwinMaker fornece componentes de interface de usuário de código aberto para desenvolvedores AWS IoT de aplicativos. Usando esses componentes de interface do usuário, os desenvolvedores podem criar aplicativos web personalizados com AWS IoT TwinMaker recursos habilitados para seus gêmeos digitais.

AWS IoT TwinMaker Os componentes da interface do usuário fazem parte do AWS IoT Application Kit, uma biblioteca de código aberto do lado do cliente que permite que os desenvolvedores de aplicativos de IoT simplifiquem o desenvolvimento de aplicativos complexos de IoT

AWS IoT TwinMaker Os componentes da interface do usuário incluem:

- AWS IoT TwinMaker fonte:

Um componente de conector de dados que permite recuperar dados e interagir com seus AWS IoT TwinMaker dados e gêmeos digitais.

Para obter mais informações, consulte a documentação da [fonte AWS IoT TwinMaker](#).

- Visualizador de cena:

Um componente de renderização 3D construído sobre `@react-three/fiber` que renderiza seu gêmeo digital e permite que você interaja com ele.

Para obter mais informações, consulte a documentação do [visualizador de cena](#).

- Reprodutor de vídeo:

Um componente de player de vídeo que permite transmitir um vídeo do Kinesis Video AWS IoT TwinMaker Streams por meio de.

Para obter mais informações, consulte a documentação do [Reprodutor de vídeo](#).

Para saber mais sobre como usar o Kit de AWS IoT Aplicativos, visite a página do [Github do Kit de AWS IoT Aplicativos](#).

Para obter instruções sobre como iniciar um novo aplicativo web usando o Kit de AWS IoT Aplicativos, visite a página oficial de documentação do [IoT App Kit](#).

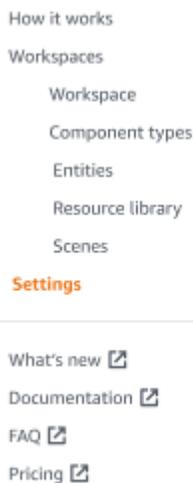
# Alterne AWS IoT TwinMaker os modos de preços

AWS IoT TwinMaker atualmente tem três modos de preços: pacote básico, padrão ou hierárquico. O modo de preço padrão é definido como o modo inicial.

Você pode mudar do modo de preços baseado no uso para o modo por nível a qualquer momento, mas a alteração entrará em vigor no início do próximo ciclo de cobrança. Depois de mudar do modo de preços baseado no uso para o modo de preços por nível, você não pode voltar para o modo de preços baseado no uso nos próximos três ciclos de uso. Caso mude do básico para o padrão, a alteração entrará em vigor imediatamente. Para obter detalhes e informações sobre custos, consulte [AWS IoT TwinMaker Preços](#)

Este procedimento mostra como alternar seu modo de preço no [console do AWS IoT TwinMaker](#):

1. Abra o [console de AWS IoT TwinMaker](#).
2. No painel de navegação esquerdo, selecione Configurações. A página de preços é aberta.



3. Escolha Alterar modo de preço.
4. Selecione os modos de pacote padrão ou por nível, conforme exibido na captura de tela a seguir.

### Select price mode

**Basic**

Basic pricing mode is determined by the data access calls sent during the current billing cycle. Does not include Knowledge Graph.

**Standard (current price mode)**

Standard pricing mode is determined by the entities used, queries made, and data access calls sent during the current billing cycle.

**Tiered bundle**

Tiered bundle pricing mode is based on 4 tiers of usage. Each tier is set by number of entities, and a usage threshold based on queries made.

---

### Standard pricing

The Standard pricing mode is determined by the entities used, queries made, and data access calls sent during the current billing cycle.

Pricing element	Pricing unit	Usage threshold
Unified data access calls	per MM	n/a
Queries	per 10K	n/a
Entities	per entity/month	n/a

Cancel Save

5. Escolha Salvar para confirmar seu novo modo de preços.
6. Agora você alterou seu modo de preços.

#### i Note

Você pode mudar do modo de preços baseado no uso para o modo por nível a qualquer momento, mas a alteração entrará em vigor no início do próximo ciclo de cobrança. Depois de mudar do modo de preços baseado no uso para o modo de preços por nível, você não pode voltar para o modo de preços baseado no uso nos próximos três ciclos de uso. Caso mude do básico para o padrão, a alteração entrará em vigor imediatamente.

# AWS IoT TwinMaker gráfico de conhecimento

O gráfico de AWS IoT TwinMaker conhecimento organiza todas as informações contidas em seus AWS IoT TwinMaker espaços de trabalho e as apresenta em um formato de gráfico visual. Você pode executar consultas em suas entidades, componentes e tipos de componentes para gerar gráficos visuais que mostrem as relações entre seus recursos AWS IoT TwinMaker .

Os tópicos a seguir mostram como usar e integrar o gráfico de conhecimento.

## Tópicos

- [AWS IoT TwinMaker conceitos básicos do gráfico de conhecimento](#)
- [Como executar consultas ao gráfico de AWS IoT TwinMaker conhecimento](#)
- [Integração da cena do gráfico de conhecimento](#)
- [Como usar o gráfico de AWS IoT TwinMaker conhecimento com a Grafana](#)
- [AWS IoT TwinMaker recursos adicionais do gráfico de conhecimento](#)

## AWS IoT TwinMaker conceitos básicos do gráfico de conhecimento

Este tópico aborda os principais conceitos e vocabulário do atributo de gráfico de conhecimento.

Como funciona o gráfico de conhecimento:

O gráfico de conhecimento cria relacionamentos entre entidades e seus componentes com as APIs existentes [CreateEntity](#) ou as [UpdateEntity](#) APIs. Um relacionamento é apenas uma propriedade de um tipo de dados especial [RELATIONSHIP](#) definido em um componente de uma entidade. AWS IoT TwinMaker O knowledge graph chama a [ExecuteQuery](#) API para fazer uma consulta com base em quaisquer dados nas entidades ou nos relacionamentos entre elas. O Knowledge Graph usa a linguagem de consulta PartiQL flexível (usada por muitos AWS serviços) que tem suporte recém-adicionado à sintaxe de correspondência gráfica para ajudá-lo a escrever suas consultas. Depois que as chamadas forem feitas, você poderá visualizar os resultados como uma tabela ou visualizá-los como um gráfico de nós e bordas conectados.

Termos-chave do gráfico de conhecimento:

- Grafo de entidades: uma coleção de nós e bordas em um espaço de trabalho.
- Nó: cada entidade em seu espaço de trabalho se torna um nó no grafo de entidades.

- **Borda:** cada propriedade de relacionamento definida em um componente de uma entidade se torna uma borda no grafo de entidades. Além disso, um relacionamento hierárquico pai-filho definido usando o `parentEntityId` campo de uma entidade também se torna uma borda no gráfico de entidades com um nome de relacionamento `isChildOf`. Todas as bordas são bordas direcionais.
- **Relacionamento:** Um AWS IoT TwinMaker relacionamento é um tipo especial de propriedade do componente de uma entidade. Você pode usar a [UpdateEntity](#) API AWS IoT TwinMaker [CreateEntity](#) ou para definir e editar um relacionamento. Em AWS IoT TwinMaker, um relacionamento deve ser definido em um componente de uma entidade. Um relacionamento não pode ser definido como um recurso isolado. Um relacionamento deve ser direcional de uma entidade para outra.

## Como executar consultas ao gráfico de AWS IoT TwinMaker conhecimento

Antes de usar o gráfico de AWS IoT TwinMaker conhecimento, verifique se você preencheu os seguintes pré-requisitos:

- Crie um AWS IoT TwinMaker espaço de trabalho. É possível criar um espaço de trabalho no [console do AWS IoT TwinMaker](#).
- Familiarize-se com AWS IoT TwinMaker o sistema de componentes de entidade e como criar entidades. Para ter mais informações, consulte [Crie sua primeira entidade](#).
- Familiarize-se com os conectores AWS IoT TwinMaker de dados da. Para ter mais informações, consulte [AWS IoT TwinMaker conectores de dados](#).

### Note

Para usar o gráfico de AWS IoT TwinMaker conhecimento, você precisa estar nos modos de preços de pacote padrão ou em camadas. Para ter mais informações, consulte [Altere AWS IoT TwinMaker os modos de preços](#).

Os procedimentos a seguir mostram como escrever, executar, salvar e editar consultas.

## Abra o editor de consulta

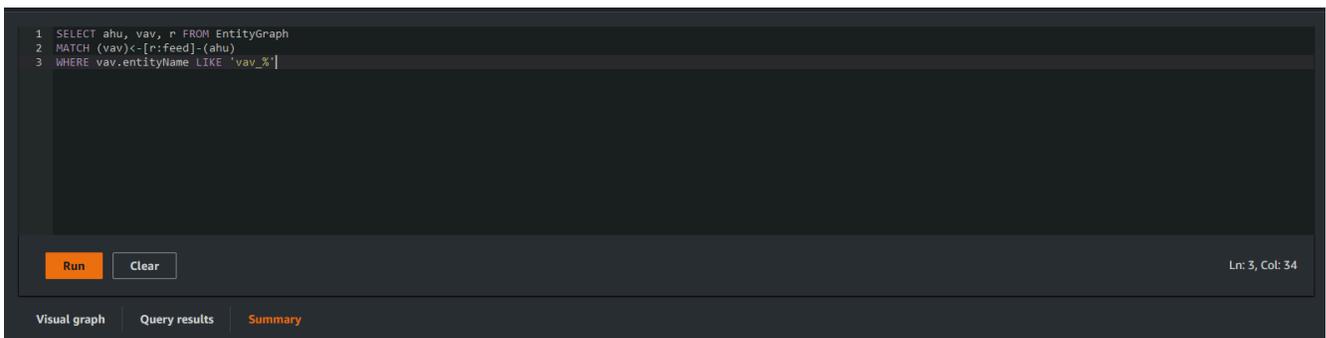
Para navegar até o editor de consultas do gráfico de conhecimento

1. Abra o [console de AWS IoT TwinMaker](#).
2. Abra o espaço de trabalho no qual você deseja usar o gráfico de conhecimento.
3. No menu de navegação esquerdo, escolha Editor de consulta.
4. O editor de consultas é aberto. Agora, você está pronto para executar consultas nos recursos do seu espaço de trabalho.

## Execute uma consulta

Para executar uma consulta e gerar um gráfico

1. No editor de consulta, escolha a guia Editor para abrir o editor de sintaxe.
2. No espaço do editor, escreva a consulta que você deseja executar nos recursos do seu espaço de trabalho.



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

Run Clear Ln: 3, Col: 34

Visual graph Query results Summary

No exemplo mostrado, a solicitação pesquisa entidades que contêm `vav_%` em seu nome e, em seguida, organiza essas entidades pelo `feed` relacionamento entre elas, usando o código a seguir.

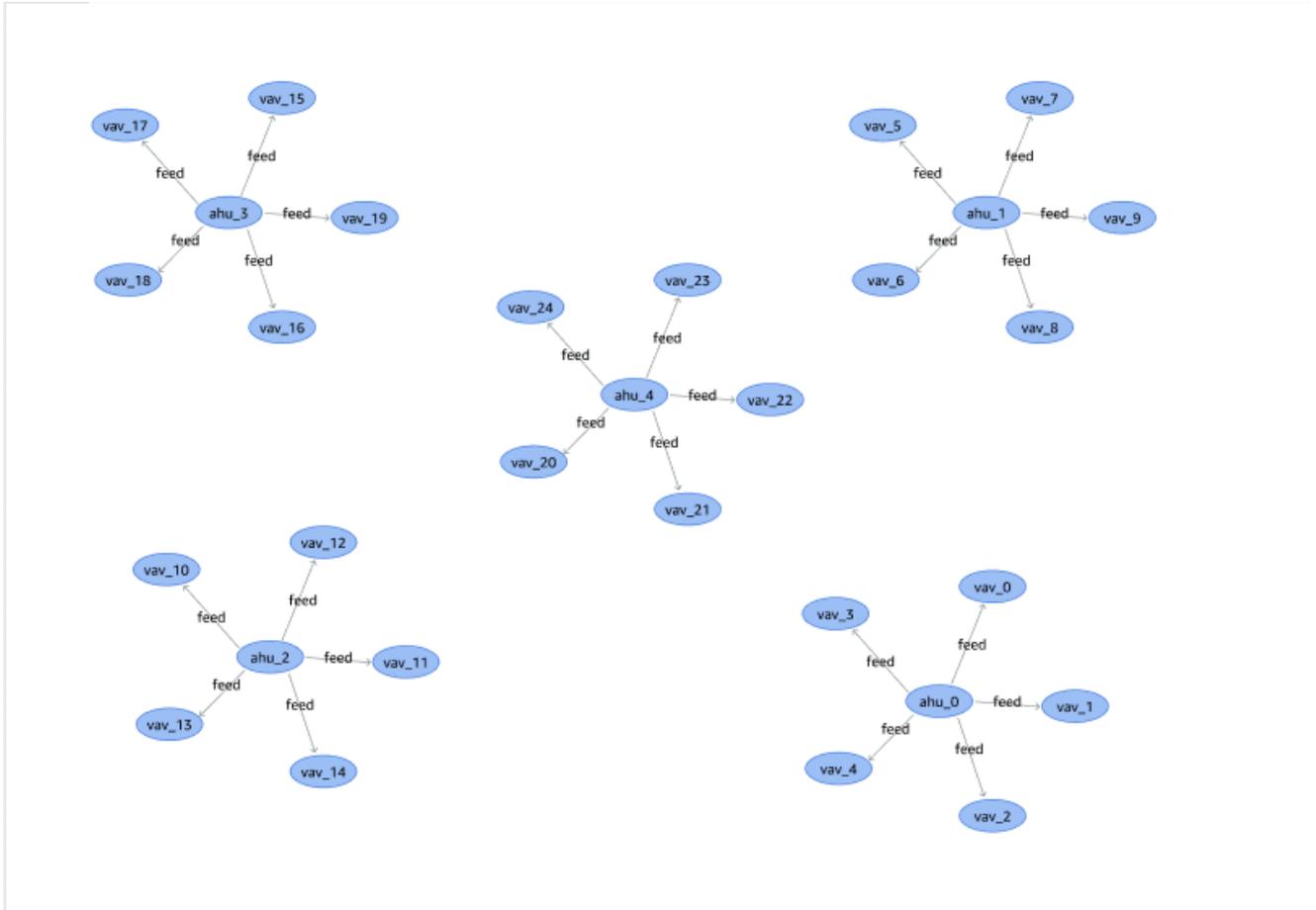
```
SELECT ahu, vav, r FROM EntityGraph
MATCH (vav)-[:feed]-(ahu)
WHERE vav.entityName LIKE 'vav_%'
```

**Note**

A sintaxe do gráfico de conhecimento usa [partiQL](#). Para obter informações sobre essa sintaxe, consulte [AWS IoT TwinMaker recursos adicionais do gráfico de conhecimento](#).

3. Escolha Executar consulta para executar a solicitação que você criou.

Um gráfico é gerado com base na sua solicitação.



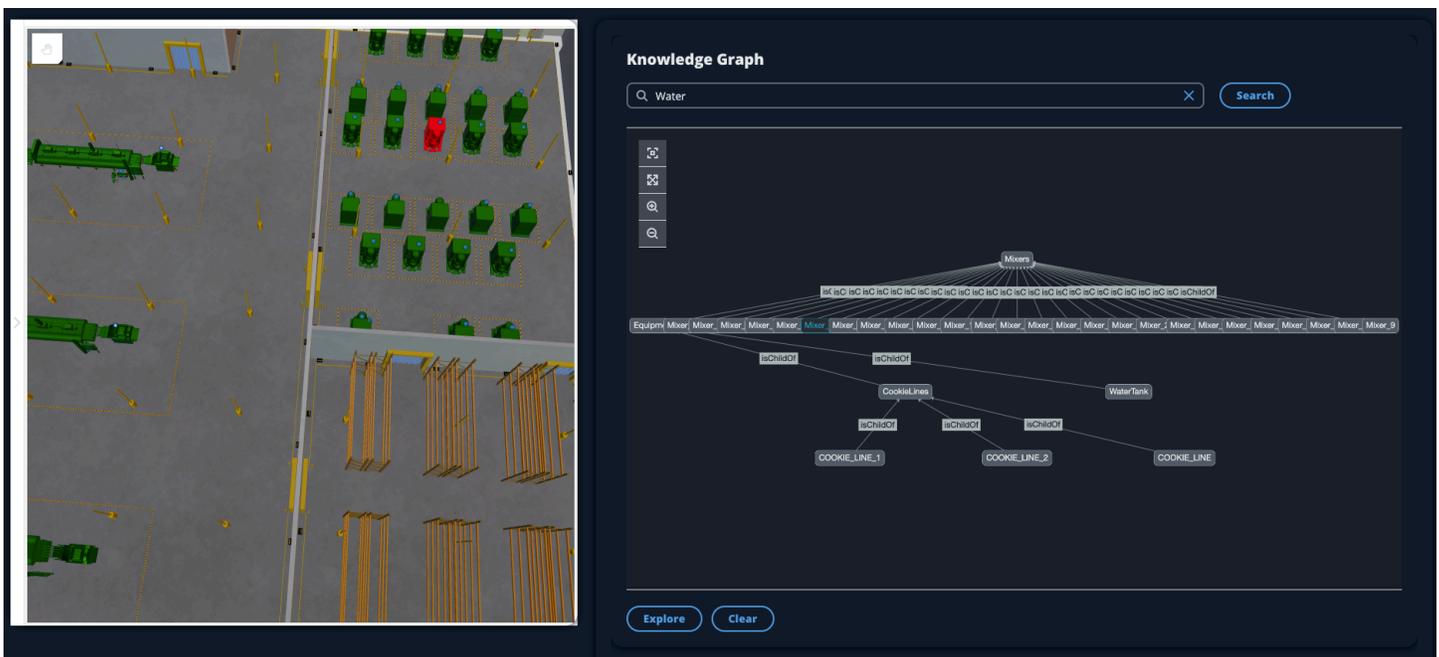
O gráfico de exemplo mostrado acima é baseado no exemplo de consulta na etapa 2.

4. Os resultados da consulta também são apresentados em uma lista. Escolha os resultados para ver os resultados da consulta em uma lista.
5. Opcionalmente, escolha Exportar como para exportar os resultados da consulta no formato JSON ou CSV.

Isso abrange o uso básico do gráfico de conhecimento no console. Para obter mais informações e exemplos que demonstram a sintaxe do gráfico de conhecimento, consulte [AWS IoT TwinMaker recursos adicionais do gráfico de conhecimento](#).

## Integração da cena do gráfico de conhecimento

Você pode usar os componentes AWS IoT do kit de aplicativos para criar um aplicativo web que integre o gráfico de conhecimento às suas AWS IoT TwinMaker cenas. Isso permite gerar gráficos com base nos nós 3D (os modelos 3D que representam seus equipamentos ou sistemas) que estão presentes em sua cena. Para criar um aplicativo que representa graficamente os nós 3D da sua cena, primeiro vincule os nós 3D às entidades em seu espaço de trabalho. Com esse mapeamento, AWS IoT TwinMaker representa graficamente as relações entre os modelos 3D presentes em sua cena e as entidades em seu espaço de trabalho. Em seguida, você pode criar um aplicativo web, selecionar modelos 3D com sua cena e explorar suas relações com outras entidades em formato gráfico.



Para ver um exemplo de um aplicativo web funcional que utiliza os componentes do kit de AWS IoT aplicativos para gerar gráficos em uma AWS IoT TwinMaker cena, consulte o [AWS IoT TwinMaker exemplo do aplicativo react](#) no github.

## AWS IoT TwinMaker pré-requisitos do gráfico de cena

Antes de criar um aplicativo web que use o gráfico de AWS IoT TwinMaker conhecimento em suas cenas, preencha os seguintes pré-requisitos:

- Crie um AWS IoT TwinMaker espaço de trabalho. É possível criar um espaço de trabalho no [console do AWS IoT TwinMaker](#).
- Familiarize-se com AWS IoT TwinMaker o sistema de componentes de entidade e como criar entidades. Para ter mais informações, consulte [Crie sua primeira entidade](#).
- Crie uma AWS IoT TwinMaker cena preenchida com modelos 3D.
- Familiarize-se com AWS IoT TwinMaker os componentes do kit de AWS IoT aplicativos. Para obter mais informações sobre os AWS IoT TwinMaker componentes, consulte [Crie um aplicativo web personalizado usando componentes de AWS IoT TwinMaker interface do usuário](#).
- Fique familiarizado com os conceitos do gráfico de conhecimento e a terminologia chave. Consulte [AWS IoT TwinMaker conceitos básicos do gráfico de conhecimento](#).

### Note

Para usar o gráfico de AWS IoT TwinMaker conhecimento e quaisquer recursos relacionados, você precisa estar nos modos de preços de pacote padrão ou em camadas. Para obter mais informações sobre AWS IoT TwinMaker preços, consulte [Altere AWS IoT TwinMaker os modos de preços](#).

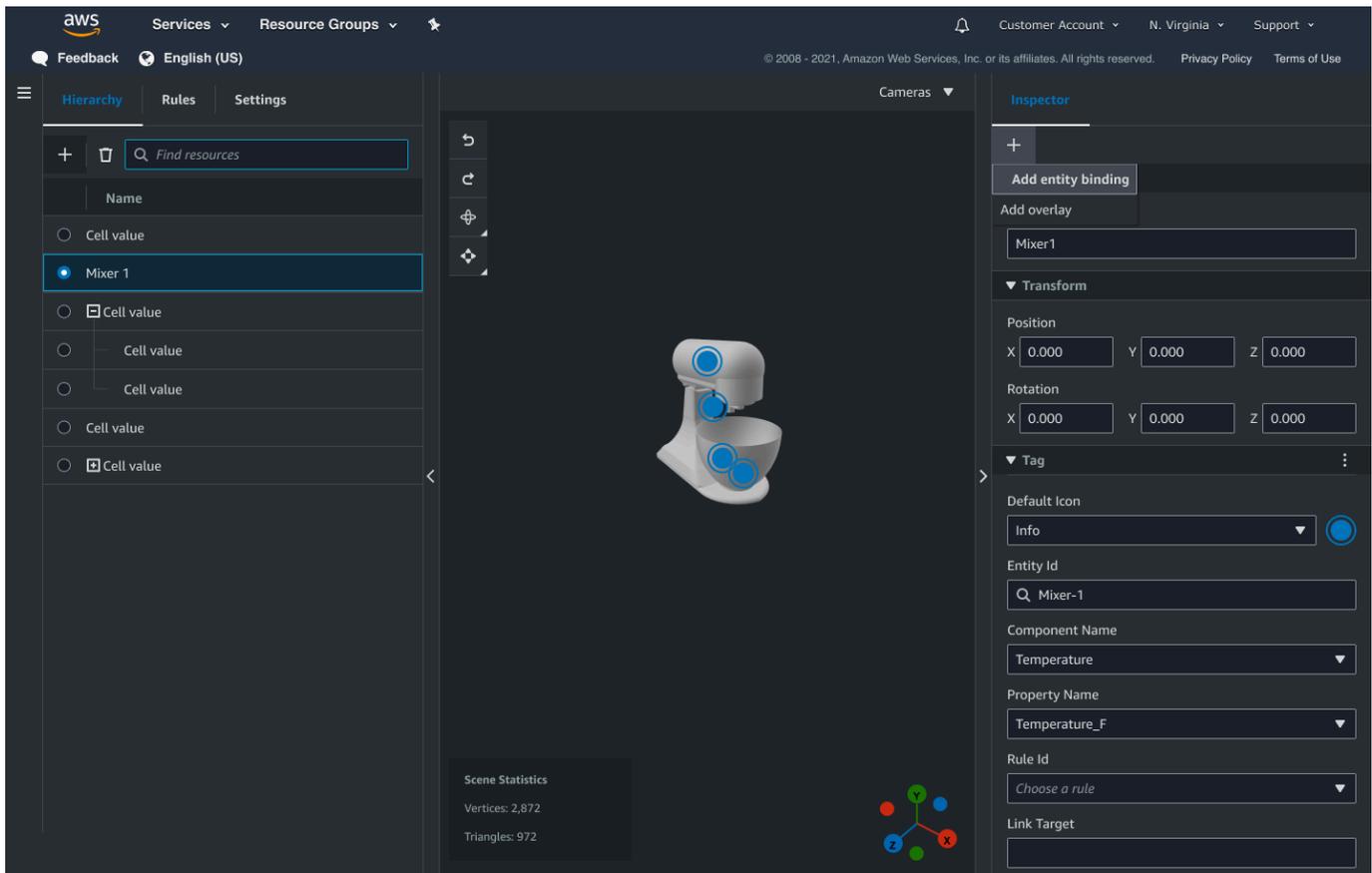
## Vincule nós 3D em sua cena

Antes de criar um aplicativo web que integre o gráfico de conhecimento à sua cena, vincule os modelos 3D, chamados de nós 3D, que estão presentes em sua cena à entidade de espaço de trabalho associada. Por exemplo, se você tiver um modelo de equipamento de mixagem em uma cena e uma entidade correspondente chamada `mixer_0`, crie uma associação de dados entre o modelo do mixer e a entidade que representa o mixer, para que o modelo e a entidade possam ser representados graficamente.

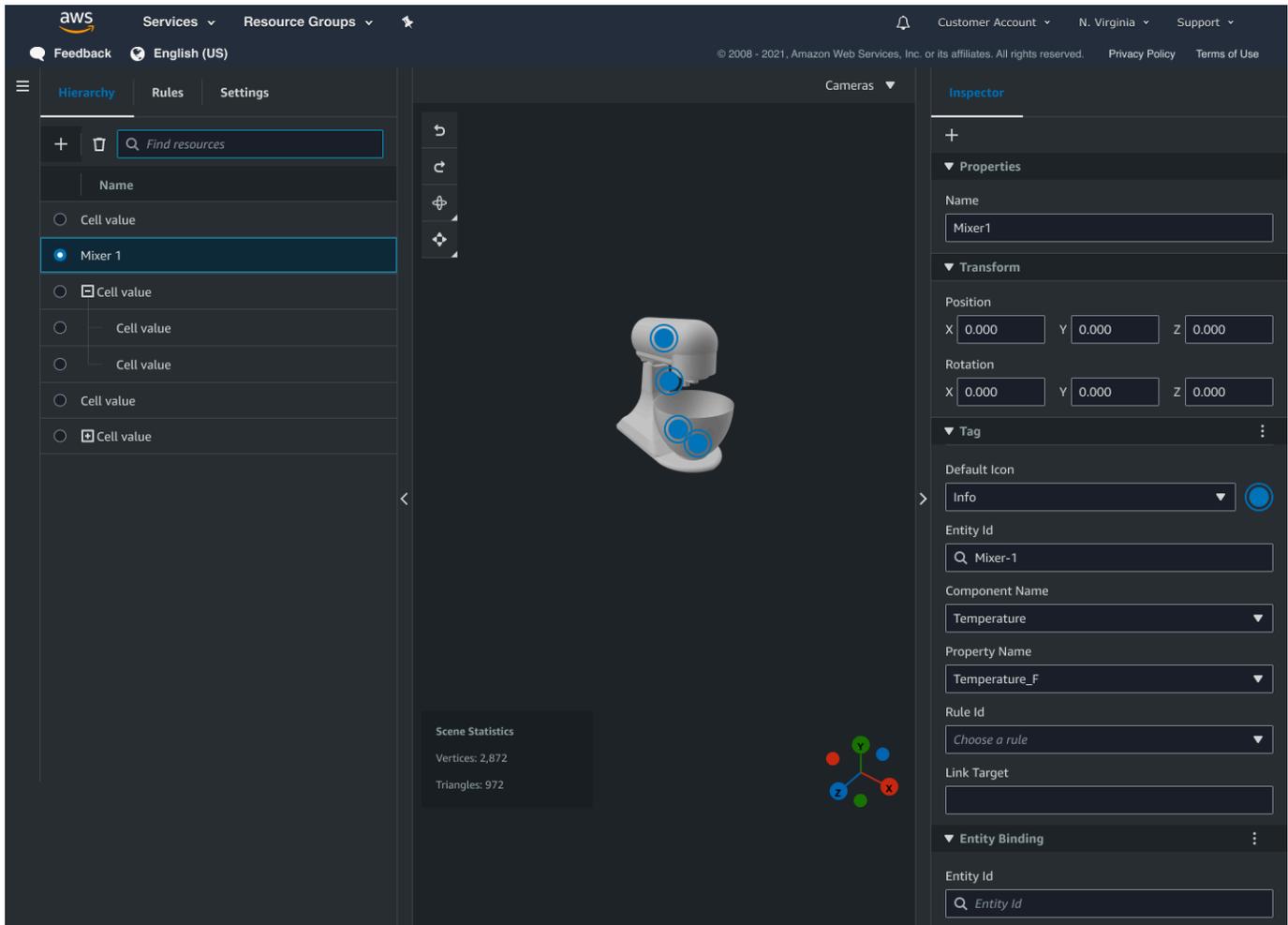
Para realizar uma ação de vinculação de dados

1. Fazer login no [console do AWS IoT TwinMaker](#).

2. Abra seu espaço de trabalho e selecione uma cena com os nós 3D que você deseja vincular.
3. Selecione um nó (modelo 3D) no compositor de cenas. Quando você seleciona um nó, ele abre um painel de inspeção no lado direito da tela.
4. No painel do inspetor, navegue até a parte superior do painel e selecione o botão +. Em seguida, escolha a opção Adicionar associação de entidade. Isso abrirá uma lista suspensa onde você poderá selecionar uma entidade para vincular ao seu nó atualmente selecionado.



5. No menu suspenso de vinculação de dados, selecione o ID da entidade que você deseja mapear para o modelo 3D. Para os campos Nome do componente e Nome da propriedade, selecione os componentes e propriedades que você deseja vincular.



Depois de fazer as seleções para os campos ID da entidade, Nome do componente e Nome da propriedade, a vinculação estará completa.

6. Repita esse processo para todos os modelos e entidades que você deseja representar graficamente.

#### Note

A mesma operação de vinculação de dados pode ser realizada em suas tags de cena. Basta selecionar uma tag em vez de uma entidade e seguir o mesmo processo para vincular a tag a um nó.

## Crie um aplicativo web

Depois de vincular suas entidades, use a biblioteca do kit de AWS IoT aplicativos para criar um aplicativo web com um widget de gráfico de conhecimento que permite visualizar sua cena e explorar as relações entre os nós e as entidades da cena.

Use os seguintes recursos para criar sua própria aplicação:

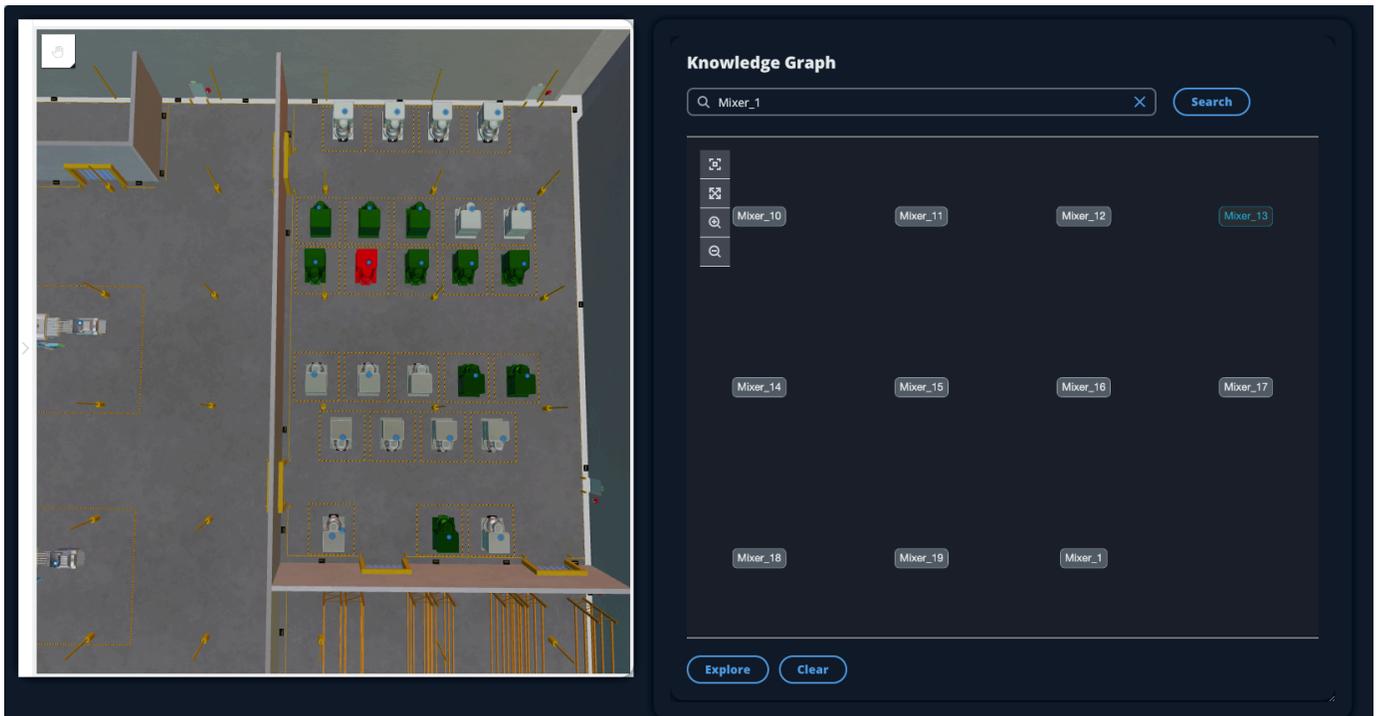
- O AWS IoT TwinMaker exemplo da documentação [Readme](#) do github do aplicativo react.
- O AWS IoT TwinMaker exemplo da [fonte](#) do aplicativo react no github.
- Documentação de [introdução](#) do kit de AWS IoT aplicativos.
- A documentação do [componente Video Player do](#) kit de AWS IoT aplicativos.
- A documentação do [componente Scene Viewer do](#) kit de AWS IoT aplicativos.

O procedimento a seguir demonstra a funcionalidade do componente visualizador de cena em um aplicativo web.

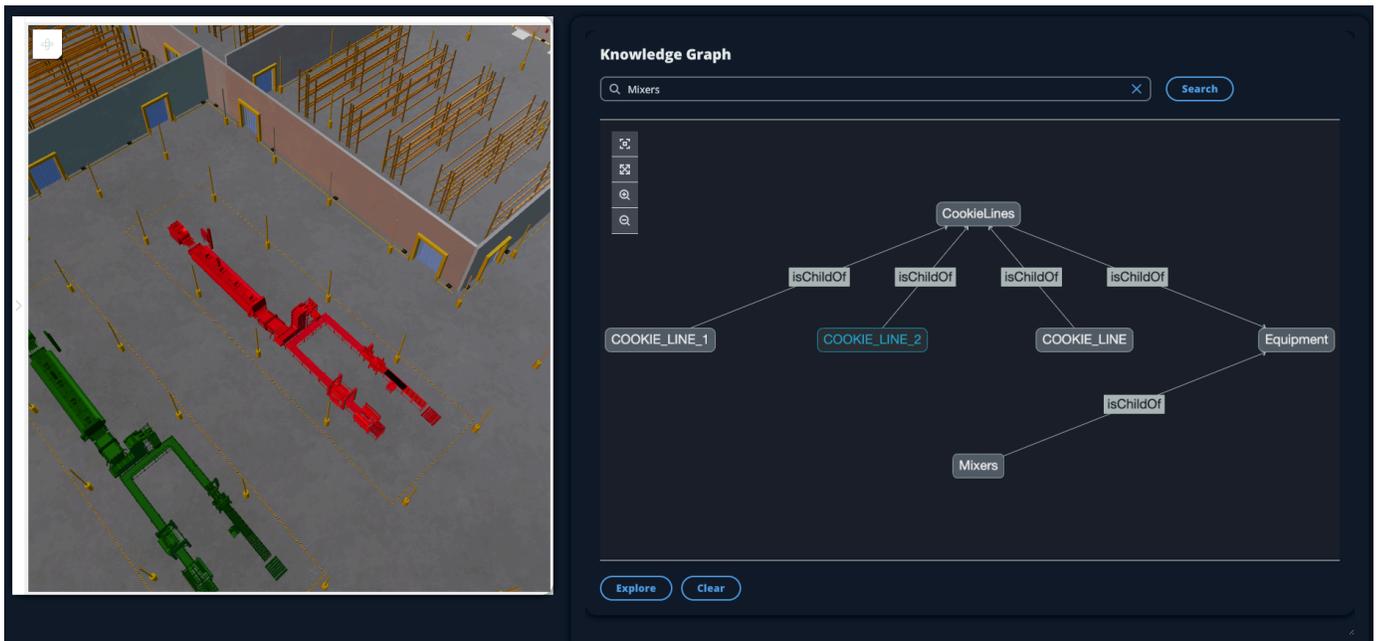
### Note

Esse procedimento é baseado na implementação do componente de visualização de cena do AWS IoT kit de aplicativos no aplicativo react de AWS IoT TwinMaker amostra.

1. Abra o componente de visualização de cena do aplicativo de reação de AWS IoT TwinMaker amostra. No campo de pesquisa, digite um nome de entidade ou nome de entidade parcial (pesquisa com distinção entre maiúsculas e minúsculas) e selecione o botão Pesquisar. Se um modelo estiver vinculado ao ID da entidade, o modelo na cena será destacado e um nó da entidade será mostrado no painel do visualizador de cena.



2. Para gerar um gráfico de todos os relacionamentos, selecione um nó no widget do visualizador de cena e selecione o botão Explorar.



3. Pressione o botão Limpar para limpar a seleção atual do gráfico e começar de novo.

# Como usar o gráfico de AWS IoT TwinMaker conhecimento com a Grafana

Esta seção mostra como adicionar um painel do editor de consultas ao painel do AWS IoT TwinMaker Grafana para executar e exibir consultas.

## AWS IoT TwinMaker pré-requisitos do editor de consultas

Antes de usar o gráfico de AWS IoT TwinMaker conhecimento na Grafana, preencha os seguintes pré-requisitos:

- Crie um AWS IoT TwinMaker espaço de trabalho. É possível criar um espaço de trabalho no [console do AWS IoT TwinMaker](#).
- Configure AWS IoT TwinMaker para uso com o Grafana. Para obter instruções, consulte [Integração do painel do Grafana AWS IoT TwinMaker](#).

### Note

Para usar o gráfico de AWS IoT TwinMaker conhecimento, você precisa estar nos modos de preços de pacote padrão ou em camadas. Para ter mais informações, consulte [Altere AWS IoT TwinMaker os modos de preços](#).

## AWS IoT TwinMaker permissões do editor de consultas

Para usar o editor de AWS IoT TwinMaker consultas no Grafana, você precisa ter uma função do IAM com permissão para a ação. `iottwinmaker:ExecuteQuery` Adicione essa permissão à sua função no painel do espaço de trabalho, conforme mostrado neste exemplo:

```
{
  "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

Ao configurar sua fonte de dados do AWS IoT TwinMaker Grafana, certifique-se de usar a função com essa permissão para o campo ARN de assumir função. Depois de adicioná-lo, você pode selecionar seu espaço de trabalho no menu suspenso ao lado de Espaço de trabalho.

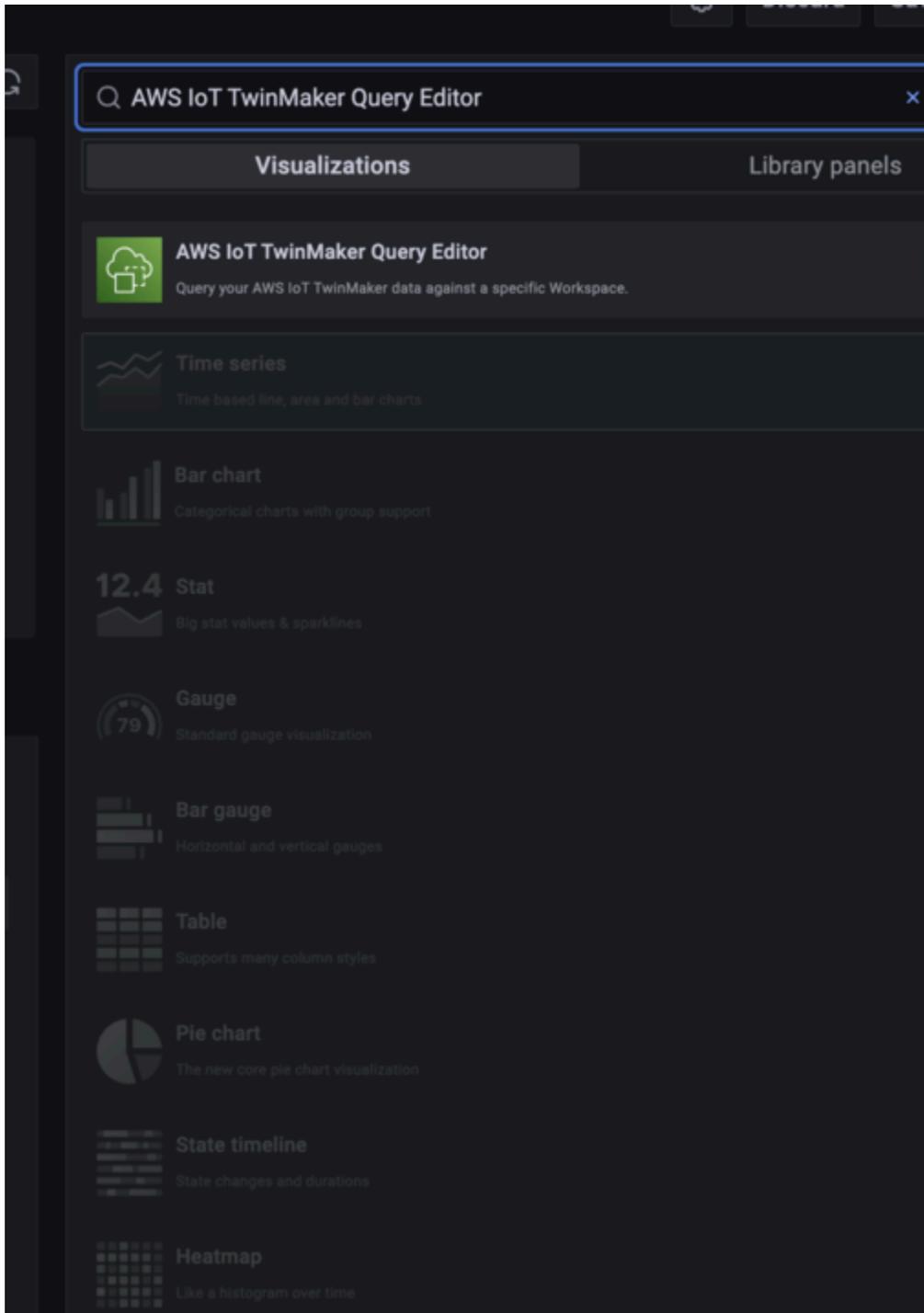
Para ter mais informações, consulte [Como criar um perfil do IAM para o painel](#).

## Configurar o painel do editor de AWS IoT TwinMaker consultas

Para configurar um novo painel de controle da Grafana para o gráfico de conhecimento

1. Abra seu painel da AWS IoT TwinMaker Grafana.
2. Crie um novo painel de controle. Para obter etapas detalhadas sobre como criar um painel, consulte [Criar um painel na documentação](#) da Grafana.

3. Na lista de visualizações, selecione Editor de AWS IoT TwinMaker consultas.



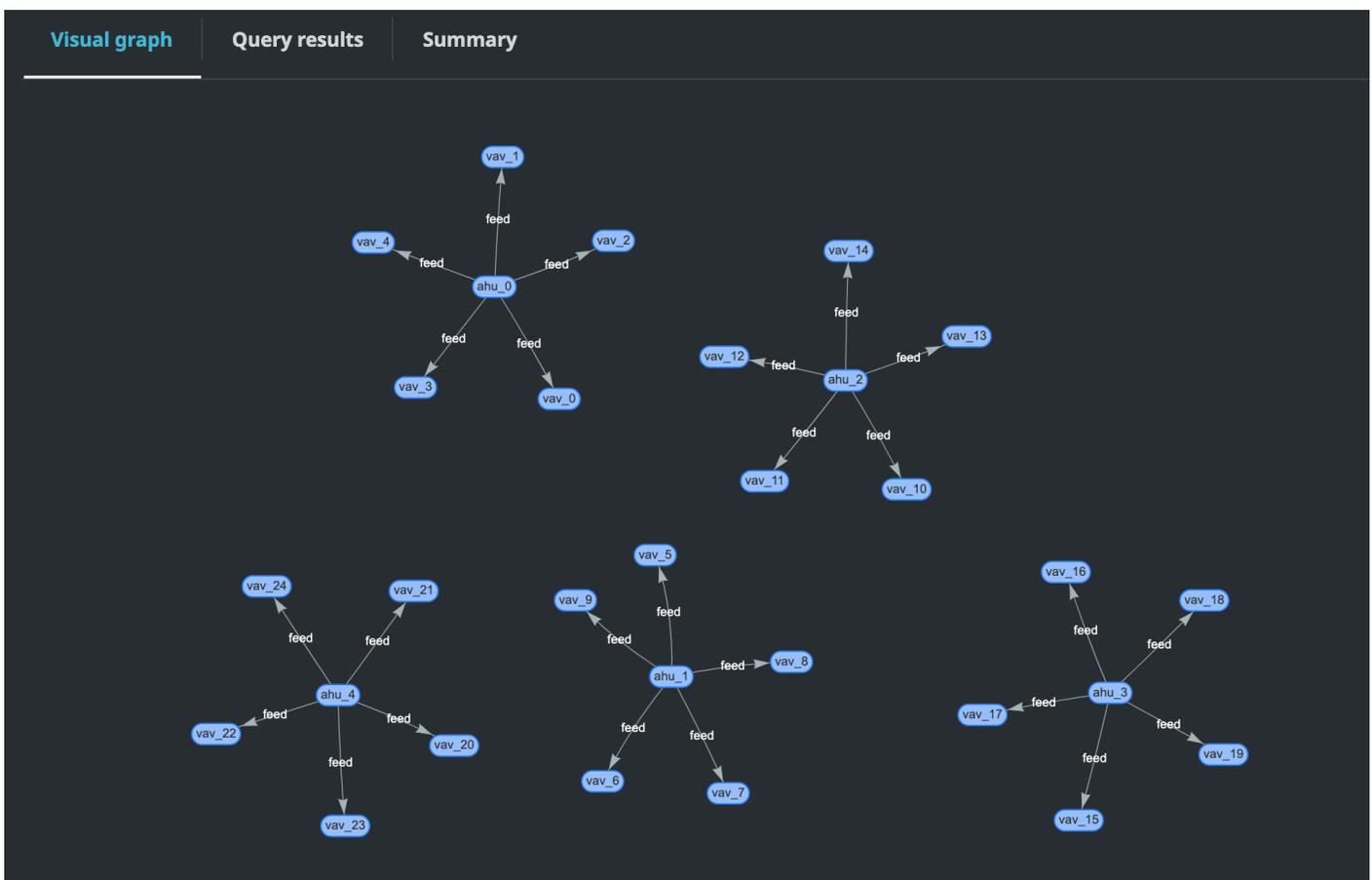
4. Selecione a fonte de dados na qual executará consultas.
5. (Opcional) Adicione um nome para o novo painel no campo fornecido.
6. Selecione Aplicar para salvar e confirmar seu novo painel.

O painel do gráfico de conhecimento funciona de forma semelhante ao editor de consultas fornecido no AWS IoT TwinMaker console. Você pode executar, escrever e limpar as consultas feitas no painel. Para obter mais informações sobre como escrever consultas, consulte [AWS IoT TwinMaker recursos adicionais do gráfico de conhecimento](#).

### Como usar o editor AWS IoT TwinMaker de consultas

Os resultados de suas consultas são exibidos de três maneiras, conforme mostrado nas imagens a seguir: visualizados em um gráfico, listados em uma tabela ou apresentados como um resumo da execução.

- Visualização gráfica:



O gráfico visual exibe somente dados para consultas que tenham pelo menos uma relação no resultado. O gráfico exibe entidades como nós e relacionamentos como bordas direcionadas no gráfico.

- Dados tabulares:

Visual graph
Query results
Summary

**Results returned (25)**

Export as ▼

< 1 >
⚙️

ahu	vav	r
<pre>{ "arn": "arn:aws:iottwinmaker:us-east-1:086801877023:workspace/SmartBuilding/entity/ahu_23565bbb-3ec6-3ca0-9fce-e9b0cfeae7b1", "creationDate": 1667895668496, "entityId": "ahu_23565bbb-3ec6-3ca0-9fce-e9b0cfeae7b1", "entityName": "ahu_0", "lastUpdateDate": 1667895669319, "workspaceId": "SmartBuilding", "description": "", "components": [ { "componentName": "AhuComponent", "componentTypeId": "com.example.query.equipment.ahu", "properties": [ ] } ] }</pre>	<pre>{ "arn": "arn:aws:iottwinmaker:us-east-1:086801877023:workspace/SmartBuilding/entity/vav_66461816-02ab-355f-afd2-62a2cc92d336", "creationDate": 1667895664133, "entityId": "vav_66461816-02ab-355f-afd2-62a2cc92d336", "entityName": "vav_2", "lastUpdateDate": 1667895665269, "workspaceId": "SmartBuilding", "description": "", "components": [ { "componentName": "VavComponent", "componentTypeId": "com.example.query.equipment.vav", "properties": [ { "propertyName": "airTerminalUnitCertificates", "propertyValue": [ "AHRI", "UL" ] }, { "propertyName": "airTerminalUnitBranchCount", "propertyValue": 2 }, { "propertyName": "airTerminalUnitDimension", "propertyValue": { "width": 15, "length": 30, "height": 15 } } ] } ] }</pre>	<pre>{ "relationshipName": "feed", "sourceEntityId": "ahu_23565bbb-3ec6-3ca0-9fce-e9b0cfeae7b1", "targetEntityId": "vav_66461816-02ab-355f-afd2-62a2cc92d336", "sourceComponentName": "AhuComponent", "sourceComponentTypeId": "com.example.query.equipment.ahu" }</pre>

O formato de dados tabulares exibe os dados de todas as consultas. Você pode pesquisar na tabela resultados específicos ou subconjuntos dos resultados. Os dados podem ser exportados no formato JSON ou CSV.

- **Resumo da execução**

Visual graph	Query results	Summary		
Start	Status	Response	Statement	Duration
2022-11-15 11:36:08 UTC-0800	✔ Success	25 returned	SELECT ahu, vav, r FROM EntityGraph MATCH (vav)<-[r:feed]->(ahu) WHERE vav.entityName LIKE 'vav_%'	0.833 sec

O resumo da execução exibe a consulta e os metadados sobre o status da consulta.

# AWS IoT TwinMaker recursos adicionais do gráfico de conhecimento

Esta seção fornece exemplos básicos da sintaxe do partiQL usada para escrever consultas no gráfico de conhecimento, bem como links para a documentação do partiQL que fornecem informações sobre o modelo de dados do gráfico de conhecimento.

- [Documentação do modelo de dados gráficos PartiQL](#)
- [Documentação de consulta gráfica partiQL](#)

Esse conjunto de exemplos mostra consultas básicas com suas respostas. Use isso como referência para escrever suas próprias consultas.

## Consultas básicas

- Obtenha todas as entidades com um filtro

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

Essa consulta retorna todas as entidades em um espaço de trabalho com o nome `room_0`.

**FROM** cláusula: `EntityGraph` é a coleção de gráficos que contém todas as entidades e seus relacionamentos em um espaço de trabalho. Essa coleção é criada e gerenciada automaticamente AWS IoT TwinMaker com base nas entidades em seu espaço de trabalho.

**Cláusula MATCH**: especifica um padrão que corresponde a uma parte do gráfico. Nesse caso, o padrão `(entity)` corresponde a cada nó no gráfico e está vinculado à variável da entidade. A cláusula `FROM` deve ser seguida pela cláusula `MATCH`.

**WHERE** cláusula: especifica um filtro no `entityName` campo do nó, onde o valor deve corresponder. `room_0`

**SELECT** cláusula: especifica a `entity` variável para que todo o nó da entidade seja retornado.

Resposta:

```
{
```

```

"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` Retorna metadados sobre a coluna, como nome e tipo. O tipo retornado é `NODE`. Isso indica que o nó inteiro foi retornado. Outros valores para o tipo podem `EDGE` indicar uma relação ou `VALUE` indicar um valor escalar, como um inteiro ou uma string.

O `rows` retorna uma lista de linhas. Como apenas uma entidade foi correspondida, um `rowData` é retornado, que contém todos os campos em uma entidade.

**Note**

Ao contrário do SQL, em que você só pode retornar valores escalares, você pode retornar um objeto (como JSON) usando partiQL.

Cada nó contém todos os campos em nível de entidade `entityId`, como, `arn` e campos em nível de componente `components`, como `componentName`, `componentTypeId` e também campos em nível de propriedade, `properties` como e, tudo como um JSON `propertyName` aninhado `propertyValue`.

- Obtenha todos os relacionamentos com um filtro:

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

Essa consulta retorna todos os relacionamentos em um espaço de trabalho com o nome do relacionamento `isLocationOf`.

A `MATCH` cláusula: especifica um padrão que corresponde a dois nós (indicados por `()`) conectados por uma borda direcionada (indicada por `-[]->`) e vinculados a uma variável chamada `relationship`

A `WHERE` cláusula: especifica um filtro no `relationshipName` campo da borda, onde está o valor `isLocationOf`

A `SELECT` cláusula: especifica a variável de relacionamento para que todo o nó de borda seja retornado.

Resposta

```

{
  "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
]
}

```

O tipo da coluna em `columnDescriptions` é um `EDGE`.

Cada um `rowData` representa uma borda com campos como `relationshipName`. É o mesmo nome da propriedade do relacionamento definido na entidade. O `sourceEntityId`, `sourceComponentName` e `sourceComponentTypeId` fornece informações sobre em qual entidade e componente a propriedade do relacionamento foi definida. `targetEntityId` especifica para qual entidade esse relacionamento está apontando.

- Obtenha todas as entidades com um relacionamento específico com uma entidade específica

```

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

```

Essa consulta retorna todos os nomes de entidades de todas as entidades que têm um `isLocationOf` relacionamento com a `room_0` entidade.

A `MATCH` cláusula: especifica um padrão que corresponde a quaisquer dois nós (`e1,e2`) que tenham uma borda direcionada (`r`).

A cláusula WHERE: especifica um filtro no nome do relacionamento e no nome da entidade de origem.

A SELECT cláusula: retorna o `entityName` campo no e2 nó.

Resposta

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

Em `ColumnDescriptions`, o tipo da coluna é `VALUE` porque `entityName` é uma string.

Uma entidade, `floor_0`, é retornada.

## MATCH

Os seguintes padrões são suportados em uma MATCH cláusula:

- Combine o nó 'b' apontando para o nó 'a':

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

- Combine o nó 'a' apontando para o nó 'b':

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

Não há nenhuma variável vinculada a um relacionamento, supondo que um filtro não precise ser especificado no relacionamento.

- Combine o nó 'a' apontando para o nó 'b' e o nó 'b' apontando para o nó 'a':

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

Isso retornará duas correspondências: uma de 'a' para 'b' e outra de 'b' para 'a', então a recomendação é usar bordas direcionadas sempre que possível.

- O nome do relacionamento também é um rótulo do gráfico de propriedades EntityGraph, então você pode simplesmente especificar o nome do relacionamento após dois pontos (:) em vez de especificar um filtro `rel.relationshipName` na WHERE cláusula.

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

- Encadeamento: os padrões podem ser encadeados para corresponder a vários relacionamentos.

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

- Os padrões de salto variáveis também podem abranger vários nós e bordas:

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

Essa consulta corresponde a qualquer padrão com bordas de saída do nó 'a' em 1 a 5 saltos. Os quantificadores permitidos são:

`{m, n}`- entre m e n repetições

`{m, }`- m ou mais repetições.

FROM:

Um nó de entidade pode conter dados aninhados, como componentes que, por sua vez, contêm mais dados aninhados, como propriedades. Eles podem ser acessados desaninhando o resultado do padrão 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'
```

Acesse campos aninhados pontilhando `.` em uma variável. Uma vírgula (`,`) é usada para desaninhar (ou unir) entidades com os componentes internos e, em seguida, com as propriedades dentro desses componentes. `AS` é usado para vincular uma variável às variáveis não aninhadas para que elas possam ser usadas nas cláusulas `or`. `WHERE` `SELECT` Essa consulta retorna todas as entidades que contêm uma propriedade nomeada `roomFunction` com valor `meeting` em um componente com `id` do tipo de componente `com.example.query.construction.room`.

Para acessar vários campos aninhados de um campo, como vários componentes em uma entidade, use a notação de vírgula para fazer uma junção.

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

**SELECT:**

- Retorne um nó:

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

- Retorne uma borda:

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

- Retorne um 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
```

Formate o nome do campo de saída atribuindo um alias a ele usando `AS`. Aqui, em vez de `propertyValue` como nome da coluna na resposta, `roomfunction` é retornado.

- Retorne os aliases:

```
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%'
```

É altamente recomendável usar aliases para ser explícito, aumentar a legibilidade e evitar ambigüidades em suas consultas.

#### WHERE:

- Os operadores lógicos suportados são ANDNOT, OR e.
- Os operadores de comparação compatíveis são <, <=, >, =>, = e !=.
- Use a IN palavra-chave se quiser especificar várias OR condições no mesmo campo.
- Filtrar em um campo de entidade, componente ou propriedade:

```
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'
```

- Filtre na configuration propriedade. Aqui unit está a chave no mapa de configuração e Celsius é o valor.

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

- Verifique se uma propriedade do mapa contém uma determinada chave e valor:

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

- Verifique se uma propriedade do mapa contém uma determinada chave:

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

- Verifique se uma propriedade da lista contém um determinado valor:

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

- Use a função lower() para comparações sem distinção entre maiúsculas e minúsculas. Por padrão, todas as comparações são sensíveis a maiúsculas e minúsculas.

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

## LIKE:

Útil se você não souber o valor exato de um campo e puder realizar uma pesquisa de texto completo no campo especificado. % representa zero ou mais.

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

- Pesquisa de infixos: %room%
- Pesquisa de prefixos: room%
- Pesquisa de sufixo: %room
- Se você tiver '%' em seus valores, coloque um caractere de escape no LIKE e especifique o caractere de escape com ESCAPE.

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

## DISTINCT:

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

- A palavra-chave DISTINCT elimina duplicatas do resultado final.

DISTINCT não é compatível com tipos de dados complexos.

## CONTAGEM

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

- A COUNT palavra-chave calcula o número de itens em um resultado de consulta.
- COUNT não é compatível com campos complexos aninhados e campos de padrões gráficos.
- COUNT a agregação não é compatível com consultas DISTINCT aninhadas.

Por exemplo, não há suporte para COUNT(DISTINCT e.entityId).

## CAMINHO

As seguintes projeções de padrões são suportadas na consulta usando a projeção de caminho:

- Consultas de salto variável

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

Essa consulta combina e projeta metadados de nós de qualquer padrão com bordas de saída do nó a em 1 a 3 saltos.

- Consultas de salto fixo

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

Essa consulta combina e projeta metadados de entidades e bordas de entrada para b.

- Consultas não direcionadas

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

Essa consulta combina e projeta os metadados dos nós em padrões de 1 salto conectando a e c via 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"
        }
      ]
    }
  ]
}
]
}
}

```

Essa resposta de *PATH* consulta é composta apenas por metadados que identificam todos os nós e bordas de cada caminho/padrão entre a e c via b.

#### LIMITE e COMPENSAÇÃO:

```

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

```

**LIMIT** especifica o número de resultados a serem retornados na consulta e **OFFSET** especifica o número de resultados a serem ignorados.

## LIMIT e maxResults:

O exemplo a seguir mostra uma consulta que retorna 500 resultados no total, mas exibe apenas 50 por vez por chamada de API. Esse padrão pode ser usado quando você precisa limitar a quantidade de resultados exibidos, por exemplo, se você só puder exibir 50 resultados em uma interface de usuário.

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

- A `LIMIT` palavra-chave afeta a consulta e limita as linhas resultantes. Se você precisar controlar o número de resultados retornados por chamada de API sem limitar o número total de resultados retornados, use `LIMIT`.
- `max-results` é um parâmetro opcional para a [ação ExecuteQuery da API](#). `max-results` só se aplica à API e à forma como os resultados são lidos dentro dos limites da consulta acima.

O uso `max-results` em uma consulta permite reduzir o número de resultados exibidos sem limitar o número real de resultados retornados.

A consulta abaixo percorre a próxima página de resultados. Essa consulta usa a chamada de `ExecuteQuery` API para retornar as linhas 51 a 100, onde a próxima página de resultados é especificada pelo `next-token` — nesse caso, o token é: `"H7kyGmvK376L"`

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

- A `next-token` string especifica a próxima página de resultados. Para obter mais informações, consulte a ação [ExecuteQuery](#) da API.

AWS IoT TwinMaker a consulta do gráfico de conhecimento tem os seguintes limites:

Nome do limite	Cota	Ajustável
Tempo limite de execução da consulta	10 segundos	Não
Número máximo de saltos	10	Sim
Número máximo de JOIN selos	20	Sim
Número máximo de campos projetados	20	Sim
Número máximo de expressões condicionais (AND,OR,NOT)	10	Sim
Tamanho máximo de um padrão de LIKE expressão (incluindo curingas e escapes)	20	Sim
Número máximo de itens que podem ser especificados em uma IN cláusula	10	Sim
Valor máximo para OFFSET	3000	Sim
Valor máximo para LIMIT	3000	Sim
Valor máximo para travessias (+) OFFSET LIMIT	3000	Sim

# Sincronização de ativos com o AWS IoT SiteWise

AWS IoT TwinMaker suporta sincronização de ativos (sincronização de ativos) para seus AWS IoT SiteWise ativos e modelos de ativos. Usando o tipo de AWS IoT SiteWise componente, a sincronização de ativos usa AWS IoT SiteWise ativos e modelos de ativos existentes e converte esses recursos em AWS IoT TwinMaker entidades, componentes e tipos de componentes. As seções a seguir explicam como configurar a sincronização de ativos e quais AWS IoT SiteWise ativos e modelos de ativos podem ser sincronizados com seu espaço de AWS IoT TwinMaker trabalho.

## Tópicos

- [Usar a sincronização de ativos com o AWS IoT SiteWise](#)
- [Diferenças entre espaços de trabalho personalizados e padrão](#)
- [Recursos sincronizados de AWS IoT SiteWise](#)
- [Analise o status e os erros de sincronização](#)
- [Excluir uma tarefa de sincronização](#)
- [Limites de sincronização de ativos](#)

## Usar a sincronização de ativos com o AWS IoT SiteWise

Este tópico mostra como ativar e configurar a sincronização AWS IoT SiteWise de ativos. Siga os procedimentos apropriados com base no tipo de espaço de trabalho que você está usando.

### Important

Consulte [the section called “Diferenças entre espaços de trabalho personalizados e padrão”](#) para obter informações sobre as diferenças entre os espaços de trabalho personalizados e padrão.

## Tópicos

- [Usando um espaço de trabalho personalizado](#)
- [Usando o IoT SiteWiseDefaultWorkspace](#)

## Usando um espaço de trabalho personalizado

Revise esses pré-requisitos antes de ativar a sincronização de ativos.

### Pré-requisitos

Antes de usar AWS IoT SiteWise, o seguinte deve ser preenchido:

- Você tem um AWS IoT TwinMaker espaço de trabalho.
- Você tem ativos e modelos de ativos em AWS IoT SiteWise. Para obter mais informações, consulte [Criar modelos de ativos](#).
- Uma função existente do IAM com permissões de leitura para as seguintes AWS IoT SiteWise ações:
  - ListAssets
  - ListAssetModels
  - DescribeAsset
  - DescribeAssetModel
- A função do IAM deve ter as seguintes permissões de gravação para AWS IoT TwinMaker:
  - CreateEntity
  - UpdateEntity
  - DeleteEntity
  - CreateComponentType
  - UpdateComponentType
  - DeleteComponentType
  - ListEntities
  - GetEntity
  - ListComponentTypes

Use a seguinte função do IAM como modelo para a função necessária:

```
// 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": [

```

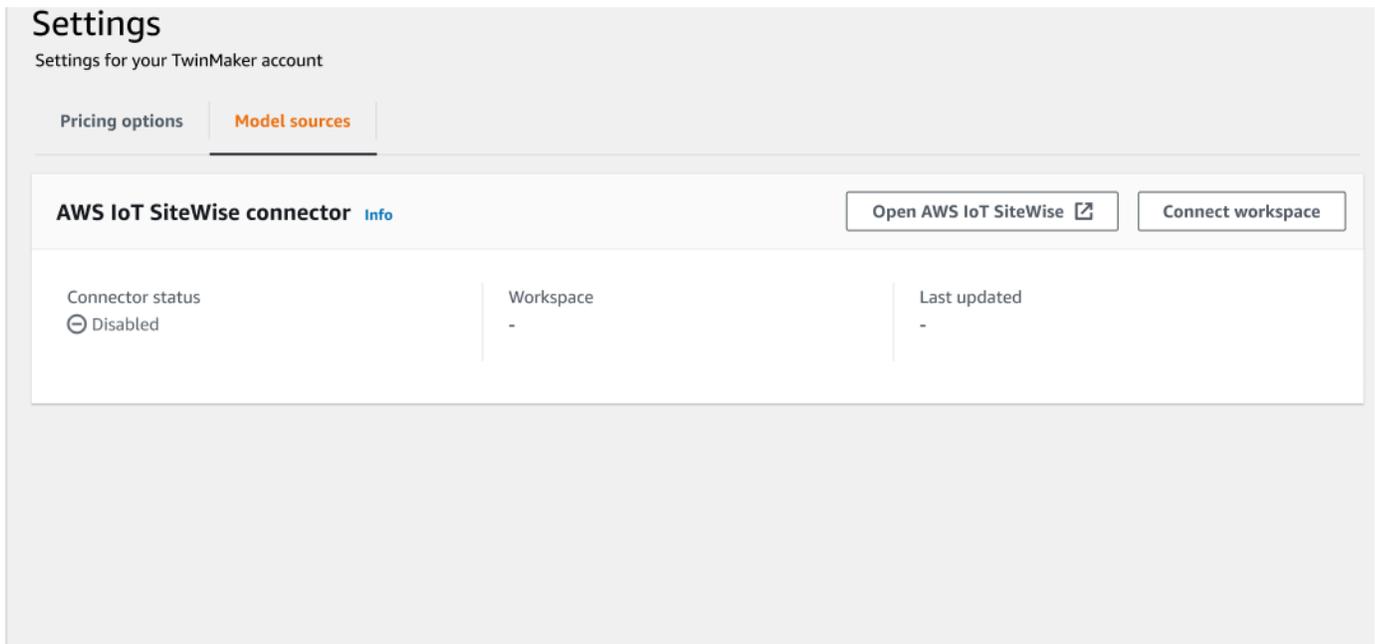
```

        "*"
    ]
},
{
    "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/*"
    ]
}
]
}

```

Use o procedimento a seguir para ativar e configurar a sincronização de ativos AWS IoT SiteWise .

1. No [console do AWS IoT TwinMaker](#), navegue até a página Configurações.
2. Abra a guia Origens do modelo.



**Settings**  
Settings for your TwinMaker account

Pricing options | **Model sources**

**AWS IoT SiteWise connector** [Info](#) Open AWS IoT SiteWise Connect workspace

Connector status	Workspace	Last updated
⊖ Disabled	-	-

3. Escolha Connect workspace para vincular seu AWS IoT TwinMaker espaço de trabalho aos seus AWS IoT SiteWise ativos.

**Note**

Você só pode usar a sincronização de ativos com um único AWS IoT TwinMaker espaço de trabalho. Você deve desconectar a sincronização de um espaço de trabalho e conectar-se a outro espaço de trabalho se quiser sincronizar em um espaço de trabalho diferente.

4. Em seguida, navegue até o espaço de trabalho no qual você deseja usar a sincronização de ativos.
5. Escolha Adicionar origens. Isso abre a página Adicionar origem do modelo de entidade.

AWS IoT TwinMaker > Workspaces > cookieFactory > Add entity model source

## Add entity model source

Add an entity model source to your workspace.

### Add entity model source

Select a source to connect with your AWS IoT TwinMaker workspace. With external sources, you can connect the work you have already configured and import it into this workspace.

AWS IoT SiteWise

This will connect your AWS IoT SiteWise data with this workspace. Descriptive text about what the connector does.

**IAM role**  
This role will be used for XYZ.

Select IAM role

6. Na página Adicionar fonte do modelo de entidade, confirme se o campo de origem é exibido AWS IoT SiteWise. Selecione o perfil do IAM que você criou como pré-requisito para o perfil do IAM.
7. Agora você ativou a sincronização AWS IoT SiteWise de ativos. Você deve ver um banner de confirmação ser exibido na parte superior da página Espaço de trabalho selecionado, confirmando que a sincronização de ativos está ativa. Agora, você também deve ver uma fonte de sincronização listada na seção Origens do modelo de entidade.

cookieFactory Info View ▼ Delete

### Workspace information Edit

<p><b>Name</b> cookieFactory</p> <p><b>Description</b> This is a fully functioning cookie factory workspace.</p>	<p><b>ARN</b> arn:aws:iottwinmaker-us-east-1:2345workspace</p> <p><b>Date created</b> December 17, 2021, 14:32 (UTC+3:30)</p> <p><b>Last modified</b> February 2, 2022, 13:18 (UTC+3:30)</p>	<p><b>S3 resource</b> roci-workspace-myws-348503018462</p> <p><b>Execution role</b> executionRole</p>
--------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------

### Entity model sources (1) Add source

Source	Status	Date last updated
AWS IoT SiteWise	✔ Synced	March 28, 2022, 14:32 (UTC+3:30)

## Usando o IoT Site WiseDefaultWorkspace

Quando você opta pela [AWS IoT SiteWiseAWS IoT TwinMaker integração](#), um espaço de trabalho padrão chamado IoT SiteWiseDefaultWorkspace é criado e sincronizado automaticamente com AWS IoT SiteWise.

Você também pode usar a AWS IoT TwinMaker CreateWorkspace API para criar um espaço de trabalho chamado IoT SiteWiseDefaultWorkspace.

### Pré-requisitos

Antes de criar IoT SiteWiseDefaultWorkspace, certifique-se de ter feito o seguinte:

- Crie uma função AWS IoT TwinMaker vinculada ao serviço. Consulte [Usar perfis vinculados a serviço do AWS IoT TwinMaker](#) para obter mais informações.
- Abra o console do IAM em <https://console.aws.amazon.com/iam/>.

Revise a função ou o usuário e verifique se ele tem permissão para `iotsitewise:EnableSiteWiseIntegration`.

Se necessário, adicione permissão à função ou ao usuário:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "iotsitewise:EnableSiteWiseIntegration",
      "Resource": "*"
    }
  ]
}
```

## Diferenças entre espaços de trabalho personalizados e padrão

### Important

Novos AWS IoT SiteWise recursos, como [CompositionModel](#), só estão disponíveis em `IoTSiteWiseDefaultWorkspace`. Recomendamos que você use um espaço de trabalho padrão em vez de um espaço de trabalho personalizado.

Ao usar o `IoTSiteWiseDefaultWorkspace`, há algumas diferenças notáveis em relação ao uso de um espaço de trabalho personalizado com sincronização de ativos.

- Quando você cria um espaço de trabalho padrão, a localização do Amazon S3 e a função do IAM são opcionais.

### Note

Você pode usar `UpdateWorkspace` para fornecer a localização do Amazon S3 e a função do IAM.

- O `IoTSiteWiseDefaultWorkspace` não tem um limite de contagem de recursos para sincronizar AWS IoT SiteWise recursos AWS IoT TwinMaker.
- Quando você sincroniza recursos de AWS IoT SiteWise, eles `SyncSource` serão `SITWISE_MANAGED`. Isso inclui `Entities ComponentTypes` e.
- Novos AWS IoT SiteWise recursos, como `CompositionModel` os que estão disponíveis apenas no `IoTSiteWiseDefaultWorkspace`.

Existem algumas limitações específicas `IoTSiteWiseDefaultWorkspace`, elas são:

- O espaço de trabalho padrão não pode ser excluído.
- Para excluir recursos, você deve primeiro excluir os AWS IoT SiteWise recursos e, em seguida, os recursos correspondentes serão AWS IoT TwinMaker excluídos.

# Recursos sincronizados de AWS IoT SiteWise

Este tópico lista quais ativos você pode sincronizar AWS IoT SiteWise com seu AWS IoT TwinMaker espaço de trabalho.

## Important

Consulte [Diferenças entre espaços de trabalho personalizados e padrão](#) para obter informações sobre as diferenças entre os espaços de trabalho personalizados e padrão.

## Espaços de trabalho personalizados e padrão

Os seguintes recursos estão sincronizados e estão disponíveis nos espaços de trabalho personalizados e padrão:

### Modelos de ativo

AWS IoT TwinMaker cria um novo tipo de componente para cada modelo de ativo em AWS IoT SiteWise.

- O componente TypeId do modelo de ativos usará um dos seguintes padrões:
  - Espaço de trabalho personalizado - `iotsitewise.assetmodel:assetModelId`
  - Espaço de trabalho padrão - `assetModelId`
- Cada propriedade no modelo de ativo é uma nova propriedade no tipo de componente, com um dos seguintes padrões de nomenclatura:
  - Espaço de trabalho personalizado - `Property_propertyId`
  - Espaço de trabalho padrão - `propertyId`

O nome da propriedade em AWS IoT SiteWise é armazenado como `displayName` na definição da propriedade.

- Cada hierarquia no modelo de ativo é uma nova propriedade do tipo LIST e `nestedType` está RELATIONSHIP no tipo de componente. A hierarquia é mapeada para a propriedade com um nome prefixado por um dos seguintes:
  - Espaço de trabalho personalizado - `Hierarchy_hierarchyId`
  - Espaço de trabalho padrão - `hierarchyId`

## Ativo

AWS IoT TwinMaker cria uma nova entidade para cada ativo em AWS IoT SiteWise.

- O `entityId` é o mesmo que o de `assetId` dentro AWS IoT SiteWise.
- Essas entidades têm um único componente chamado `sitewiseBase`, que tem o tipo de componente correspondente ao modelo de ativo para esse ativo.
- Quaisquer substituições de nível de ativo, como definir um apelido de propriedade ou unidade de medida, são refletidas na entidade no AWS IoT TwinMaker.

## Somente espaço de trabalho padrão

Os seguintes ativos são sincronizados e estão disponíveis somente no espaço de trabalho padrão, `IoTSiteWiseDefaultWorkspace`

### AssetModelComponents

AWS IoT TwinMaker cria um novo tipo de componente para cada `AssetModelComponents` entrada AWS IoT SiteWise.

- O componente `TypeId` do modelo de ativos usa o seguinte padrão: `assetModelId`.
- Cada propriedade no modelo de ativo é uma nova propriedade no tipo de componente, com o nome da propriedade como `propertyId`. O nome da propriedade em AWS IoT SiteWise é armazenado como `displayName` na definição da propriedade.
- Cada hierarquia no modelo de ativo é uma nova propriedade do tipo `LIST` e `nestedType` está `RELATIONSHIP` no tipo de componente. A hierarquia é mapeada para a propriedade com um nome prefixado por `hierarchyId`.

### AssetModelCompositeModel

AWS IoT TwinMaker cria um novo tipo de componente para cada `AssetModelCompositeModel` entrada AWS IoT SiteWise.

- O componente `TypeId` do modelo de ativos usa o seguinte padrão: `assetModelId_assetModelCompositeModelId`.
- Cada propriedade no modelo de ativo é uma nova propriedade no tipo de componente, com o nome da propriedade como `propertyId`. O nome da propriedade em AWS IoT SiteWise é armazenado como `displayName` na definição da propriedade.

## AssetCompositeModels

AWS IoT TwinMaker cria um novo componente composto para cada entrada `AssetCompositeModel`. AWS IoT SiteWise

- O `componentName` é o mesmo que o de `assetModelCompositeModelId` dentro AWS IoT SiteWise.

## Recursos não sincronizados

Os seguintes recursos não são sincronizados:

Ativos e modelos de ativos não sincronizados

- Os modelos de alarme serão sincronizados como `CompositeModels`, mas os dados correspondentes no ativo relacionados aos alarmes não serão sincronizados.
- [AWS IoT SiteWise os fluxos de dados](#) não são sincronizados. Somente propriedades modeladas no modelo de ativo são sincronizadas.
- Os valores de propriedades para atributos, medidas, transformações, agregações e cálculos de metadados, como fórmula e janela, não são sincronizados. Somente os metadados sobre as propriedades, como alias, unidade de medida e tipo de dados, são sincronizados. Os valores podem ser consultados usando a API normal do conector de AWS IoT TwinMaker dados, [GetPropertyValueHistory](#).

## Use entidades sincronizadas e tipos de componentes em AWS IoT TwinMaker

Depois que os ativos são sincronizados AWS IoT SiteWise, os tipos de componentes sincronizados são lidos somente em AWS IoT TwinMaker. Qualquer ação de atualização ou exclusão deve ser feita em AWS IoT SiteWise, e essas alterações serão sincronizadas AWS IoT TwinMaker se o `SyncJob` ainda estiver ativo.

As entidades sincronizadas e o componente AWS IoT SiteWise base também são lidos somente em AWS IoT TwinMaker. Você pode adicionar outros componentes não sincronizados à entidade sincronizada, desde que nenhum atributo no nível da entidade, como a descrição ou `entityName`, seja atualizado.

Algumas restrições se aplicam à forma como você pode interagir com entidades sincronizadas. Você não pode criar entidades secundárias em uma entidade sincronizada na hierarquia da entidade

sincronizada. Além disso, você não pode criar tipos de componentes não sincronizados que se estendam a partir de um tipo de componente sincronizado.

#### Note

Componentes adicionais são excluídos junto com a entidade se o ativo for excluído AWS IoT SiteWise ou se você excluir o trabalho de sincronização.

Você pode usar essas entidades sincronizadas nos painéis do Grafana e adicioná-las como tags no compositor de cenas, como entidades regulares. Você também pode emitir consultas ao gráfico de conhecimento para essas entidades sincronizadas.

#### Note

Entidades sincronizadas sem modificação não são cobradas, mas você será cobrado por essas entidades se tiverem sido feitas alterações no AWS IoT TwinMaker. Por exemplo, se você adicionar um componente não sincronizado a uma entidade sincronizada, essa entidade agora será cobrada. AWS IoT TwinMaker Para obter mais informações, consulte [Preços do AWS IoT TwinMaker](#).

## Analise o status e os erros de sincronização

Este tópico fornece orientação sobre como analisar erros e status de sincronização.

#### Important

Consulte [the section called “Diferenças entre espaços de trabalho personalizados e padrão”](#) para obter informações sobre as diferenças entre os espaços de trabalho personalizados e padrão.

## Sincronizar status de tarefa

Um trabalho de sincronização tem um dos status a seguir, dependendo de seu estado.

- O CREATING estado da tarefa de sincronização significa que a tarefa está verificando as permissões e carregando dados AWS IoT SiteWise para preparar a sincronização.

- O **INITIALIZING** estado da tarefa de sincronização significa que todos os recursos existentes AWS IoT SiteWise estão sincronizados com o AWS IoT TwinMaker. Essa etapa pode levar mais tempo para ser concluída se o usuário tiver um grande número de ativos e modelos de ativos no AWS IoT SiteWise. Você pode monitorar o número de recursos que foram sincronizados verificando a tarefa de sincronização no [console do AWS IoT TwinMaker](#) ou chamando a API `ListSyncResources`.
- O estado da tarefa de sincronização **ACTIVE** significa que a etapa de inicialização foi concluída. A tarefa agora está pronta para sincronizar qualquer nova atualização do AWS IoT SiteWise.
- O estado da tarefa de sincronização **ERROR** indica um erro com qualquer um dos estados anteriores. Examine a mensagem de erro. Pode haver um problema com a configuração do perfil do IAM. Se você quiser usar uma nova função do IAM, exclua a tarefa de sincronização que teve o erro e crie uma nova com a nova função.

Os erros de sincronização aparecem na página de origem do modelo, que é acessada a partir da tabela **Origens do modelo de entidade** no seu espaço de trabalho. A página de origem do modelo exibe uma lista de recursos que falharam na sincronização. A maioria dos erros é repetida automaticamente pela tarefa de sincronização, mas, se o recurso exigir uma ação, ele permanecerá no estado **ERROR**. Você também pode obter uma lista de erros usando a [ListSyncResourcesAPI](#).

Para exibir todos os erros listados para a fonte atual, use o procedimento a seguir.

1. Navegue até seu espaço de trabalho no [console do AWS IoT TwinMaker](#).
2. Selecione a AWS IoT SiteWise fonte listada no modal **Fontes do modelo de entidade** para abrir a página de detalhes da sincronização de ativos.

AWS IoT TwinMaker > Workspaces > SWSync > Source: AWS IoT SiteWise

## AWS IoT SiteWise source Disconnect

### Overview

Data Source AWS IoT SiteWise	Status ACTIVE	Date created January 20, 1970 at 02:23:23 (UTC-5:00)
Role syncRole	Status reason -	Last modified January 20, 1970 at 02:23:23 (UTC-5:00)
Total resources 8	In Sync 6	Error 2

### Errors (2)

Find resources

Resource name	External id	Status	Status reason
e8a7fff4-289c-4b28-8814-6dc3e5a13612	e8a7fff4-289c-4b28-8814-6dc3e5a13612	ERROR	{"code":"SYNC_INITIALIZING_ERROR","message":"SYNC INITIALIZING ERROR"}
18fd0d54-a268-4558-b40a-34c3f7af9228	18fd0d54-a268-4558-b40a-34c3f7af9228	ERROR	{"code":"SYNC_INITIALIZING_ERROR","message":"SYNC INITIALIZING ERROR"}

3. Conforme mostrado na captura de tela anterior, todos os recursos com erros persistentes são listados na tabela Erros. Você pode usar essa tabela para rastrear e corrigir erros relacionados a recursos específicos.

Possíveis erros incluem:

- Embora AWS IoT SiteWise ofereça suporte a nomes de ativos duplicados, AWS IoT TwinMaker só os suporta no ROOT nível, não na mesma entidade principal. Se você tiver dois ativos com o mesmo nome em uma entidade principal em AWS IoT SiteWise, um deles falhará ao sincronizar. Para corrigir esse erro, exclua um dos ativos ou mova um para outro ativo principal AWS IoT SiteWise antes de sincronizar.
- Se você já tiver uma entidade com o mesmo ID do AWS IoT SiteWise ativo, esse ativo não será sincronizado até que você exclua a entidade existente.

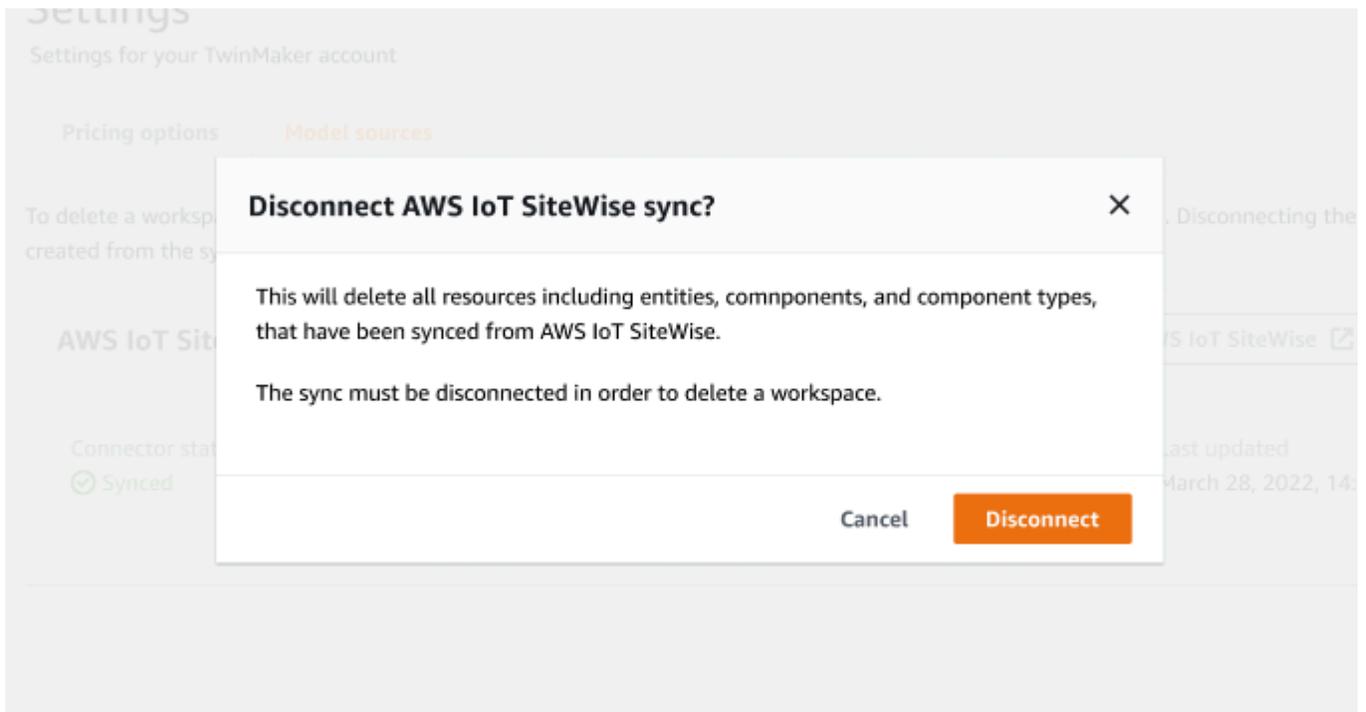
## Excluir uma tarefa de sincronização

Use o procedimento a seguir para excluir uma tarefa de sincronização.

**⚠ Important**

Consulte [the section called “Diferenças entre espaços de trabalho personalizados e padrão”](#) para obter informações sobre as diferenças entre os espaços de trabalho personalizados e padrão.

1. Navegue até o [console do AWS IoT TwinMaker](#).
2. Abra o espaço de trabalho do qual você deseja excluir o trabalho de sincronização.
3. Em Origens do modelo de entidade, selecione a fonte AWS IoT SiteWise para abrir a página de detalhes da fonte.
4. Para interromper o trabalho de sincronização, escolha Desconectar. Confirme sua escolha de excluir totalmente a tarefa de sincronização.



Depois que uma tarefa de sincronização for excluída, você poderá criá-la novamente no mesmo espaço de trabalho ou em um espaço de trabalho diferente.

Você não pode excluir um espaço de trabalho se houver algum trabalho de sincronização nesse espaço de trabalho. Exclua as tarefas de sincronização antes de excluir um espaço de trabalho.

Se houver algum erro durante a exclusão da tarefa de sincronização, o trabalho de sincronização permanecerá no estado DELETING e será repetido automaticamente. Agora você pode excluir manualmente qualquer uma das entidades sincronizadas ou tipos de componentes se houver algum erro relacionado à exclusão de um recurso.

 Note

Todos os recursos que foram sincronizados AWS IoT SiteWise são excluídos primeiro e, em seguida, o trabalho de sincronização em si é excluído.

## Limites de sincronização de ativos

 Important

Consulte [the section called “Diferenças entre espaços de trabalho personalizados e padrão”](#) para obter informações sobre as diferenças entre os espaços de trabalho personalizados e padrão.

Como as [cotas AWS IoT SiteWise](#) são maiores do que as [cotas AWS IoT TwinMaker](#) padrão, estamos aumentando os limites a seguir para entidades e tipos de componentes sincronizados a partir de AWS IoT SiteWise.

- 1000 tipos de componentes sincronizados em um espaço de trabalho, já que ele só pode sincronizar 1000 modelos de ativos de. AWS IoT SiteWise
- 100.000 entidades sincronizadas em um espaço de trabalho, já que ele só pode sincronizar 100.000 ativos de. AWS IoT SiteWise
- Máximo de 2.000 entidades secundárias por entidade primária. Sincroniza 2.000 ativos secundários por ativo primário.

 Note

A [GetEntity](#)API retorna somente as primeiras 50 entidades secundárias de uma propriedade hierárquica, mas você pode usar a [GetPropertyValue](#)API para paginar e recuperar a lista de todas as entidades secundárias.

- 600 propriedades por componente sincronizado AWS IoT SiteWise, que podem sincronizar modelos de ativos com 600 propriedades e hierarquias totais.

 Note

Esses limites são aplicáveis somente às entidades sincronizadas. Solicite um aumento de cota se precisar aumentar esses limites para recursos não sincronizados.

# Integração do painel do Grafana AWS IoT TwinMaker

AWS IoT TwinMaker suporta a integração com o Grafana por meio de um plug-in de aplicativo. Use o Grafana versão 10.4.0 e versões posteriores para interagir com seu aplicativo digital duplo. O AWS IoT TwinMaker plug-in fornece painéis personalizados, modelos de painel e uma fonte de dados para se conectar aos dados de seus gêmeos digitais.

Para obter mais informações sobre como se integrar ao Grafana e configurar permissões para o seu painel, consulte os seguintes tópicos:

## Tópicos

- [Configuração CORS para o visualizador de cenas do Grafana](#)
- [Como configurar seu ambiente do Grafana](#)
- [Como criar um perfil do IAM para o painel](#)
- [Criação de uma política AWS IoT TwinMaker de reprodutor de vídeo](#)

### Note

Você precisa modificar a configuração CORS (compartilhamento de recursos de origem cruzada) do bucket Amazon S3 para permitir que a interface de usuário do Grafana carregue recursos do bucket. Para obter instruções, consulte [Configuração CORS para o visualizador de cenas do Grafana](#).

Para obter mais informações sobre o plug-in AWS IoT TwinMaker Grafana, consulte a documentação do [AWS IoT TwinMaker aplicativo](#).

Para obter mais informações sobre os principais componentes do complemento do Grafana, consulte:

- [AWS IoT TwinMaker datasource](#)
- [Modelos de painel](#)
- [Painel Visualizador de cenas](#)
- [Painel do reprodutor de vídeo](#)

# Configuração CORS para o visualizador de cenas do Grafana

O plug-in AWS IoT TwinMaker Grafana requer uma configuração CORS (compartilhamento de recursos de origem cruzada), que permite que a interface de usuário do Grafana carregue recursos do bucket do Amazon S3. Sem a configuração do CORS, você receberá uma mensagem de erro como “Falha no carregamento da cena 3D com falha na rede” no visualizador de cena, pois o domínio Grafana não pode acessar os recursos no bucket do Amazon S3.

Para configurar seu bucket Amazon S3 com CORS, use as seguintes etapas:

1. Faça login no console IAM e abra o [console Amazon S3](#).
2. Na lista Buckets, escolha o nome do bucket que você usa como bucket de recursos do seu AWS IoT TwinMaker espaço de trabalho.
3. Escolha Permissões.
4. Na seção Compartilhamento de recursos de origem cruzada selecione Editar para abrir o editor CORS.
5. Na caixa de texto do editor de configuração CORS, digite ou copie e cole a seguinte configuração JSON CORS substituindo o domínio da área de trabalho do Grafana pelo seu domínio. *GRAFANA-WORKSPACE-DOMAIN*

## Note

Você precisa manter o caractere \* com asterisco no início do elemento "AllowedOrigins": JSON.

```
{
  [
    "AllowedHeaders": [
      "*"
    ],
    "AllowedMethods": [
      "GET",
      "PUT",
      "POST",
      "DELETE",
      "HEAD"
    ]
  ]
}
```

```
    ],
    "AllowedOrigins": [
      "*GRAFANA-WORKSPACE-DOMAIN"
    ],
    "ExposeHeaders": [
      "ETag"
    ]
  }
]
```

6. Selecione Salvar alterações para finalizar a configuração do CORS.

Para obter mais informações sobre CORS com buckets do Amazon S3, [consulte Uso do compartilhamento de recursos de origem cruzada \(CORS\)](#).

## Como configurar seu ambiente do Grafana

Você pode usar o Amazon Managed Grafana para um serviço totalmente gerenciado ou configurar um ambiente do Grafana que você mesmo gerencie. Com o Amazon Managed Grafana, você pode rapidamente implantar, operar e escalar o Grafana de código aberto de acordo com suas necessidades. Também há a opção de configurar sua própria infraestrutura para gerenciar os servidores do Grafana.

Para obter mais informações sobre as duas opções do ambiente Grafana, consulte os seguintes tópicos:

- [Amazon Managed Grafana](#)
- [Grafana autogerenciado](#)

## Amazon Managed Grafana

O Amazon Managed Grafana fornece um AWS IoT TwinMaker plug-in para que você possa se integrar rapidamente ao AWS IoT TwinMaker Grafana. Como o Amazon Managed Grafana gerencia os servidores do Grafana para você, você pode visualizar seus dados sem precisar criar, empacotar ou implantar qualquer hardware ou qualquer outra infraestrutura do Grafana. Para obter mais informações sobre o Amazon Managed Grafana, consulte [O que é Amazon Managed Grafana?](#).

**Note**

Atualmente, o Amazon Managed Grafana oferece suporte à versão 1.3.1 do plug-in AWS IoT TwinMaker Grafana.

## Pré-requisitos Amazon Managed Grafana

Para usar AWS IoT TwinMaker em um painel do Amazon Managed Grafana, primeiro preencha os seguintes pré-requisitos:

- Crie um AWS IoT TwinMaker espaço de trabalho. Para obter mais informações sobre a criação de espaços de trabalho, consulte [Introdução ao AWS IoT TwinMaker](#).

**Note**

Quando você cria pela primeira vez um espaço de trabalho Amazon Managed Grafana no AWS Management Console, AWS IoT TwinMaker não está listado. No entanto, o complemento já está instalado em todos os espaços de trabalho. Você pode encontrar o complemento AWS IoT TwinMaker na lista de complementos Grafana de código aberto. Você pode encontrar a datasource AWS IoT TwinMaker escolhendo Adicionar uma fonte de dados na página Datasources.

Quando você cria um espaço de trabalho Amazon Managed Grafana, um perfil do IAM é criado automaticamente para gerenciar as permissões para a instância do Grafana. Isso é chamado de perfil do IAM do espaço de trabalho. É a opção do provedor de autenticação que você usará para configurar todas as datasources AWS IoT TwinMaker do Grafana. O Amazon Managed Grafana não suporta a adição automática de permissões para AWS IoT TwinMaker, portanto, você deve configurar essas permissões manualmente. Para obter mais informações sobre como definir as permissões manuais, consulte [Como criar um perfil do IAM para o painel](#).

## Grafana autogerenciado

Você pode optar por host sua própria infraestrutura para executar o Grafana. Para obter informações sobre como executar o Grafana localmente em sua máquina, consulte [Instalar o Grafana](#). O complemento AWS IoT TwinMaker está disponível no catálogo público do Grafana. Para obter

informações sobre como instalar esse complemento em seu ambiente Grafana, consulte [Aplicativo AWS IoT TwinMaker](#).

Quando você executa o Grafana localmente, não é possível compartilhar painéis com facilidade ou fornecer acesso a vários usuários. Para obter um guia de início rápido com script sobre o compartilhamento de painéis usando o Grafana local, consulte o [repositório de exemplos AWS IoT TwinMaker](#). Este recurso orienta você na hospedagem de um ambiente Grafana no Cloud9 e na Amazon em um endpoint público. EC2

Você deve determinar qual provedor de autenticação você usará para configurar as fontes de TwinMaker dados. Você configura as credenciais para o ambiente com base na cadeia de credenciais padrão (consulte [Como usar a cadeia de fornecedores de credenciais padrão](#)). As credenciais padrão podem ser as credenciais permanentes de qualquer usuário ou função. Por exemplo, se você estiver executando o Grafana na Amazon EC2, a cadeia de credenciais padrão tem acesso à [função de EC2 execução da Amazon](#), que seria então seu provedor de autenticação. O nome do recurso da Amazon (ARN) do IAM do provedor de autenticação é obrigatório nas etapas do [Como criar um perfil do IAM para o painel](#).

## Como criar um perfil do IAM para o painel

Com AWS IoT TwinMaker, você pode controlar o acesso aos dados em seus painéis da Grafana. Os usuários do painel do Grafana devem ter diferentes escopos de permissão para visualizar dados e, em alguns casos, gravar dados. Por exemplo, um operador de alarme pode não ter permissão para ver vídeos, enquanto um administrador tem permissão para todos os recursos. Grafana define as permissões por meio de datasources, nas quais as credenciais e um perfil do IAM são fornecidos. A AWS IoT TwinMaker fonte de dados busca AWS credenciais com permissões para essa função. Se uma função do IAM não for fornecida, a Grafana usa o escopo das credenciais, que não pode ser reduzido em. AWS IoT TwinMaker

Para usar seus AWS IoT TwinMaker painéis no Grafana, você cria uma função do IAM e anexa políticas. Os modelos a seguir podem ser usados para ajudar a criar essas políticas.

## Criar uma política do IAM

Crie uma política do IAM chamada *YourWorkspaceId*DashboardPolicy no console do IAM. Essa política dá aos seus espaços de trabalho acesso ao bucket e aos recursos AWS IoT TwinMaker do Amazon S3. Você também pode decidir usar o [AWS IoT Greengrass Edge Connector para o Amazon Kinesis Video](#) Streams, o que requer permissões para o Kinesis AWS IoT SiteWise Video Streams

e ativos configurados para o componente. Para se adequar ao seu caso de uso, escolha um dos modelos de política a seguir.

## 1. Nenhuma política de permissões de vídeo

Se você não quiser usar o [painel do reprodutor de vídeo](#) Grafana, crie a política usando o modelo a seguir.

```
{
  "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": "*"
    }
  ]
}
```

Um bucket do Amazon S3 é criado para cada espaço de trabalho. Ele contém os modelos e cenas 3D para visualizar em um painel. O [SceneViewer](#) painel carrega itens desse compartimento.

## 2. Política de permissões de vídeo com escopo reduzido

Para limitar o acesso ao painel do Video Player no Grafana, agrupe seus recursos do AWS IoT Greengrass Edge Connector para Amazon Kinesis Video Streams por tags. Para obter mais informações sobre a redução de escopo de permissões para seus recursos de vídeo, consulte [Criação de uma política AWS IoT TwinMaker de reprodutor de vídeo](#).

## 3. Todas as permissões de vídeo

Se você não quiser agrupar seus vídeos, poderá torná-los acessíveis a partir do reprodutor de vídeo do Grafana. Qualquer pessoa com acesso a um espaço de trabalho da Grafana pode reproduzir vídeo para qualquer stream em sua conta e ter acesso somente de leitura a qualquer ativo. AWS IoT SiteWise Isso inclui todos os recursos criados no futuro.

Crie a política com o modelo a seguir:

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

```

Este modelo de política fornece as seguintes permissões:

- Acesso somente leitura a um bucket do S3 para carregar uma cena.
- Acesso somente AWS IoT TwinMaker para leitura a todas as entidades e componentes em um espaço de trabalho.
- Acesso somente leitura para transmitir todos os vídeos do Kinesis Video Streams em sua conta.
- Acesso somente para leitura ao histórico do valor da propriedade de todos os AWS IoT SiteWise ativos em sua conta.

- Ingestão de dados em qualquer propriedade de um AWS IoT SiteWise ativo marcado com a chave `EdgeConnectorForKVS` e o valor `workspaceId`.

## Marcando seu AWS IoT SiteWise ativo de câmera, solicite o upload de vídeo do Edge

Usando o reprodutor de vídeo no Grafana, os usuários podem solicitar manualmente que o vídeo seja carregado do cache de Edge para o Kinesis Video Streams. Você pode ativar esse recurso para qualquer AWS IoT SiteWise ativo associado ao seu AWS IoT Greengrass Edge Connector para Amazon Kinesis Video Streams e que esteja marcado com a chave `EdgeConnectorForKVS`

O valor da tag pode ser uma lista de `workspaceIDs` delimitada por qualquer um dos seguintes caracteres: `.` `:` `+` `=` `@` `_` `/` `-`. Por exemplo, se você quiser usar um AWS IoT SiteWise ativo associado a um AWS IoT Greengrass Edge Connector para Amazon Kinesis Video Streams em todos os espaços de trabalho, você pode usar uma tag que segue esse padrão: `WorkspaceA/WorkspaceB/WorkspaceC`. O plug-in Grafana exige que o AWS IoT TwinMaker `WorkspaceID` seja usado para agrupar a ingestão de dados de ativos. AWS IoT SiteWise

## Adicione mais permissões à sua política de painel

O plug-in AWS IoT TwinMaker Grafana usa seu provedor de autenticação para chamar a função `AssumeRole` de painel que você cria. Internamente, o plug-in restringe o maior escopo de permissões às quais você tem acesso usando uma política de sessão na `AssumeRole` chamada. Para obter mais informações sobre as políticas de sessão, consulte [Políticas de Sessão](#).

Essa é a política máxima de permissividade que você pode ter em sua função de painel para um espaço de trabalho AWS IoT TwinMaker :

```
{
  "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*"
    }
  }
}

```

```
}  
]  
}
```

Se você adicionar declarações com Allow mais permissões, elas não funcionarão no AWS IoT TwinMaker plug-in. Isso ocorre intencionalmente para garantir que as permissões mínimas necessárias sejam usadas pelo complemento.

No entanto, você pode reduzir ainda mais as permissões. Para ter mais informações, consulte [Criação de uma política AWS IoT TwinMaker de reprodutor de vídeo](#).

## Como criar o perfil do IAM do painel do Grafana

No console do IAM, crie um perfil do IAM chamado *YourWorkspaceId*DashboardRole. Anexe o *YourWorkspaceId*DashboardPolicy à função.

Para editar a política de confiança da função do painel, é necessário conceder permissão para que o provedor de autenticação do Grafana chame AssumeRole na função do painel. Atualize a política de confiança com o seguinte modelo:

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Principal": {  
        "AWS": "ARN of Grafana authentication provider"  
      },  
      "Action": "sts:AssumeRole"  
    }  
  ]  
}
```

Para obter mais informações sobre como criar um ambiente Grafana e encontrar seu provedor de autenticação, consulte [Como configurar seu ambiente do Grafana](#).

# Criação de uma política AWS IoT TwinMaker de reprodutor de vídeo

A seguir está um modelo de política com todas as permissões de vídeo necessárias para o complemento AWS IoT TwinMaker do Grafana:

```
{
  "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 obter mais informações sobre a política completa, consulte o modelo de política de permissões para todos os vídeos no tópico [Criar uma política do IAM](#).

## Diminua o acesso aos seus recursos

O painel Video Player no Grafana chama diretamente o Kinesis Video Streams e o SiteWise IoT para fornecer uma experiência completa de reprodução de vídeo. Para evitar o acesso não autorizado a recursos que não estão associados ao seu AWS IoT TwinMaker espaço de trabalho, adicione condições à política do IAM para sua função no painel do espaço de trabalho.

## Diminua o escopo das permissões GET

Você pode reduzir o acesso aos seus Amazon Kinesis Video AWS IoT SiteWise Streams e ativos marcando recursos. Talvez você já tenha marcado seu recurso de AWS IoT SiteWise câmera com base no AWS IoT TwinMaker WorkspaceID para ativar o recurso de solicitação de upload de vídeo. Consulte o tópico [Carregar vídeo do Edge](#). Você pode usar o mesmo par de chave-valor de tag para limitar o acesso GET aos AWS IoT SiteWise ativos e também para marcar seus Kinesis Video Streams da mesma forma.

Em seguida, você pode adicionar essa condição às declarações kinesisisvideo e iotsitewise em *YourWorkspaceId*DashboardPolicy:

```
"Condition": {
  "StringLike": {
    "aws:ResourceTag/EdgeConnectorForKVS": "*workspaceId*"
  }
}
```

## Caso de uso na vida real: agrupamento de câmeras

Neste cenário, você tem uma grande variedade de câmeras monitorando o processo de assar cookies em uma fábrica. Lotes de massa de cookie são feitos na sala de massa, a massa é congelada na sala do freezer e os cookies são assados na sala de cozimento. Há câmeras em cada uma dessas salas com diferentes equipes de operadores monitorando separadamente cada processo. Você quer que cada grupo de operadores seja autorizado para sua respectiva sala. Ao criar um gêmeo digital para a fábrica de cookies, um único espaço de trabalho é usado, mas as permissões da câmera precisam ser definidas por sala.

Você pode conseguir essa separação de permissões marcando grupos de câmeras com base em seu groupingID. Nesse cenário, os GroupingIDs são BatterRoom, e. FreezerRoom BakingRoom A câmera em cada sala está conectada ao Kinesis Video Streams e deve ter uma tag com: chave =EdgeConnectorForKVS, valor = BatterRoom. O valor pode ser uma lista de agrupamentos delimitados por um destes caracteres: . : + = @ \_ / -

Para alterar o *YourWorkspaceId*DashboardPolicy, use as seguintes declarações 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": "*groupId*"
    }
  }
},
...

```

Essas declarações restringem a reprodução de streaming de vídeo e o acesso ao histórico de AWS IoT SiteWise propriedades a recursos específicos em um agrupamento. O *groupId* é definido pelo seu caso de uso. No nosso cenário, seria o RoomID.

## AWS IoT SiteWise BatchPutAssetPropertyValue Permissão para reduzir o escopo

Fornecer essa permissão ativa o [atributo de solicitação de carregamento de vídeo no reprodutor de vídeo](#). Ao enviar um vídeo, você pode especificar um intervalo de tempo e enviar a solicitação escolhendo Enviar no painel do Grafana.

Para conceder BatchPutAssetPropertyValue permissões ao iotsitewise:, use a política padrão:

```

...,
{
  "Effect": "Allow",
  "Action": [
    "iotsitewise:BatchPutAssetPropertyValue"
  ],
  "Resource": "*",
  "Condition": {
    "StringLike": {
      "aws:ResourceTag/EdgeConnectorForKVS": "*workspaceId*"
    }
  }
},
...

```

Ao usar essa política, os usuários podem ligar `BatchPutAssetPropertyValue` para qualquer propriedade no ativo da AWS IoT SiteWise câmera. Você pode restringir a autorização para um `PropertyID` específico especificando-o na condição da declaração.

```
{
  ...
  "Condition": {
    "StringEquals": {
      "iotsitewise:propertyId": "propertyId"
    }
  }
  ...
}
```

O painel Video Player no Grafana ingere dados na propriedade de medição, chamada `VideoUploadRequest`, para iniciar o upload do vídeo do cache de borda para o Kinesis Video Streams. Encontre o `propertyID` dessa propriedade no console. AWS IoT SiteWise Para alterar o `YourWorkspaceIdDashboardPolicy`, use a seguinte declaração de política:

```
...,
{
  "Effect": "Allow",
  "Action": [
    "iotsitewise:BatchPutAssetPropertyValue"
  ],
  "Resource": "*",
  "Condition": {
    "StringLike": {
      "aws:ResourceTag/EdgeConnectorForKVS": "*workspaceId"
    },
    "StringEquals": {
      "iotsitewise:propertyId": "VideoUploadRequestPropertyId"
    }
  }
},
...
```

Essa declaração restringe a ingestão de dados a uma propriedade específica do seu ativo AWS IoT SiteWise de câmera marcado. Para obter mais informações, consulte [Como AWS IoT SiteWise funciona com IAM](#).

# Conecte AWS IoT SiteWise os alarmes aos painéis da AWS IoT TwinMaker Grafana

## Note

Este atributo está em versão prévia pública e está sujeito a alterações.

AWS IoT TwinMaker é capaz de importar alarmes AWS IoT SiteWise e eventos em AWS IoT TwinMaker componentes. Isso permite que você possa consultar o status do alarme e configurar os limites do alarme sem implementar um conector de dados personalizado para migração de AWS IoT SiteWise dados. Você pode usar o plug-in AWS IoT TwinMaker Grafana para visualizar o status do alarme e configurar o limite do alarme no Grafana, sem fazer chamadas AWS IoT TwinMaker de API ou interagir diretamente com os alarmes. AWS IoT SiteWise

## AWS IoT SiteWise pré-requisitos de configuração de alarme

Antes de criar alarmes e integrá-los ao painel do Grafana, certifique-se de ter revisado os seguintes pré-requisitos:

- Familiarize-se com AWS IoT SiteWise o modelo e o sistema de ativos da. Para obter mais informações, consulte [Criação de modelos](#) de ativos e [Criação de ativos](#) no Guia AWS IoT SiteWise do usuário.
- Familiarize-se com os modelos de alarme do IoT Events e saiba como anexá-los a AWS IoT SiteWise um modelo. Para obter mais informações, consulte [Definindo alarmes de AWS IoT eventos](#) no Guia do AWS IoT SiteWise usuário.
- Integre AWS IoT TwinMaker com o Grafana para que você possa acessar seus AWS IoT TwinMaker recursos no Grafana. Para obter mais informações, consulte, [Integração do painel do Grafana AWS IoT TwinMaker](#).

## Defina a função do IAM do componente de AWS IoT SiteWise alarme

AWS IoT TwinMaker usa a função do IAM do espaço de trabalho para consultar e configurar o limite de alarme no Grafana. As seguintes permissões são necessárias na função AWS IoT TwinMaker de espaço de trabalho para interagir com os AWS IoT SiteWise alarmes na Grafana:

```
{
  "Effect": "Allow",
  "Action": [
    "iotevents:DescribeAlarmModel",
  ],
  "Resource": ["{IoTEventsAlarmModelArn}"],
}, {
  "Effect": "Allow",
  "Action": [
    "iotsitewise:BatchPutAssetPropertyValue"
  ],
  "Resource": ["{IoTSitewiseAssetArn}"]
}
```

No [AWS IoT TwinMaker console](#), crie uma entidade que represente seu AWS IoT SiteWise ativo. Certifique-se de adicionar um componente para essa entidade usando `amazon.iotsitewise.alarm` como tipo de componente e escolher os modelos de ativos e alarmes correspondentes.

## Add component

### Component information

Name

DustAlarm

Type

Types of components include documents, time-series data, structured data, and unstructured data.

com.amazon.iotsitewise.alarm

Asset Model

Choose an asset model.

ConstructionSpot

Asset

Choose an asset.

Spot1

Alarm Model

Choose an alarm model.

ConstructionSpotDustAlarm

A captura de tela acima é um exemplo de criação dessa entidade com o `tipocom.amazon.iotsitewise.alarm`.

Quando você cria esse componente, importa AWS IoT TwinMaker automaticamente as propriedades de alarme relacionadas de AWS IoT SiteWise AWS IoT Events e. Você pode então repetir esse padrão de tipo de componente de alarme para criar componentes de alarme para todos os ativos necessários em seu espaço de trabalho.

## Consulte e atualize por meio da AWS IoT TwinMaker API

Depois de criar componentes de alarme, você pode consultar o status e o limite do alarme e atualizar os limites do alarme por meio da AWS IoT TwinMaker API.

Abaixo está um exemplo de solicitação para consultar o status do alarme:

```
aws iottwinmaker get-property-value-history --cli-input-json \  
{  
  "workspaceId": "{workspaceId}",  
  "entityId": "{entityId}",  
  "componentName": "{componentName}",
```

```

    "selectedProperties": ["alarm_status"],
    "startTime": "{startTimeIsoString}",
    "endTime": "{endTimeIsoString}"
  }'

```

Abaixo está um exemplo de solicitação para consultar o limite de alarme.

```

aws iottwinmaker get-property-value-history --cli-input-json \
'{
  "workspaceId": "{workspaceId}",
  "entityId": "{entityId}",
  "componentName": "{componentName}",
  "selectedProperties": ["alarm_threshold"],
  "startTime": "{startTimeIsoString}",
  "endTime": "{endTimeIsoString}"
}'

```

Abaixo está um exemplo de solicitação para atualizar o limite de alarme:

```

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}"
        }
      ]
    }
  ]
}'

```

## Configure seu painel Grafana para alarmes

Uma segunda função do IAM do painel habilitada para gravação precisa ser criada, que é uma função normal, mas com permissão para que a ação `iottwinmaker:BatchPutPropertyValues` seja adicionada ao TwinMaker espaço de trabalho arn, como no exemplo abaixo.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iottwinmaker:Get*",
        "iottwinmaker:List*",
        "iottwinmaker:BatchPutPropertyValues"
      ],
      "Resource": [
        "{workspaceArn}",
        "{workspaceArn}/*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": "iottwinmaker:ListWorkspaces",
      "Resource": "*"
    }
  ]
}
```

Como alternativa, você pode adicionar essa declaração ao final do seu perfil do IAM:

```
{
  "Effect": "Allow",
  "Action": [
    "iottwinmaker:BatchPutPropertyValues"
  ],
  "Resource": [
    "{workspaceArn}",
    "{workspaceArn}/*"
  ]
}
```

A fonte de dados precisa ter seu campo de gravação definido com o perfil de gravação do painel criado por você.

Depois de modificar sua função do IAM, faça login no painel do Grafana para assumir a função atualizada arn. Selecione a caixa de seleção Definir permissões de gravação para o Painel de Configuração de Alarmes e copie no arn para a função Gravação.

The screenshot shows the Grafana Settings page for a data source named "AWS IoT TwinMaker-4". The page is divided into several sections:

- Settings:** Name: AWS IoT TwinMaker-4, Default:
- Connection Details:**
  - Authentication Provider: AWS SDK Default
  - Assume Role ARN: arn:aws:iam:\*
  - External ID: External ID
  - Endpoint: https://{service}.{region}.amazonaws.com
  - Default Region: us-east-1
- Assume Role ARN:** A warning box with a triangle icon and text: "Specify an IAM role to narrow the permission scope of this datasource. Follow the documentation here to create policies and a role with minimal permissions for your TwinMaker workspace."
- TwinMaker settings:**
  - Workspace: enter workspace ID
  - Define write permissions for Alarm Configuration Panel
  - Assume Role ARN Write: arn:aws:iam:\*
- Buttons:** Back, Explore, Delete, Save & test

The "Assume Role ARN Write" field, containing the value "arn:aws:iam:\*", is circled in red in the original image.

## Use o painel Grafana para a visualização de alarmes

Use o procedimento a seguir para adicionar um painel de configuração de alarme ao seu painel e configurá-lo:

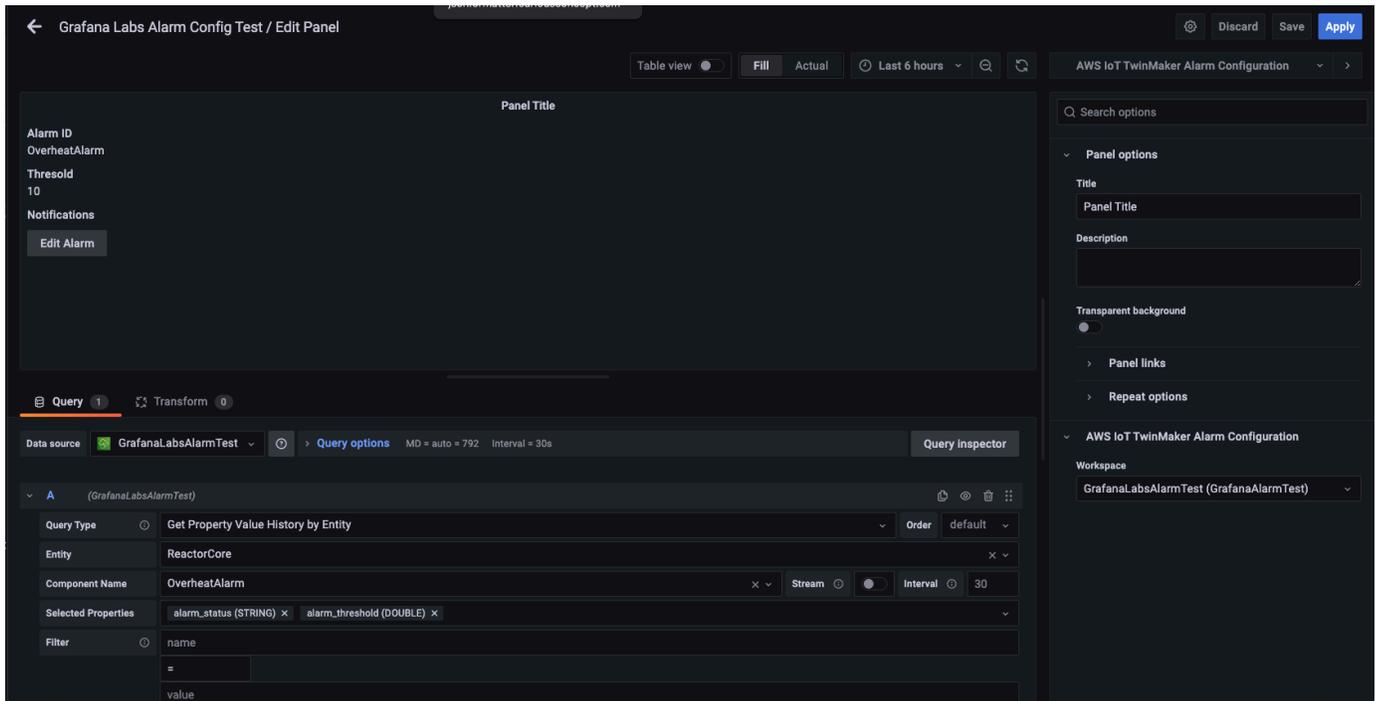
1. Selecione a área de trabalho nas opções do painel.
2. Defina sua fonte de dados na configuração da consulta.
3. Use o seguinte tipo de consulta: `Get Property Value History by Entity`.
4. Selecione uma entidade ou variável de entidade à qual você deseja adicionar um alarme.
5. Depois de selecionar a entidade, selecione um componente ou variável do componente à qual aplicar uma propriedade.
6. Para a propriedade, escolha: `alarm_status` e `alarm_threshold`.

Quando estiver conectado, você deverá ver o ID do alarme e seu limite atual.

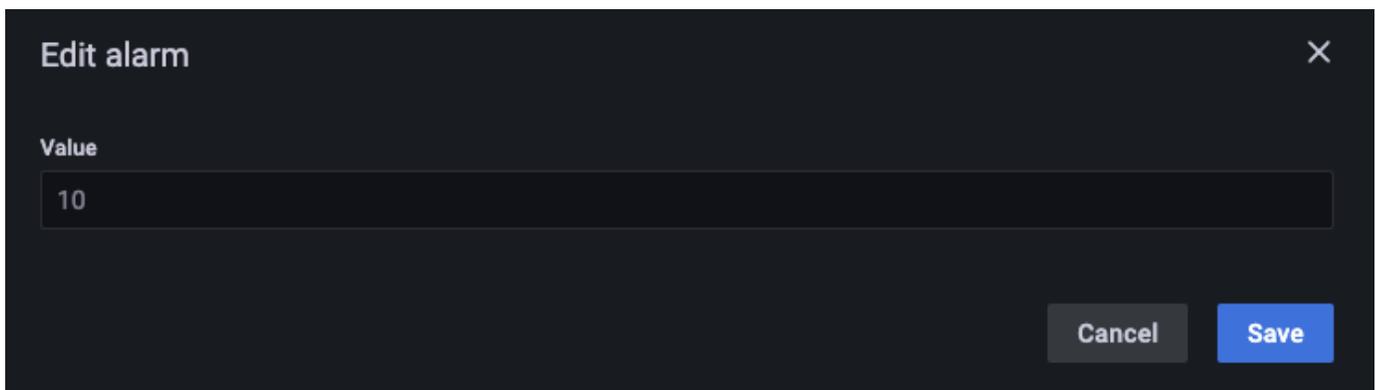
### Note

Para a pré-visualização pública, nenhuma notificação é exibida. Você deve revisar o status e o limite do alarme para garantir que as propriedades foram aplicadas corretamente.

7. A ordem ascendente de consulta padrão deve ser usada para que o valor mais recente seja exibido.
8. A seção de filtro da consulta pode ser deixada vazia. Uma configuração completa é mostrada abaixo:



9. Ao usar o botão Editar alarme, você pode abrir uma caixa de diálogo para alterar o limite atual do alarme.
10. Selecione Salvar para definir o novo valor limite.



### Note

Este painel deve ser usado apenas com um intervalo de tempo ao vivo que inclua o presente. Usá-lo com intervalos de tempo que terminam e iniciam no passado pode mostrar valores inesperados ao editar os limites de alarme sempre como o limite atual.

# AWS IoT TwinMaker Integração com Matterport

O Matterport fornece uma variedade de opções de captura para a digitalização de ambientes do mundo real e criação de modelos 3D imersivos, também conhecidos como gêmeos digitais Matterport. Esses modelos são chamados de espaços Matterport. O AWS IoT TwinMaker é compatível com integração com o Matterport, permitindo que você importe seus gêmeos digitais do Matterport para suas cenas AWS IoT TwinMaker. Ao emparelhar gêmeos digitais Matterport com AWS IoT TwinMaker, você pode visualizar e monitorar seu sistema de gêmeos digitais em um ambiente virtual.



Para obter mais informações sobre como usar o Matterport, leia a documentação do Matterport na página [AWS IoT TwinMaker e Matterport](#).

## Tópicos de integração

- [Visão geral da integração](#)
- [Pré-requisitos da integração com o Matterport](#)
- [Gere e registre suas credenciais do Matterport](#)
- [Armazene suas credenciais da Matterport em AWS Secrets Manager](#)
- [Importe espaços do Matterport para cenas AWS IoT TwinMaker](#)

- [Use espaços do Matterport em seu painel do Grafana AWS IoT TwinMaker](#)
- [Use espaços Matterport em seu aplicativo web AWS IoT TwinMaker](#)

## Visão geral da integração

Essa integração permite que você:

- Use suas tags e espaços Matterport no kit do AWS IoT TwinMaker aplicativo.
- Visualize seus dados importados do matterport no painel do Grafana AWS IoT TwinMaker . Para obter mais informações sobre como usar o AWS IoT TwinMaker Grafana, leia a documentação de integração do painel do [Grafana](#).
- Importe seus espaços do Matterport para suas AWS IoT TwinMaker cenas.
- Selecione e importe suas tags Matterport que você gostaria de vincular aos dados em sua cena. AWS IoT TwinMaker
- Mostre automaticamente seu espaço no Matterport, marque as alterações em sua AWS IoT TwinMaker cena e aprove quais sincronizar.

O processo de integração é composto por três etapas críticas.

1. [Gere e registre suas credenciais do Matterport](#)
2. [Armazene suas credenciais da Matterport em AWS Secrets Manager](#)
3. [Importe espaços do Matterport para cenas AWS IoT TwinMaker](#)

Você inicia sua integração no [console do AWS IoT TwinMaker](#). Na página Configurações do console, em Recursos de terceiros, abra Integração com o Matterport para navegar entre os diferentes recursos necessários para a integração.

The screenshot shows the AWS IoT TwinMaker console interface. The top navigation bar includes the AWS logo, 'Services', 'Resource Groups', a search icon, a notification bell, 'Customer Account', 'N. Virginia', and 'Support'. The left sidebar contains navigation options: 'AWS IoT TwinMaker', 'How it works', 'Workspaces', 'Workspace', 'Component types', 'Entities', 'Resource library', 'Scenes', 'Query editor', 'Settings', 'What's new', 'Documentation', 'FAQ', and 'Pricing'. The main content area is titled 'Settings' and 'Settings for your AWS IoT TwinMaker account'. It has three tabs: 'Pricing options', 'Model sources', and '3rd party resources'. Below the tabs, a heading reads '3rd party software integration. You can configure AWS IoT TwinMaker to work with 3rd party software. This pages lists which software is available for integration'. A section titled '▼ Matterport integration (1) Info' contains a 'How it works' section with four steps:
 

- Step 1. Contact Matterport**: In order to integrate your Matterport assets into AWS IoT TwinMaker you need to contact matterport directly to request an integration with AWS IoT TwinMaker. We recommend that you reach out to your point of contact, or request a contact [here](#) so that they can provide guidance on the necessary components for enabling the integration. A 'Contact Matterport' button is provided.
- Step 2. Record your Matterport SDK credentials**: Record the client id and client secret for your Private Model Embed (PME) application and the SDK key from your Matterport account. A 'Go to Matterport' button is provided.
- Step 3. Add your Matterport credentials into AWS secrets manager.**: Add your Matterport credentials as a secret in the AWS Secrets Manager. When adding you secret in AWS Secrets Manager, you will have to add a tag with key 'AWSIoT TwinMaker\_Matterport' to this secret since it's a 3rd party key. An 'AWS Secret Manager' button is provided.
- Step 4. Select your Matterport account in a scene composer scene.**: Add Matterport scans to your scene, by selecting the connected Matterport account from within the scene settings page. A 'Go to Workspaces' button is provided.

 Below the steps is a 'Connected accounts' section with a table with columns 'Name', 'Description', and 'ARN'. The table is empty, and a message states 'No connections. Copy you Matterport SDK credentials key into AWS Secret Manager.' with an 'AWS Secret Manager' button.

## Pré-requisitos da integração com o Matterport

Antes de integrar o Matterport com AWS IoT TwinMaker , certifique-se de atender aos seguintes pré-requisitos:

- Você adquiriu uma conta Matterport de nível corporativo e os produtos [Matterport](#) necessários para a integração. AWS IoT TwinMaker
- Você tem um AWS IoT TwinMaker espaço de trabalho. Para obter mais informações, consulte [Introdução ao AWS IoT TwinMaker](#).

- Você atualizou sua função no AWS IoT TwinMaker espaço de trabalho. Para obter mais informações sobre como criar um perfil de serviço do espaço de trabalho, consulte [Criar e gerenciar uma função de serviço para AWS IoT TwinMaker](#).

Adicione o conteúdo a seguir ao seu perfil do espaço de trabalho:

```
{
  "Effect": "Allow",
  "Action": "secretsmanager:GetSecretValue",
  "Resource": [
    "AWS Secrets Manager secret ARN"
  ]
}
```

- Você deve entrar em contato com o Matterport para configurar o licenciamento necessário para permitir a integração. O Matterport também habilitará um Private Model Embed (PME) para a integração.

Se você já tem um gerente de contas Matterport, entre em contato com ele diretamente.

Use o procedimento a seguir para entrar em contato com o Matterport e solicitar uma integração caso não tenha um ponto de contato com o Matterport:

1. Abra a página [Matterport e AWS IoT TwinMaker](#).
2. Pressione o botão Fale conosco para abrir o formulário de contato.
3. Preencha as informações necessárias no formulário.
4. Quando estiver pronto, escolha DIGA OLÁ para enviar sua solicitação para o Matterport.

Depois de solicitar a integração, você poderá gerar as credenciais necessárias do SDK do Matterport e do Private Model Embed (PME) necessárias para dar continuidade ao processo de integração.

 Note

Isso pode envolver a cobrança de uma taxa pela compra de novos produtos ou serviços.

## Gere e registre suas credenciais do Matterport

Para integrar o Matterport com AWS IoT TwinMaker, você deve fornecer AWS Secrets Manager as credenciais do Matterport. Use o procedimento a seguir para gerar as credenciais do SDK do Matterport.

1. Faça login na sua conta [Matterport](#).
2. Navegue até a página de configurações da sua conta.
3. Uma vez na página de configurações, selecione a opção Ferramentas do desenvolvedor.
4. Na página Ferramentas do desenvolvedor, acesse a seção Gerenciamento de chaves do SDK.
5. Na seção Gerenciamento de chaves do SDK, selecione a opção para adicionar uma nova chave do SDK.
6. Depois de ter a chave do Matterport SDK, adicione domínios à chave e ao AWS IoT TwinMaker seu servidor Grafana. Se você estiver usando o kit de AWS IoT TwinMaker aplicativos, certifique-se de adicionar também seu domínio personalizado.
7. Então, localize a seção Gerenciamento de integração de aplicativos. Nela, você deve ver seu aplicativo PME listado. Registre as seguintes informações:
  - O ID do cliente
  - O Segredo do cliente

### Note

Como o Segredo do cliente é apresentado a você apenas uma vez, recomendamos que você registre o Segredo do cliente. Você deve apresentar seu Segredo do cliente no console do AWS Secrets Manager para dar continuidade à integração com o Matterport.

Essas credenciais são criadas automaticamente quando você adquire os componentes necessários e o PME da sua conta é ativado pelo Matterport. Se essas credenciais não forem exibidas, entre em contato com a Matterport. Para solicitar contato, consulte o [Matterport e o formulário de contato AWS IoT TwinMaker](#).

Para obter mais informações sobre as credenciais do SDK do Matterport, consulte a documentação oficial do SDK do Matterport em [isãõ geral da documentação do SDK](#).

# Armazene suas credenciais da Matterport em AWS Secrets Manager

Use o procedimento a seguir para armazenar suas credenciais do Matterport em AWS Secrets Manager

## Note

O Segredo do cliente, criado a partir do procedimento no tópico [Gere e registre suas credenciais do Matterport](#), é necessário para dar continuidade à integração com o Matterport.

1. Faça login no AWS Secrets Manager console.
2. Navegue até a página Segredos e selecione Armazenar um novo segredo.
3. Para o Tipo de segredo, selecione Outro tipo de segredo.
4. Na seção Pares de chave/valor, adicione os seguintes pares de chave-valor, usando suas credenciais do Matterport como valores:
  - Crie um par de chave-valor, com Chave: `application_key` e Valor: *<suas credenciais do Matterport>*.
  - Crie um par de chave-valor, com Chave: `client_id` e Valor: *<suas credenciais do Matterport>*.
  - Crie um par de chave-valor, com Chave: `client_secret` e Valor: *<suas credenciais do Matterport>*.

Ao concluir, você deverá ter uma configuração semelhante à do seguinte exemplo:

Key/value	Plaintext	
application_key	matterport_application_key	Remove
client_id	matterport_oauth_app_client_id	Remove
client_secret	matterport_oauth_app_client_secret	Remove
<a href="#">+ Add row</a>		

5. Em Chave de criptografia, você pode deixar a chave de criptografia padrão `aws/secretsmanager` selecionada.
6. Escolha **Próximo** para ir para a página **Configurar segredo**.
7. Preencha o campo para o Nome do segredo e a Descrição.
8. Adicione uma tag a esse segredo na seção **Tags**.

Ao criar a tag, atribua a chave `AWSIoTtwinMaker_Matterport` conforme mostrado na captura de tela a seguir:

AWS Secrets Manager > Secrets > Store a new secret

Step 1  
Choose secret type

Step 2  
**Configure secret**

Step 3  
Configure rotation - optional

Step 4  
Review

## Configure secret

### Secret name and description [Info](#)

**Secret name**  
A descriptive name that helps you find your secret later.

Secret name must contain only alphanumeric characters and the characters /\_+@-

**Description - optional**

Maximum 250 characters.

### Tags - optional

Key	Value - optional	
<input type="text" value="AWSIoTtwinMaker_Matterport"/>	<input type="text" value="Enter value"/>	<input type="button" value="Remove"/>
<input type="button" value="Add"/>		

 Note

É necessário adicionar uma tag. Tags são obrigatórias ao adicionar segredos de terceiros ao AWS Secrets Manager, mesmo que as Tags sejam listadas como opcionais.

O campo Valor é opcional. Depois de fornecer uma Chave, você pode selecionar Adicionar para passar para a próxima etapa.

- Escolha Próximo para ir para a página Configurar alternância. Configurar uma alternância de segredos é opcional. Se quiser terminar de adicionar seu segredo e não precisar de uma rotação, escolha Próximo novamente. Para obter mais informações sobre rotação de segredos, consulte [Rotacionar AWS Secrets Manager segredos](#).
- Confirme a configuração do seu segredo na página Revisão. Quando estiver pronto para adicionar seu segredo, escolha Loja.

Para obter mais informações sobre o uso AWS Secrets Manager, consulte a seguinte AWS Secrets Manager documentação:

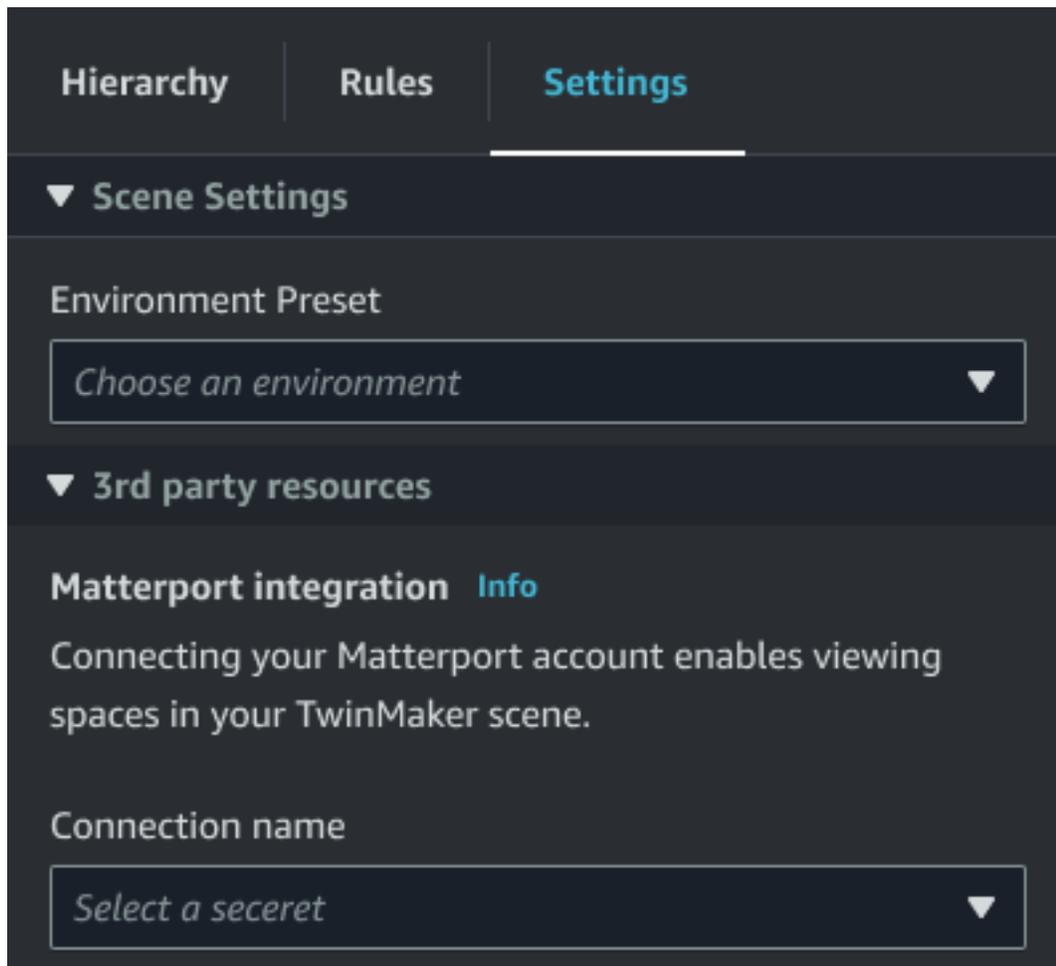
- [Crie e gerencie segredos com AWS Secrets Manager](#)
- [O que é AWS Secrets Manager?](#)
- [Gire segredos AWS Secrets Manager](#)

Agora você está pronto para importar seus ativos do Matterport para AWS IoT TwinMaker cenas. Consulte o procedimento na seção a seguir, [Importe espaços do Matterport para cenas AWS IoT TwinMaker](#)

## Importe espaços do Matterport para cenas AWS IoT TwinMaker

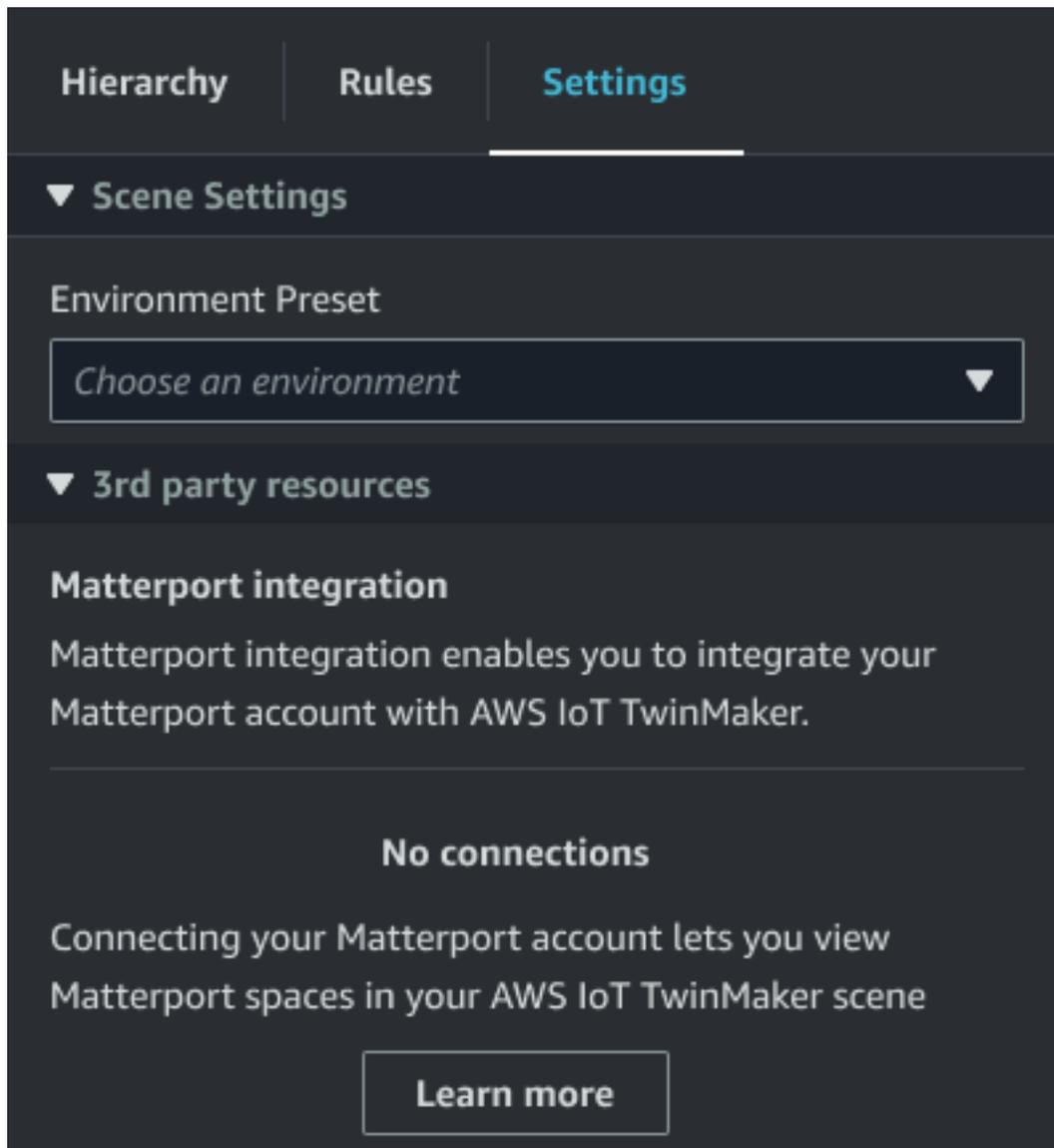
Adicione digitalizações do Matterport à sua cena selecionando a conta conectada do Matterport na página de configurações de cena. Utilize o procedimento a seguir para importar suas digitalizações e tags do Matterport:

1. Fazer login no [console do AWS IoT TwinMaker](#).
2. Crie ou abra uma AWS IoT TwinMaker cena existente na qual você deseja usar um espaço Matterport.
3. Depois que a cena for aberta, navegue até a guia Configurações.
4. Em Configurações, em Recursos de terceiros, localize Nome da conexão e insira o segredo que você criou no formulário de procedimento do [Armazene suas credenciais da Matterport em AWS Secrets Manager](#).

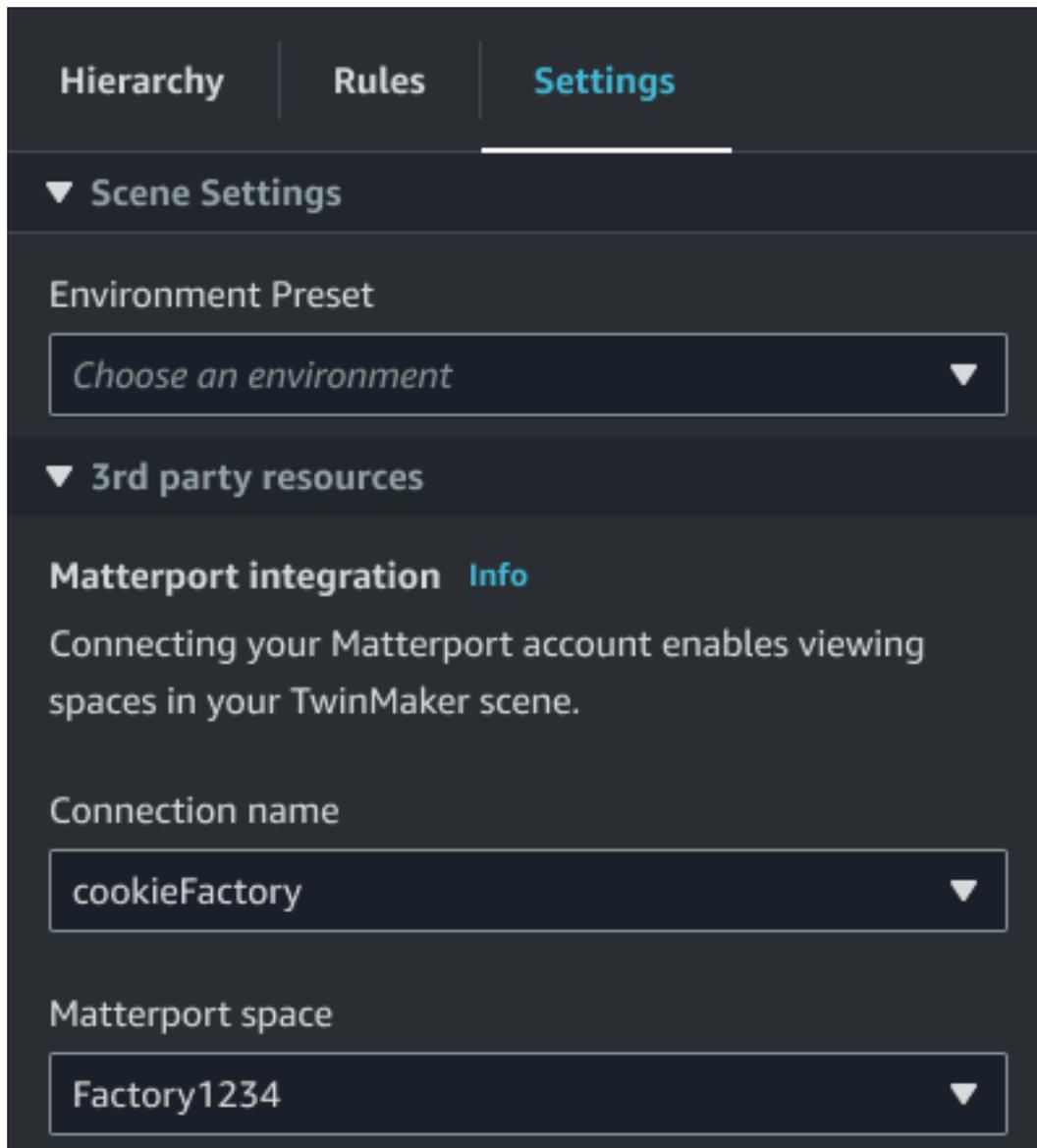


**Note**

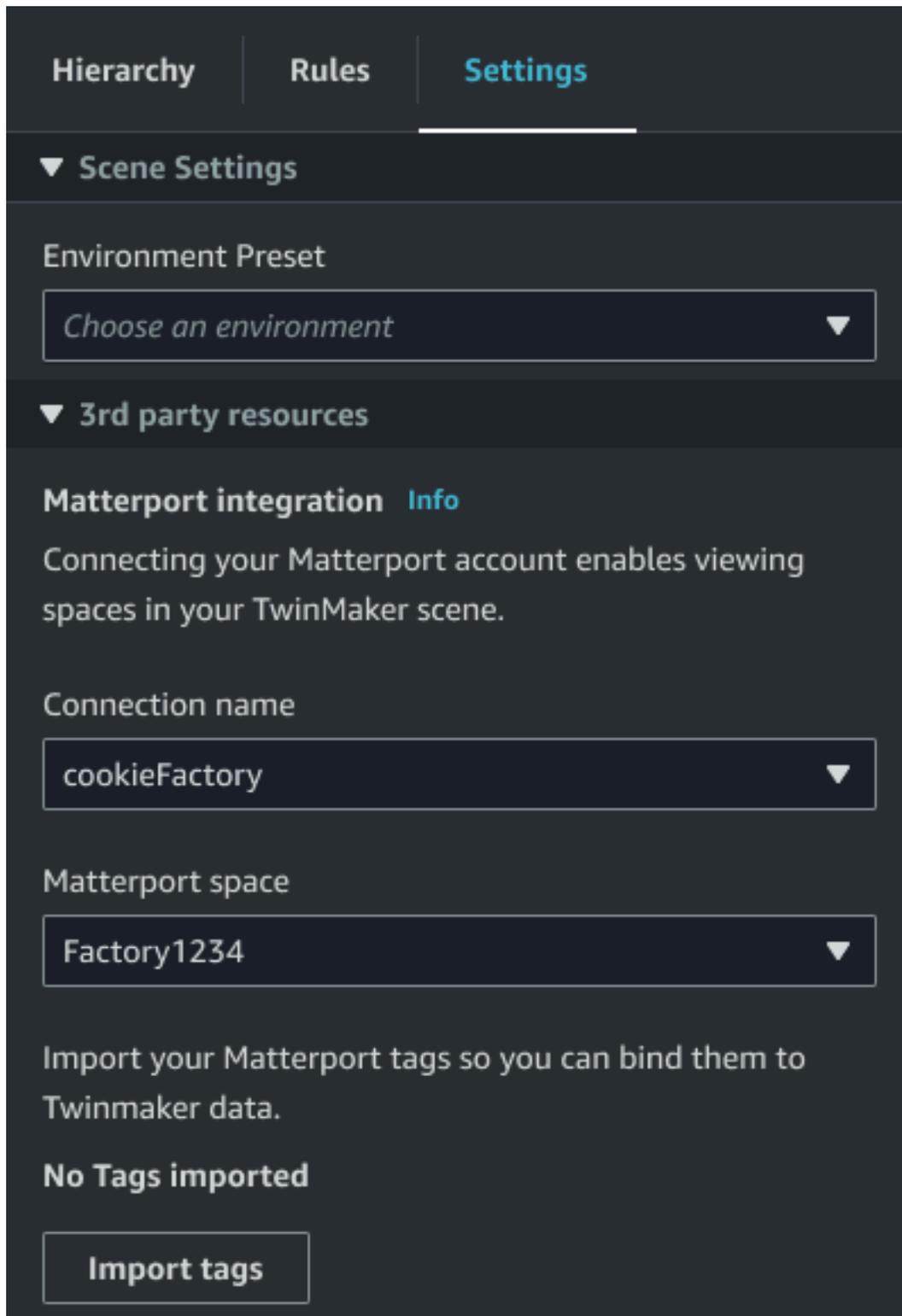
Caso seja uma mensagem informando Conexões indisponíveis, navegue até a página de configurações do [console do AWS IoT TwinMaker](#) para iniciar o processo de integração com o Matterport.



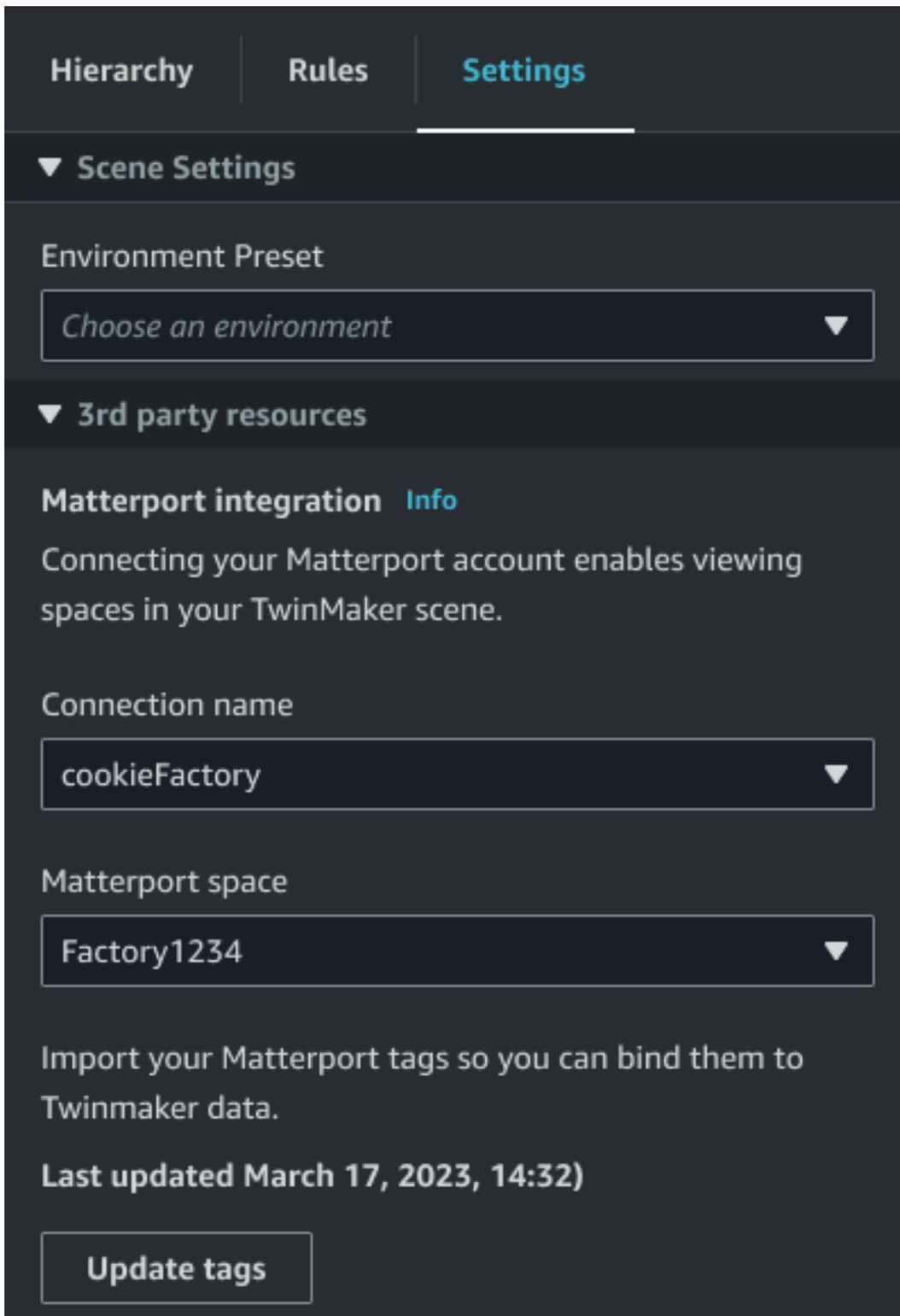
5. Em seguida, escolha o espaço Matterport que você gostaria de usar em sua cena selecionando-o no menu suspenso do espaço Matterport.



6. Depois de selecionar um espaço, você pode importar suas tags Matterport e convertê-las em tags de AWS IoT TwinMaker cerna pressionando o botão Importar tags.



Depois de importar as tags do Matterport, o botão é substituído pelo botão Atualizar tags. Você pode atualizar continuamente suas tags Matterport AWS IoT TwinMaker para que elas sempre reflitam as alterações mais recentes em sua conta Matterport.



7. Você se integrou AWS IoT TwinMaker com sucesso ao Matterport e agora sua AWS IoT TwinMaker cena tem seu espaço e tags importados do Matterport. Você pode trabalhar nessa cena como faria com qualquer outra AWS IoT TwinMaker cena.

Para obter mais informações sobre como trabalhar com AWS IoT TwinMaker cenas, consulte [Criação e edição de AWS IoT TwinMaker cenas](#).

## Use espaços do Matterport em seu painel do Grafana AWS IoT TwinMaker

Depois de importar seu espaço do Matterport para uma AWS IoT TwinMaker cena, você pode visualizar essa cena com o espaço do Matterport no painel do Grafana. Se você já configurou o Grafana com AWS IoT TwinMaker, basta abrir o painel do Grafana para ver sua cena com o espaço importado do Matterport.

Se você ainda não configurou AWS IoT TwinMaker com o Grafana, conclua primeiro o processo de integração do Grafana. Você tem duas opções ao se integrar ao AWS IoT TwinMaker Grafana. Você pode usar uma instância autogerenciada do Grafana ou usar o Amazon Managed Grafana.

Consulte a documentação a seguir para saber mais sobre as opções e o processo de integração do Grafana:

- AWS IoT TwinMaker Integração do [painel Grafana](#).
- [Amazon Managed Grafana](#).
- [Grafana autogerenciado](#).

## Use espaços Matterport em seu aplicativo web AWS IoT TwinMaker

Depois de importar o espaço do Matterport para uma AWS IoT TwinMaker cena, você pode visualizar essa cena com o espaço do Matterport no AWS IoT aplicativo web do kit de aplicativos.

Consulte a documentação a seguir para saber mais sobre como usar o kit de AWS IoT aplicativos:

- Para saber mais sobre como usar AWS IoT TwinMaker com o kit de AWS IoT aplicativos, consulte [Crie um aplicativo web personalizado usando componentes de AWS IoT TwinMaker interface do usuário](#).
- Para saber mais sobre como usar o kit de AWS IoT aplicativos, visite a página do [Github do kit de AWS IoT aplicativos](#).
- Para obter instruções sobre como iniciar um novo aplicativo web usando o kit de AWS IoT aplicativos, visite a página oficial de documentação do [IoT App Kit](#).

# AWS IoT TwinMaker integração de vídeo

Câmeras de vídeo apresentam uma boa oportunidade para a simulação de gêmeo digital. Você pode usar o AWS IoT TwinMaker para simular a localização e as condições físicas da sua câmera. Crie entidades AWS IoT TwinMaker para suas câmeras locais e use componentes de vídeo para transmitir vídeo ao vivo e metadados do seu site para sua AWS IoT TwinMaker cena ou para um painel da Grafana.

AWS IoT TwinMaker pode capturar vídeo de dispositivos de ponta de duas maneiras. Você pode transmitir vídeo de dispositivos de borda com o conector de borda para transmissão de vídeo na transmissão de vídeo do Kinesis ou salvar vídeos no dispositivo de borda e iniciar o carregamento de vídeo com mensagens MQTT. Use esse componente para transmitir dados de vídeo de seus dispositivos para uso com AWS IoT serviços. Para gerar os recursos necessários e implantar o conector de borda para o Kinesis Video Streams, [consulte Introdução ao conector de borda para o stream de vídeo do Kinesis](#) em. GitHub Para obter mais informações sobre o AWS IoT Greengrass componente, consulte a AWS IoT Greengrass documentação sobre o [conector de borda para Kinesis Video Streams](#).

Depois de criar os AWS IoT SiteWise modelos necessários e configurar o componente Greengrass do Kinesis Video Streams, você pode transmitir ou gravar vídeo na borda para seu aplicativo digital duplo no console. AWS IoT TwinMaker Você também pode ver transmissões ao vivo e metadados de seus dispositivos em um painel do Grafana. Para obter mais informações sobre a integração do Grafana AWS IoT TwinMaker e, consulte. [Integração do painel do Grafana AWS IoT TwinMaker](#)

## Use o conector edge para o stream de vídeo do Kinesis para transmitir vídeo em AWS IoT TwinMaker

Com o conector edge para transmissão de vídeo do Kinesis, você pode transmitir vídeo e dados para uma entidade em sua AWS IoT TwinMaker cena. Você usa um componente de vídeo para fazer isso. Para criar o componente de vídeo para uso nas suas cenas, realize o procedimento a seguir.

### Pré-requisitos

Antes de criar o componente de vídeo em sua AWS IoT TwinMaker cena, verifique se você preencheu os seguintes pré-requisitos.

- Criou os AWS IoT SiteWise modelos e ativos necessários para o conector de borda para o stream de vídeo do Kinesis. Para obter mais informações sobre a criação dos ativos do AWS IoT SiteWise para o conector, consulte [Introdução ao conector de borda para transmissão de vídeo do Kinesis](#).
- Implantou o conector Kinesis Video Stream Edge em seu dispositivo. AWS IoT Greengrass Para obter mais informações sobre a implantação do componente Conector de borda para transmissão de vídeo do Kinesis, consulte o [README](#) de implantação.

## Crie componentes de vídeo para AWS IoT TwinMaker cenas

Conclua as etapas a seguir para criar o conector de borda para transmissão de vídeo do Kinesis para a sua cena.

1. No AWS IoT TwinMaker console, abra a cena à qual você deseja adicionar o componente de vídeo.
2. Depois que a cena for aberta, escolha uma entidade existente ou crie a entidade à qual você deseja adicionar o componente e, em seguida, escolha Adicionar componente.
3. No painel Adicionar componente, insira um nome para o componente e, em Tipo, escolha `com.amazon.iotsitewise.connector.edgevideo`.
4. Escolha um modelo de ativo selecionando o nome do modelo de AWS IoT SiteWise câmera que você criou. Esse nome deve ter o seguinte formato: `EdgeConnectorForKVSCameraModel-0abc`, em que a sequência de letras e números no final corresponda ao nome do seu próprio ativo.
5. Em Ativo, escolha os ativos da AWS IoT SiteWise câmera dos quais você deseja transmitir vídeo. Uma pequena janela aparece mostrando uma prévia da transmissão de vídeo atual.

### Note

Para testar sua transmissão de vídeo, escolha Testar. Esse teste envia um evento MQTT para iniciar a transmissão de vídeo ao vivo. Aguarde alguns instantes para que o vídeo apareça no reproduzidor.

6. Para adicionar o componente de vídeo à sua entidade, escolha Adicionar componente.

# Adicione vídeo e metadados da transmissão de vídeo do Kinesis a um painel do Grafana

Depois de criar um componente de vídeo para sua entidade em sua AWS IoT TwinMaker cena, você pode configurar o painel de vídeo no Grafana para ver as transmissões ao vivo. Verifique se você se integrou adequadamente AWS IoT TwinMaker ao Grafana. Para obter mais informações, consulte [Integração do painel do Grafana AWS IoT TwinMaker](#).

## Important

Para ver o vídeo em seu painel do Grafana, você deve garantir que as fontes de dados do Grafana tenham as permissões adequadas do IAM. Para criar o perfil e a política necessária, consulte [Como criar um perfil do IAM para o painel](#).

Conclua as etapas a seguir para ver o Kinesis Video Streams e os metadados no seu painel do Grafana.

1. Abra o AWS IoT TwinMaker painel.
2. Escolha Adicionar painel e, em seguida, escolha Adicionar um painel vazio.

## Note

Para Grafana v10.4, o player de AWS IoT TwinMaker vídeo é encontrado em Widget. Selecione Adicionar >> Widget.

3. Na lista de painéis, escolha o painel do reprodutor de vídeo AWS IoT TwinMaker .
4. No painel do player de AWS IoT TwinMaker vídeo, insira o nome do stream do KinesisVideoStreamName, com o nome do stream de vídeo do Kinesis do qual você deseja transmitir o vídeo.

## Note

Para transmitir metadados para o painel de vídeo Grafana, você deve primeiro ter criado uma entidade com um componente de transmissão de vídeo.

5. Opcional: Para transmitir metadados dos AWS IoT SiteWise ativos para o player de vídeo, em Entidade, escolha a AWS IoT TwinMaker entidade que você criou na sua AWS IoT TwinMaker cena. Para o Nome do componente, escolha o componente de vídeo que você criou para a entidade na sua cena AWS IoT TwinMaker .

# Como usar a biblioteca Flink AWS IoT TwinMaker

AWS IoT TwinMaker fornece uma biblioteca Flink que você pode usar para ler e gravar dados em armazenamentos de dados externos usados em seus gêmeos digitais.

Você usa a biblioteca Flink AWS IoT TwinMaker instalando-a como um conector personalizado no Managed Service for Apache Flink e executando consultas SQL do Flink em um notebook Zeppelin no Managed Service for Apache Flink. O notebook pode ser promovido a um aplicativo de processamento de stream em execução contínua. A biblioteca utiliza componentes AWS IoT TwinMaker para recuperar dados do seu espaço de trabalho.

A biblioteca Flink AWS IoT TwinMaker exige o seguinte.

## Pré-requisitos

1. Um espaço de trabalho totalmente preenchido com cenas e componentes. Use os tipos de componentes integrados para dados de serviços AWS (AWS IoT SiteWise e Kinesis Video Streams). Crie tipos de componentes personalizados para dados de fontes de terceiros. Para ter mais informações, consulte [???](#).
2. Uma compreensão dos notebooks Studio com Managed Service for Apache Flink para Apache Flink. Esses notebooks são alimentados pelo [Apache Zeppelin](#) e usam a estrutura [Apache Flink](#). Para obter mais informações, consulte [Como usar um notebook Studio com o Managed Service for Apache Flink para Apache Flink](#).

Para obter instruções sobre como usar a biblioteca, consulte o [Guia do usuário da biblioteca Flink AWS IoT TwinMaker](#).

Para obter instruções sobre a configuração AWS IoT TwinMaker com o início rápido em [exemplos AWS IoT TwinMaker](#), consulte o [arquivo README do aplicativo de insights de exemplos](#).

# Registrar em log e monitorar no AWS IoT TwinMaker

O monitoramento é uma parte importante para manter a confiabilidade, a disponibilidade e o desempenho do AWS IoT TwinMaker e das outras soluções da AWS. O AWS IoT TwinMaker é compatível com seguintes ferramentas de monitoramento para observar o serviço, informar quando algo está errado e executar ações automaticamente quando apropriado:

- O Amazon CloudWatch monitora em tempo real os seus recursos da AWS e as aplicações que você executa na AWS. É possível coletar e rastrear métricas, criar painéis personalizados e definir alarmes que o notificam ou que realizam ações quando uma métrica especificada atinge um limite definido. Por exemplo, você pode fazer o CloudWatch acompanhar o uso da CPU ou outras métricas das suas instâncias do Amazon EC2 e iniciar automaticamente novas instâncias quando necessário. Para obter mais informações, consulte o [Guia do usuário do Amazon CloudWatch](#).
- O Amazon CloudWatch Logs monitora, armazena e fornece acesso a seus arquivos de log dos gateways AWS IoT TwinMaker, CloudTrail e outras fontes. O CloudWatch Logs pode monitorar informações nos arquivos de log e notificar você quando determinados limites forem atingidos. Você também pode arquivar seus dados de log em armazenamento resiliente. Para obter mais informações, consulte o [Guia do usuário do Amazon CloudWatch Logs](#).
- O AWS CloudTrail captura chamadas de API e eventos relacionados feitos por, ou em nome de, sua conta da AWS e entrega os arquivos de log a um bucket do Amazon S3 que você especificar. Você pode identificar quais usuários e contas chamaram a AWS, o endereço IP de origem do qual as chamadas foram feitas e quando elas ocorreram. Para obter mais informações, consulte o [Guia do usuário do AWS CloudTrail](#).

## Tópicos

- [Monitorar o AWS IoT TwinMaker com métricas do Amazon CloudWatch](#)
- [Registrar em log chamadas de API do AWS IoT TwinMaker com o AWS CloudTrail](#)

## Monitorar o AWS IoT TwinMaker com métricas do Amazon CloudWatch

Você pode monitorar o AWS IoT TwinMaker usando o CloudWatch, que coleta dados brutos e os processa em métricas legíveis quase em tempo real. Essas estatísticas são mantidas por 15 meses, de maneira que você possa acessar informações históricas e ter uma perspectiva melhor de como

o aplicativo web ou o serviço está se saindo. Você também pode definir alarmes que observam determinados limites e enviam notificações ou realizam ações quando esses limites são atingidos. Para obter mais informações, consulte o [Guia do usuário do Amazon CloudWatch](#).

O AWS IoT TwinMaker publica as métricas e as dimensões listadas nas seções seguintes no namespace `AWS/IoTTwinMaker`.

#### Tip

O AWS IoT TwinMaker publica métricas em um intervalo de um minuto. Ao visualizar essas métricas em gráficos no console do CloudWatch, recomendamos que você escolha um período de 1 minuto para ver a maior resolução disponível de seus dados métricos.

## Sumário

- [Métricas](#)

## Métricas

O AWS IoT TwinMaker publica as seguintes métricas.

### Métricas

Métrica	Descrição
<code>ComponentTypeCreationFailure</code>	<p>Essa métrica informa se a criação do tipo de componente foi bem-sucedida.</p> <p>A métrica é publicada quando um tipo de componente está no estado <code>CREATING</code>. Isso acontece quando um tipo de componente é criado com as propriedades necessárias no inicializador do esquema e essas propriedades são instanciadas com valores padrão.</p> <p>O valor da métrica pode ser <code>0</code> para sucesso ou <code>1</code> para fracasso.</p> <p>Dimensões: <code>componentTypeID</code>, <code>workspaceID</code>.</p>

Métrica	Descrição
<code>ComponentTypeUpdateFailure</code>	<p>Unidades: contagem</p> <p>Essa métrica informa se a atualização do tipo de componente foi bem-sucedida.</p> <p>A métrica é publicada quando um tipo de componente está no estado UPDATING. Isso acontece quando um tipo de componente é atualizado com as propriedades necessárias no inicializador do esquema e essas propriedades são instanciadas com valores padrão.</p> <p>O valor da métrica pode ser 0 para sucesso ou 1 para fracasso.</p> <p>Dimensões: <code>componentTypeID</code>, <code>workspaceID</code>.</p> <p>Unidades: contagem</p>
<code>EntityCreationFailure</code>	<p>Essa métrica informa se a criação da entidade foi bem-sucedida. A métrica é publicada quando uma entidade está no estado CREATING. Isso acontece quando uma entidade é criada com um componente.</p> <p>O valor da métrica pode ser 0 para sucesso ou 1 para fracasso.</p> <p>Dimensões: <code>entityName</code>, <code>entityId</code>, <code>workspaceID</code>.</p> <p>Unidades: contagem</p>

Métrica	Descrição
EntityUpdateFailure	<p>Essa métrica informa se a atualização da entidade foi bem-sucedida. A métrica é publicada quando uma entidade está no estado UPDATING. Isso acontece quando uma entidade é atualizada.</p> <p>O valor da métrica pode ser 0 para sucesso ou 1 para fracasso.</p> <p>Dimensões: entityName, entityId, workspaceId.</p> <p>Unidades: contagem</p>
EntityDeletionFailure	<p>Essa métrica informa se a exclusão da entidade foi bem-sucedida. A métrica é publicada quando uma entidade está no estado DELETING. Isso acontece quando uma entidade é excluída.</p> <p>O valor da métrica pode ser 0 para sucesso ou 1 para fracasso.</p> <p>Dimensões: entityName, entityId, workspaceId.</p> <p>Unidades: contagem</p>

 Tip

Todas as métricas são publicadas no namespace `AWS/IoTTwinMaker`.

## Registrar em log chamadas de API do AWS IoT TwinMaker com o AWS CloudTrail

O AWS IoT TwinMaker é integrado ao AWS CloudTrail, um serviço que fornece um registro das ações realizadas por um usuário, uma função ou um serviço da AWS no AWS IoT TwinMaker.

O CloudTrail captura as chamadas de API do AWS IoT TwinMaker como eventos. As chamadas capturadas incluem as chamadas do console do AWS IoT TwinMaker e as chamadas de código para as operações da API do AWS IoT TwinMaker. Se você criar uma trilha, poderá habilitar a entrega contínua de eventos do CloudTrail para um bucket do Amazon S3, incluindo eventos para o AWS IoT TwinMaker. Se você não configurar uma trilha, ainda poderá visualizar os eventos mais recentes no console do CloudTrail em Event history (Histórico de eventos). Usando as informações coletadas pelo CloudTrail, é possível determinar a solicitação feita para o AWS IoT TwinMaker, o endereço IP no qual a solicitação foi feita, quem fez a solicitação, quando ela foi feita, além de detalhes adicionais.

Para obter mais informações sobre o CloudTrail, consulte o [Guia do usuário do AWS CloudTrail](#).

## Informações do AWS IoT TwinMaker no CloudTrail

Quando você cria sua conta AWS, o CloudTrail é habilitado automaticamente. O CloudTrail registra atividades de evento de suporte que ocorrem em AWS IoT TwinMaker, com outros eventos de serviço da AWS em Event history. Você pode visualizar, pesquisar e baixar eventos recentes em sua conta da AWS. Para obter mais informações, consulte [Visualizar eventos com o histórico de eventos do CloudTrail](#).

Para obter um registro contínuo de eventos na conta da AWS, incluindo eventos do AWS IoT TwinMaker, crie uma trilha. Uma trilha permite que o CloudTrail entregue arquivos de log a um bucket do Amazon S3. Por padrão, quando você cria uma trilha no console, ela é aplicada a todas as regiões da AWS. O CloudTrail registra eventos de todas as regiões na partição da AWS e entrega os arquivos de log ao bucket do Amazon S3 especificado. Além disso, é possível configurar outros produtos da AWS para analisar mais profundamente e agir sobre os dados de eventos coletados nos logs do CloudTrail. Para obter mais informações, consulte:

- [Visão geral da criação de uma trilha](#)
- [Serviços e integrações compatíveis com o CloudTrail](#)
- [Configurar notificações do Amazon SNS para o CloudTrail](#)
- [Receber arquivos de log do CloudTrail de várias regiões](#) e [Receber arquivos de log do CloudTrail de várias contas](#)

A maioria das operações do AWS IoT TwinMaker são registradas pelo CloudTrail e são documentadas na [Referência de API AWS IoT TwinMaker](#).

As seguintes operações do plano de dados não são registradas pelo CloudTrail:

- [GetPropertyValue](#)
- [GetPropertyValueHistory](#)
- [BatchPutPropertyValues](#)

Cada entrada de log ou evento contém informações sobre quem gerou a solicitação. As informações de identidade ajudam a determinar:

- Se a solicitação foi feita com credenciais de usuário da raiz ou do .
- Se a solicitação foi feita com credenciais de segurança temporárias de uma função ou de um usuário federado.
- Se a solicitação foi feita por outro serviço da AWS.

Para obter mais informações, consulte [Elemento userIdentity do CloudTrail](#).

# Segurança em AWS IoT TwinMaker

A segurança na nuvem AWS é a maior prioridade. Como AWS cliente, você se beneficia de data centers e arquiteturas de rede criados para atender aos requisitos das organizações mais sensíveis à segurança.

A segurança é uma responsabilidade compartilhada entre você AWS e você. O [modelo de responsabilidade compartilhada](#) descreve isso como segurança da nuvem e segurança na nuvem:

- **Segurança da nuvem** — AWS é responsável por proteger a infraestrutura que executa AWS os serviços no Nuvem AWS. AWS também fornece serviços que você pode usar com segurança. Auditores terceirizados testam e verificam regularmente a eficácia de nossa segurança como parte dos Programas de Conformidade Programas de [AWS](#) de . Para saber mais sobre os programas de conformidade que se aplicam AWS IoT TwinMaker, consulte [AWS Serviços no escopo do programa de conformidade AWS](#) .
- **Segurança na nuvem** — Sua responsabilidade é determinada pelo AWS serviço que você usa. Você também é responsável por outros fatores, incluindo a confidencialidade de seus dados, os requisitos da empresa e as leis e regulamentos aplicáveis.

Esta documentação ajuda você a entender como aplicar o modelo de responsabilidade compartilhada ao usar AWS IoT TwinMaker. Os tópicos a seguir mostram como configurar para atender AWS IoT TwinMaker aos seus objetivos de segurança e conformidade. Você também aprenderá a usar outros AWS serviços que ajudam a monitorar e proteger seus AWS IoT TwinMaker recursos.

## Tópicos

- [Proteção de dados em AWS IoT TwinMaker](#)
- [Identity and Access Management para AWS IoT TwinMaker](#)
- [AWS IoT TwinMaker e endpoints VPC de interface \(\)AWS PrivateLink](#)
- [Validação de conformidade para AWS IoT TwinMaker](#)
- [Resiliência em AWS IoT TwinMaker](#)
- [Segurança de infraestrutura em AWS IoT TwinMaker](#)

# Proteção de dados em AWS IoT TwinMaker

O modelo de [responsabilidade AWS compartilhada modelo](#) se aplica à proteção de dados em AWS IoT TwinMaker. Conforme descrito neste modelo, AWS é responsável por proteger a infraestrutura global que executa todos os Nuvem AWS. Você é responsável por manter o controle sobre o conteúdo hospedado nessa infraestrutura. Você também é responsável pelas tarefas de configuração e gerenciamento de segurança dos Serviços da AWS que usa. Para obter mais informações sobre a privacidade de dados, consulte as [Data Privacy FAQ](#). Para obter mais informações sobre a proteção de dados na Europa, consulte a postagem do blog [AWS Shared Responsibility Model and RGPD](#) no Blog de segurança da AWS .

Para fins de proteção de dados, recomendamos que você proteja Conta da AWS as credenciais e configure usuários individuais com AWS IAM Identity Center ou AWS Identity and Access Management (IAM). Dessa maneira, cada usuário receberá apenas as permissões necessárias para cumprir suas obrigações de trabalho. Recomendamos também que você proteja seus dados das seguintes formas:

- Use uma autenticação multifator (MFA) com cada conta.
- Use SSL/TLS para se comunicar com os recursos. AWS Exigimos TLS 1.2 e recomendamos TLS 1.3.
- Configure a API e o registro de atividades do usuário com AWS CloudTrail. Para obter informações sobre o uso de CloudTrail trilhas para capturar AWS atividades, consulte Como [trabalhar com CloudTrail trilhas](#) no Guia AWS CloudTrail do usuário.
- Use soluções de AWS criptografia, juntamente com todos os controles de segurança padrão Serviços da AWS.
- Use serviços gerenciados de segurança avançada, como o Amazon Macie, que ajuda a localizar e proteger dados sigilosos armazenados no Amazon S3.
- Se você precisar de módulos criptográficos validados pelo FIPS 140-3 ao acessar AWS por meio de uma interface de linha de comando ou de uma API, use um endpoint FIPS. Para obter mais informações sobre os endpoints FIPS disponíveis, consulte [Federal Information Processing Standard \(FIPS\) 140-3](#).

É altamente recomendável que nunca sejam colocadas informações confidenciais ou sigilosas, como endereços de e-mail de clientes, em tags ou campos de formato livre, como um campo Nome. Isso inclui quando você trabalha com AWS IoT TwinMaker ou Serviços da AWS usa o console, a API ou AWS SDKs. AWS CLI Quaisquer dados inseridos em tags ou em campos de texto de formato livre

usados para nomes podem ser usados para logs de faturamento ou de diagnóstico. Se você fornecer um URL para um servidor externo, recomendamos fortemente que não sejam incluídas informações de credenciais no URL para validar a solicitação a esse servidor.

## Criptografia em repouso

AWS IoT TwinMaker armazena as informações do seu espaço de trabalho em um bucket do Amazon S3 que o serviço cria para você, se você escolher. O bucket que o serviço cria para você tem a criptografia padrão do lado do servidor ativada. Se optar por usar seu próprio bucket do Amazon S3 ao criar um novo espaço de trabalho, recomendamos que você habilite a criptografia padrão do lado do servidor. Para obter mais informações sobre criptografia padrão no Amazon S3, consulte [Definir o comportamento da criptografia padrão do lado do servidor para os buckets do Amazon S3](#).

## Criptografia em trânsito

Todos os dados enviados AWS IoT TwinMaker são enviados por uma conexão TLS usando o protocolo HTTPS, portanto, eles são seguros por padrão enquanto estão em trânsito.

### Note

Recomendamos que você use HTTPS nos endereços de bucket do Amazon S3 como um controle para aplicar a criptografia em trânsito ao AWS IoT TwinMaker interagir com um bucket do Amazon S3. Para mais informações sobre buckets do Amazon S3, consulte [Criar, configurar e trabalhar com buckets do Amazon S3](#).

## Identity and Access Management para AWS IoT TwinMaker

AWS Identity and Access Management (IAM) é uma ferramenta AWS service (Serviço da AWS) que ajuda o administrador a controlar com segurança o acesso aos AWS recursos. Os administradores do IAM controlam quem pode ser autenticado (conectado) e autorizado (tem permissões) a usar AWS IoT TwinMaker os recursos. O IAM é um AWS service (Serviço da AWS) que você pode usar sem custo adicional.

### Tópicos

- [Público](#)
- [Autenticação com identidades](#)

- [Gerenciar o acesso usando políticas](#)
- [Como AWS IoT TwinMaker funciona com o IAM](#)
- [Exemplos de políticas baseadas em identidade para AWS IoT TwinMaker](#)
- [Solução de problemas AWS IoT TwinMaker de identidade e acesso](#)
- [Usar perfis vinculados a serviço do AWS IoT TwinMaker](#)
- [AWS políticas gerenciadas para AWS IoT TwinMaker](#)

## Público

A forma como você usa AWS Identity and Access Management (IAM) difere, dependendo do trabalho que você faz AWS IoT TwinMaker.

**Usuário do serviço** — Se você usar o AWS IoT TwinMaker serviço para realizar seu trabalho, seu administrador fornecerá as credenciais e as permissões de que você precisa. À medida que você usa mais AWS IoT TwinMaker recursos para fazer seu trabalho, talvez precise de permissões adicionais. Entender como o acesso é gerenciado pode ajudá-lo a solicitar as permissões corretas ao seu administrador. Se não for possível acessar um recurso no AWS IoT TwinMaker, consulte [Solução de problemas AWS IoT TwinMaker de identidade e acesso](#).

**Administrador de serviços** — Se você é responsável pelos AWS IoT TwinMaker recursos da sua empresa, provavelmente tem acesso total AWS IoT TwinMaker a. É seu trabalho determinar quais AWS IoT TwinMaker recursos e recursos seus usuários do serviço devem acessar. Envie as solicitações ao administrador do IAM para alterar as permissões dos usuários de serviço. Revise as informações nesta página para compreender os conceitos básicos do IAM. Para saber mais sobre como sua empresa pode usar o IAM com AWS IoT TwinMaker, consulte [Como AWS IoT TwinMaker funciona com o IAM](#).

**Administrador do IAM:** se você for um administrador do IAM, talvez queira saber detalhes sobre como pode gravar políticas para gerenciar o acesso ao AWS IoT TwinMaker. Para ver exemplos de políticas AWS IoT TwinMaker baseadas em identidade que você pode usar no IAM, consulte [Exemplos de políticas baseadas em identidade para AWS IoT TwinMaker](#)

## Autenticação com identidades

A autenticação é a forma como você faz login AWS usando suas credenciais de identidade. Você deve estar autenticado (conectado AWS) como o Usuário raiz da conta da AWS, como usuário do IAM ou assumindo uma função do IAM.

Você pode entrar AWS como uma identidade federada usando credenciais fornecidas por meio de uma fonte de identidade. AWS IAM Identity Center Usuários (IAM Identity Center), a autenticação de login único da sua empresa e suas credenciais do Google ou do Facebook são exemplos de identidades federadas. Quando você faz login como identidade federada, o administrador já configurou anteriormente a federação de identidades usando perfis do IAM. Ao acessar AWS usando a federação, você está assumindo indiretamente uma função.

Dependendo do tipo de usuário que você é, você pode entrar no AWS Management Console ou no portal de AWS acesso. Para obter mais informações sobre como fazer login AWS, consulte [Como fazer login Conta da AWS no Guia do Início de Sessão da AWS usuário](#).

Se você acessar AWS programaticamente, AWS fornece um kit de desenvolvimento de software (SDK) e uma interface de linha de comando (CLI) para assinar criptograficamente suas solicitações usando suas credenciais. Se você não usa AWS ferramentas, você mesmo deve assinar as solicitações. Para obter mais informações sobre como usar o método recomendado para designar solicitações por conta própria, consulte [Versão 4 do AWS Signature para solicitações de API](#) no Guia do usuário do IAM.

Independente do método de autenticação usado, também pode ser necessário fornecer informações adicionais de segurança. Por exemplo, AWS recomenda que você use a autenticação multifator (MFA) para aumentar a segurança da sua conta. Para saber mais, consulte [Autenticação multifator](#) no Guia do usuário do AWS IAM Identity Center e [Usar a autenticação multifator da AWS no IAM](#) no Guia do usuário do IAM.

## Conta da AWS usuário root

Ao criar uma Conta da AWS, você começa com uma identidade de login que tem acesso completo a todos Serviços da AWS os recursos da conta. Essa identidade é chamada de usuário Conta da AWS raiz e é acessada fazendo login com o endereço de e-mail e a senha que você usou para criar a conta. É altamente recomendável não usar o usuário-raiz para tarefas diárias. Proteja as credenciais do usuário-raiz e use-as para executar as tarefas que somente ele puder executar. Para obter a lista completa das tarefas que exigem login como usuário-raiz, consulte [Tarefas que exigem credenciais de usuário-raiz](#) no Guia do Usuário do IAM.

## Identidade federada

Como prática recomendada, exija que usuários humanos, incluindo usuários que precisam de acesso de administrador, usem a federação com um provedor de identidade para acessar Serviços da AWS usando credenciais temporárias.

Uma identidade federada é um usuário do seu diretório de usuários corporativo, de um provedor de identidade da web AWS Directory Service, do diretório do Identity Center ou de qualquer usuário que acesse usando credenciais fornecidas Serviços da AWS por meio de uma fonte de identidade. Quando as identidades federadas são acessadas Contas da AWS, elas assumem funções, e as funções fornecem credenciais temporárias.

Para o gerenciamento de acesso centralizado, é recomendável usar o AWS IAM Identity Center. Você pode criar usuários e grupos no IAM Identity Center ou pode se conectar e sincronizar com um conjunto de usuários e grupos em sua própria fonte de identidade para uso em todos os seus Contas da AWS aplicativos. Para obter mais informações sobre o Centro de Identidade do IAM, consulte [O que é o Centro de Identidade do IAM?](#) no Guia do Usuário do AWS IAM Identity Center .

## Usuários e grupos do IAM

Um [usuário do IAM](#) é uma identidade dentro da sua Conta da AWS que tem permissões específicas para uma única pessoa ou aplicativo. Sempre que possível, é recomendável contar com credenciais temporárias em vez de criar usuários do IAM com credenciais de longo prazo, como senhas e chaves de acesso. No entanto, se você tiver casos de uso específicos que exijam credenciais de longo prazo com usuários do IAM, é recomendável alternar as chaves de acesso. Para obter mais informações, consulte [Alternar as chaves de acesso regularmente para casos de uso que exijam credenciais de longo prazo](#) no Guia do Usuário do IAM.

Um [grupo do IAM](#) é uma identidade que especifica uma coleção de usuários do IAM. Não é possível fazer login como um grupo. É possível usar grupos para especificar permissões para vários usuários de uma vez. Os grupos facilitam o gerenciamento de permissões para grandes conjuntos de usuários. Por exemplo, você pode ter um grupo chamado IAMAdminse conceder a esse grupo permissões para administrar recursos do IAM.

Usuários são diferentes de perfis. Um usuário é exclusivamente associado a uma pessoa ou a uma aplicação, mas um perfil pode ser assumido por qualquer pessoa que precisar dele. Os usuários têm credenciais permanentes de longo prazo, mas os perfis fornecem credenciais temporárias. Para saber mais, consulte [Casos de uso para usuários do IAM](#) no Guia do usuário do IAM.

## Perfis do IAM

Uma [função do IAM](#) é uma identidade dentro da sua Conta da AWS que tem permissões específicas. Ele é semelhante a um usuário do IAM, mas não está associado a uma pessoa específica. Para assumir temporariamente uma função do IAM no AWS Management Console, você pode [alternar de um usuário para uma função do IAM \(console\)](#). Você pode assumir uma função chamando uma

operação de AWS API AWS CLI ou usando uma URL personalizada. Para obter mais informações sobre métodos para usar perfis, consulte [Métodos para assumir um perfil](#) no Guia do usuário do IAM.

Perfis do IAM com credenciais temporárias são úteis nas seguintes situações:

- **Acesso de usuário federado:** para atribuir permissões a identidades federadas, é possível criar um perfil e definir permissões para ele. Quando uma identidade federada é autenticada, essa identidade é associada ao perfil e recebe as permissões definidas por ele. Para ter mais informações sobre perfis para federação, consulte [Criar um perfil para um provedor de identidade de terceiros \(federação\)](#) no Guia do usuário do IAM. Se usar o Centro de Identidade do IAM, configure um conjunto de permissões. Para controlar o que suas identidades podem acessar após a autenticação, o Centro de Identidade do IAM correlaciona o conjunto de permissões a um perfil no IAM. Para obter informações sobre conjuntos de permissões, consulte [Conjuntos de Permissões](#) no Guia do Usuário do AWS IAM Identity Center .
- **Permissões temporárias para usuários do IAM:** um usuário ou um perfil do IAM pode presumir um perfil do IAM para obter temporariamente permissões diferentes para uma tarefa específica.
- **Acesso entre contas:** é possível usar um perfil do IAM para permitir que alguém (uma entidade principal confiável) em outra conta acesse recursos em sua conta. Os perfis são a principal forma de conceder acesso entre contas. No entanto, com alguns Serviços da AWS, você pode anexar uma política diretamente a um recurso (em vez de usar uma função como proxy). Para conhecer a diferença entre perfis e políticas baseadas em recurso para acesso entre contas, consulte [Acesso a recursos entre contas no IAM](#) no Guia do usuário do IAM.
- **Acesso entre serviços** — Alguns Serviços da AWS usam recursos em outros Serviços da AWS. Por exemplo, quando você faz uma chamada em um serviço, é comum que esse serviço execute aplicativos na Amazon EC2 ou armazene objetos no Amazon S3. Um serviço pode fazer isso usando as permissões da entidade principal da chamada, usando um perfil de serviço ou um perfil vinculado ao serviço.
- **Sessões de acesso direto (FAS)** — Quando você usa um usuário ou uma função do IAM para realizar ações AWS, você é considerado principal. Ao usar alguns serviços, você pode executar uma ação que inicia outra ação em um serviço diferente. O FAS usa as permissões do diretor chamando um AWS service (Serviço da AWS), combinadas com a solicitação AWS service (Serviço da AWS) para fazer solicitações aos serviços posteriores. As solicitações do FAS são feitas somente quando um serviço recebe uma solicitação que requer interações com outros Serviços da AWS ou com recursos para ser concluída. Nesse caso, você precisa ter permissões para executar ambas as ações. Para obter detalhes da política ao fazer solicitações de FAS, consulte [Sessões de acesso direto](#).

- **Perfil de serviço:** um perfil de serviço é um [perfil do IAM](#) que um serviço assume para executar ações em seu nome. Um administrador do IAM pode criar, modificar e excluir um perfil de serviço do IAM. Para obter mais informações, consulte [Criar um perfil para delegar permissões a um AWS service \(Serviço da AWS\)](#) no Guia do Usuário do IAM.
- **Função vinculada ao serviço** — Uma função vinculada ao serviço é um tipo de função de serviço vinculada a um AWS service (Serviço da AWS). O serviço pode presumir o perfil para executar uma ação em seu nome. As funções vinculadas ao serviço aparecem em você Conta da AWS e são de propriedade do serviço. Um administrador do IAM pode visualizar, mas não editar as permissões para perfis vinculados a serviço.
- **Aplicativos em execução na Amazon EC2** — Você pode usar uma função do IAM para gerenciar credenciais temporárias para aplicativos que estão sendo executados em uma EC2 instância e fazendo solicitações AWS CLI de AWS API. Isso é preferível ao armazenamento de chaves de acesso na EC2 instância. Para atribuir uma AWS função a uma EC2 instância e disponibilizá-la para todos os aplicativos, você cria um perfil de instância anexado à instância. Um perfil de instância contém a função e permite que programas em execução na EC2 instância recebam credenciais temporárias. Para obter mais informações, consulte [Usar uma função do IAM para conceder permissões a aplicativos executados em EC2 instâncias da Amazon](#) no Guia do usuário do IAM.

## Gerenciar o acesso usando políticas

Você controla o acesso AWS criando políticas e anexando-as a AWS identidades ou recursos. Uma política é um objeto AWS que, quando associada a uma identidade ou recurso, define suas permissões. AWS avalia essas políticas quando um principal (usuário, usuário raiz ou sessão de função) faz uma solicitação. As permissões nas políticas determinam se a solicitação será permitida ou negada. A maioria das políticas é armazenada AWS como documentos JSON. Para obter mais informações sobre a estrutura e o conteúdo de documentos de políticas JSON, consulte [Visão geral das políticas JSON](#) no Guia do usuário do IAM.

Os administradores podem usar políticas AWS JSON para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos e em que condições.

Por padrão, usuários e perfis não têm permissões. Para conceder permissão aos usuários para executar ações nos recursos que eles precisam, um administrador do IAM pode criar políticas do IAM. O administrador pode então adicionar as políticas do IAM aos perfis e os usuários podem assumir os perfis.

As políticas do IAM definem permissões para uma ação independentemente do método usado para executar a operação. Por exemplo, suponha que você tenha uma política que permite a ação `iam:GetRole`. Um usuário com essa política pode obter informações de função da AWS Management Console AWS CLI, da ou da AWS API.

## Políticas baseadas em identidade

As políticas baseadas em identidade são documentos de políticas de permissões JSON que você pode anexar a uma identidade, como usuário, grupo de usuários ou perfil do IAM. Essas políticas controlam quais ações os usuários e perfis podem realizar, em quais recursos e em que condições. Para saber como criar uma política baseada em identidade, consulte [Definir permissões personalizadas do IAM com as políticas gerenciadas pelo cliente](#) no Guia do Usuário do IAM.

As políticas baseadas em identidade podem ser categorizadas como políticas em linha ou políticas gerenciadas. As políticas em linha são anexadas diretamente a um único usuário, grupo ou perfil. As políticas gerenciadas são políticas autônomas que você pode associar a vários usuários, grupos e funções em seu Conta da AWS. As políticas AWS gerenciadas incluem políticas gerenciadas e políticas gerenciadas pelo cliente. Para saber como escolher entre uma política gerenciada ou uma política em linha, consulte [Escolher entre políticas gerenciadas e políticas em linha](#) no Guia do usuário do IAM.

## Políticas baseadas em recursos

Políticas baseadas em recursos são documentos de políticas JSON que você anexa a um recurso. São exemplos de políticas baseadas em recursos as políticas de confiança de perfil do IAM e as políticas de bucket do Amazon S3. Em serviços compatíveis com políticas baseadas em recursos, os administradores de serviço podem usá-las para controlar o acesso a um recurso específico. Para o atributo ao qual a política está anexada, a política define quais ações uma entidade principal especificado pode executar nesse atributo e em que condições. Você deve [especificar uma entidade principal](#) em uma política baseada em recursos. Os diretores podem incluir contas, usuários, funções, usuários federados ou. Serviços da AWS

Políticas baseadas em recursos são políticas em linha localizadas nesse serviço. Você não pode usar políticas AWS gerenciadas do IAM em uma política baseada em recursos.

## Listas de controle de acesso (ACLs)

As listas de controle de acesso (ACLs) controlam quais diretores (membros da conta, usuários ou funções) têm permissões para acessar um recurso. ACLs são semelhantes às políticas baseadas em recursos, embora não usem o formato de documento de política JSON.

O Amazon S3 e o AWS WAF Amazon VPC são exemplos de serviços que oferecem suporte. ACLs Para saber mais ACLs, consulte a [visão geral da lista de controle de acesso \(ACL\)](#) no Guia do desenvolvedor do Amazon Simple Storage Service.

## Outros tipos de política

AWS oferece suporte a tipos de políticas adicionais menos comuns. Esses tipos de política podem definir o máximo de permissões concedidas a você pelos tipos de política mais comuns.

- **Limites de permissões:** um limite de permissões é um recurso avançado no qual você define o máximo de permissões que uma política baseada em identidade pode conceder a uma entidade do IAM (usuário ou perfil do IAM). É possível definir um limite de permissões para uma entidade. As permissões resultantes são a interseção das políticas baseadas em identidade de uma entidade com seus limites de permissões. As políticas baseadas em recurso que especificam o usuário ou o perfil no campo `Principal` não são limitadas pelo limite de permissões. Uma negação explícita em qualquer uma dessas políticas substitui a permissão. Para obter mais informações sobre limites de permissões, consulte [Limites de permissões para identidades do IAM](#) no Guia do usuário do IAM.
- **Políticas de controle de serviço (SCPs)** — SCPs são políticas JSON que especificam as permissões máximas para uma organização ou unidade organizacional (OU) em AWS Organizations. AWS Organizations é um serviço para agrupar e gerenciar centralmente várias Contas da AWS que sua empresa possui. Se você habilitar todos os recursos em uma organização, poderá aplicar políticas de controle de serviço (SCPs) a qualquer uma ou a todas as suas contas. O SCP limita as permissões para entidades nas contas dos membros, incluindo cada uma Usuário raiz da conta da AWS. Para obter mais informações sobre Organizations e SCPs, consulte [Políticas de controle de serviços](#) no Guia AWS Organizations do Usuário.
- **Políticas de controle de recursos (RCPs)** — RCPs são políticas JSON que você pode usar para definir o máximo de permissões disponíveis para recursos em suas contas sem atualizar as políticas do IAM anexadas a cada recurso que você possui. O RCP limita as permissões para recursos nas contas dos membros e pode afetar as permissões efetivas para identidades, incluindo a Usuário raiz da conta da AWS, independentemente de pertencerem à sua organização. Para obter mais informações sobre Organizations e RCPs, incluindo uma lista Serviços da AWS desse suporte RCPs, consulte [Políticas de controle de recursos \(RCPs\)](#) no Guia AWS Organizations do usuário.
- **Políticas de sessão:** são políticas avançadas que você transmite como um parâmetro quando cria de forma programática uma sessão temporária para um perfil ou um usuário federado. As permissões da sessão resultante são a interseção das políticas baseadas em identidade do

usuário ou do perfil e das políticas de sessão. As permissões também podem ser provenientes de uma política baseada em recursos. Uma negação explícita em qualquer uma dessas políticas substitui a permissão. Para obter mais informações, consulte [Políticas de sessão](#) no Guia do usuário do IAM.

## Vários tipos de política

Quando vários tipos de política são aplicáveis a uma solicitação, é mais complicado compreender as permissões resultantes. Para saber como AWS determinar se uma solicitação deve ser permitida quando vários tipos de políticas estão envolvidos, consulte [Lógica de avaliação de políticas](#) no Guia do usuário do IAM.

## Como AWS IoT TwinMaker funciona com o IAM

Antes de usar o IAM para gerenciar o acesso AWS IoT TwinMaker, saiba com quais recursos do IAM estão disponíveis para uso AWS IoT TwinMaker.

### Recursos do IAM que você pode usar com AWS IoT TwinMaker

Atributo do IAM	AWS IoT TwinMaker apoio
<a href="#">Políticas baseadas em identidade</a>	Sim
<a href="#">Políticas baseadas em recurso</a>	Não
<a href="#">Ações de políticas</a>	Sim
<a href="#">Recursos de políticas</a>	Sim
<a href="#">Chaves de condição de políticas</a>	Sim
<a href="#">ACLs</a>	Não
<a href="#">ABAC (tags em políticas)</a>	Parcial
<a href="#">Credenciais temporárias</a>	Sim
<a href="#">Permissões de entidade principal</a>	Sim

Atributo do IAM	AWS IoT TwinMaker apoio
<a href="#">Perfis de serviço</a>	Sim
<a href="#">Perfis vinculados a serviço</a>	Não

Para ter uma visão de alto nível de como AWS IoT TwinMaker e outros AWS serviços funcionam com a maioria dos recursos do IAM, consulte [AWS os serviços que funcionam com o IAM](#) no Guia do AWS IAM Identity Center usuário.

## Políticas baseadas em identidade para AWS IoT TwinMaker

Compatível com políticas baseadas em identidade: sim

As políticas baseadas em identidade são documentos de políticas de permissões JSON que você pode anexar a uma identidade, como usuário do IAM, grupo de usuários ou perfil. Essas políticas controlam quais ações os usuários e perfis podem realizar, em quais recursos e em que condições. Para saber como criar uma política baseada em identidade, consulte [Definir permissões personalizadas do IAM com as políticas gerenciadas pelo cliente](#) no Guia do Usuário do IAM.

Com as políticas baseadas em identidade do IAM, é possível especificar ações e recursos permitidos ou negados, assim como as condições sob as quais as ações são permitidas ou negadas. Você não pode especificar a entidade principal em uma política baseada em identidade porque ela se aplica ao usuário ou perfil ao qual ela está anexada. Para saber mais sobre todos os elementos que podem ser usados em uma política JSON, consulte [Referência de elemento de política JSON do IAM](#) no Guia do usuário do IAM.

### Exemplos de políticas baseadas em identidade para AWS IoT TwinMaker

Para ver exemplos de políticas AWS IoT TwinMaker baseadas em identidade, consulte [Exemplos de políticas baseadas em identidade para AWS IoT TwinMaker](#)

## Políticas baseadas em recursos dentro AWS IoT TwinMaker

Compatibilidade com políticas baseadas em recursos: não

Políticas baseadas em recursos são documentos de políticas JSON que você anexa a um recurso. São exemplos de políticas baseadas em recursos as políticas de confiança de perfil do IAM e as políticas de bucket do Amazon S3. Em serviços compatíveis com políticas baseadas em recursos,

os administradores de serviço podem usá-las para controlar o acesso a um recurso específico. Para o atributo ao qual a política está anexada, a política define quais ações uma entidade principal especificado pode executar nesse atributo e em que condições. Você deve [especificar uma entidade principal](#) em uma política baseada em recursos. Os diretores podem incluir contas, usuários, funções, usuários federados ou. Serviços da AWS

Para permitir o acesso entre contas, você pode especificar uma conta inteira ou as entidades do IAM em outra conta como a entidade principal em uma política baseada em recursos. Adicionar uma entidade principal entre contas à política baseada em recurso é apenas metade da tarefa de estabelecimento da relação de confiança. Quando o principal e o recurso são diferentes Contas da AWS, um administrador do IAM na conta confiável também deve conceder permissão à entidade principal (usuário ou função) para acessar o recurso. Eles concedem permissão ao anexar uma política baseada em identidade para a entidade. No entanto, se uma política baseada em recurso conceder acesso a uma entidade principal na mesma conta, nenhuma política baseada em identidade adicional será necessária. Consulte mais informações em [Acesso a recursos entre contas no IAM](#) no Guia do usuário do IAM.

## Ações políticas para AWS IoT TwinMaker

Compatível com ações de políticas: sim

Os administradores podem usar políticas AWS JSON para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos e em que condições.

O elemento `Action` de uma política JSON descreve as ações que podem ser usadas para permitir ou negar acesso em uma política. As ações de política geralmente têm o mesmo nome da operação de AWS API associada. Existem algumas exceções, como ações somente de permissão, que não têm uma operação de API correspondente. Algumas operações também exigem várias ações em uma política. Essas ações adicionais são chamadas de ações dependentes.

Incluem ações em uma política para conceder permissões para executar a operação associada.

Para ver uma lista de AWS IoT TwinMaker ações, consulte [Ações definidas por AWS IoT TwinMaker](#) na Referência de Autorização de Serviço.

As ações de política AWS IoT TwinMaker usam o seguinte prefixo antes da ação:

```
iottwinmaker
```

Para especificar várias ações em uma única declaração, separe-as com vírgulas.

```
"Action": [  
  "iottwinmaker:action1",  
  "iottwinmaker:action2"  
]
```

Para ver exemplos de políticas AWS IoT TwinMaker baseadas em identidade, consulte [Exemplos de políticas baseadas em identidade para AWS IoT TwinMaker](#)

## Recursos políticos para AWS IoT TwinMaker

Compatível com recursos de políticas: sim

Os administradores podem usar políticas AWS JSON para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos e em que condições.

O elemento de política JSON `Resource` especifica o objeto ou os objetos aos quais a ação se aplica. As instruções devem incluir um elemento `Resource` ou `NotResource`. Como prática recomendada, especifique um recurso usando seu [nome do recurso da Amazon \(ARN\)](#). Isso pode ser feito para ações que oferecem compatibilidade com um tipo de recurso específico, conhecido como permissões em nível de recurso.

Para ações que não oferecem compatibilidade com permissões em nível de recurso, como operações de listagem, use um curinga (\*) para indicar que a instrução se aplica a todos os recursos.

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

Para ver uma lista dos tipos de AWS IoT TwinMaker recursos e seus ARNs, consulte [Recursos definidos por AWS IoT TwinMaker](#) na Referência de Autorização de Serviço. Para saber com quais ações é possível especificar o ARN de cada atributo, consulte [Ações definidas pelo AWS IoT TwinMaker](#).

Para ver exemplos de políticas AWS IoT TwinMaker baseadas em identidade, consulte [Exemplos de políticas baseadas em identidade para AWS IoT TwinMaker](#)

## Chaves de condição de política para AWS IoT TwinMaker

Compatível com chaves de condição de política específicas de serviço: sim

Os administradores podem usar políticas AWS JSON para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos e em que condições.

O elemento `Condition` (ou bloco `Condition`) permite que você especifique condições nas quais uma instrução estiver em vigor. O elemento `Condition` é opcional. É possível criar expressões condicionais que usem [agentes de condição](#), como “igual a” ou “menor que”, para fazer a condição da política corresponder aos valores na solicitação.

Se você especificar vários elementos de `Condition` em uma declaração ou várias chaves em um único elemento de `Condition`, a AWS os avaliará usando uma operação lógica AND. Se você especificar vários valores para uma única chave de condição, AWS avalia a condição usando uma OR operação lógica. Todas as condições devem ser atendidas antes que as permissões da instrução sejam concedidas.

Você também pode usar variáveis de espaço reservado ao especificar condições. Por exemplo, é possível conceder a um usuário do IAM permissão para acessar um recurso somente se ele estiver marcado com seu nome de usuário do IAM. Para obter mais informações, consulte [Elementos da política do IAM: variáveis e tags](#) no Guia do usuário do IAM.

AWS suporta chaves de condição globais e chaves de condição específicas do serviço. Para ver todas as chaves de condição AWS globais, consulte as [chaves de contexto de condição AWS global](#) no Guia do usuário do IAM.

Para ver uma lista de chaves de AWS IoT TwinMaker condição, consulte [Chaves de condição AWS IoT TwinMaker](#) na Referência de autorização de serviço. Para saber com quais ações e recursos você pode usar uma chave de condição, consulte [Ações definidas por AWS IoT TwinMaker](#).

Para ver exemplos de políticas AWS IoT TwinMaker baseadas em identidade, consulte [Exemplos de políticas baseadas em identidade para AWS IoT TwinMaker](#)

## Listas de controle de acesso (ACLs) em AWS IoT TwinMaker

Suportes ACLs: Não

As listas de controle de acesso (ACLs) controlam quais diretores (membros da conta, usuários ou funções) têm permissões para acessar um recurso. ACLs são semelhantes às políticas baseadas em recursos, embora não usem o formato de documento de política JSON.

## Controle de acesso baseado em atributos (ABAC) com AWS IoT TwinMaker

Compatível com ABAC (tags em políticas): parcial

O controle de acesso por atributo (ABAC) é uma estratégia de autorização que define as permissões com base em atributos. Em AWS, esses atributos são chamados de tags. Você pode anexar tags a entidades do IAM (usuários ou funções) e a vários AWS recursos. Marcar de entidades e atributos é a primeira etapa do ABAC. Em seguida, você cria políticas de ABAC para permitir operações quando a tag da entidade principal corresponder à tag do recurso que ela estiver tentando acessar.

O ABAC é útil em ambientes que estão crescendo rapidamente e ajuda em situações em que o gerenciamento de políticas se torna um problema.

Para controlar o acesso baseado em tags, forneça informações sobre as tags no [elemento de condição](#) de uma política usando as `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` ou chaves de condição `aws:TagKeys`.

Se um serviço for compatível com as três chaves de condição para cada tipo de recurso, o valor será Sim para o serviço. Se um serviço for compatível com as três chaves de condição somente para alguns tipos de recursos, o valor será Parcial

Para obter mais informações sobre o ABAC, consulte [Definir permissões com autorização do ABAC](#) no Guia do usuário do IAM. Para visualizar um tutorial com etapas para configurar o ABAC, consulte [Usar controle de acesso baseado em atributos \(ABAC\)](#) no Guia do usuário do IAM.

## Usando credenciais temporárias com AWS IoT TwinMaker

Compatível com credenciais temporárias: sim

Alguns Serviços da AWS não funcionam quando você faz login usando credenciais temporárias. Para obter informações adicionais, incluindo quais Serviços da AWS funcionam com credenciais temporárias, consulte Serviços da AWS "[Trabalhe com o IAM](#)" no Guia do usuário do IAM.

Você está usando credenciais temporárias se fizer login AWS Management Console usando qualquer método, exceto um nome de usuário e senha. Por exemplo, quando você acessa AWS usando o link de login único (SSO) da sua empresa, esse processo cria automaticamente credenciais temporárias. Você também cria automaticamente credenciais temporárias quando faz login no console como usuário e, em seguida, alterna perfis. Para obter mais informações sobre como alternar funções, consulte [Alternar para um perfil do IAM \(console\)](#) no Guia do usuário do IAM.

Você pode criar manualmente credenciais temporárias usando a AWS API AWS CLI ou. Em seguida, você pode usar essas credenciais temporárias para acessar AWS. AWS recomenda que você gere credenciais temporárias dinamicamente em vez de usar chaves de acesso de longo prazo. Para obter mais informações, consulte [Credenciais de segurança temporárias no IAM](#).

## Permissões principais entre serviços para AWS IoT TwinMaker

Compatibilidade com o recurso de encaminhamento de sessões de acesso (FAS): sim

Quando você usa um usuário ou uma função do IAM para realizar ações em AWS, você é considerado um principal. Ao usar alguns serviços, você pode executar uma ação que inicia outra ação em um serviço diferente. O FAS usa as permissões do diretor chamando um AWS service (Serviço da AWS), combinadas com a solicitação AWS service (Serviço da AWS) para fazer solicitações aos serviços posteriores. As solicitações do FAS são feitas somente quando um serviço recebe uma solicitação que requer interações com outros Serviços da AWS ou com recursos para ser concluída. Nesse caso, você precisa ter permissões para executar ambas as ações. Para obter detalhes da política ao fazer solicitações de FAS, consulte [Sessões de acesso direto](#).

## Funções de serviço para AWS IoT TwinMaker

Compatível com perfis de serviço: sim

O perfil de serviço é um [perfil do IAM](#) que um serviço assume para executar ações em seu nome. Um administrador do IAM pode criar, modificar e excluir um perfil de serviço do IAM. Para obter mais informações, consulte [Criar um perfil para delegar permissões a um AWS service \(Serviço da AWS\)](#) no Guia do Usuário do IAM.

### Warning

Alterar as permissões de uma função de serviço pode interromper AWS IoT TwinMaker a funcionalidade. Edite as funções de serviço somente quando AWS IoT TwinMaker fornecer orientação para fazer isso.

## Funções vinculadas a serviços para AWS IoT TwinMaker

Compatível com perfis vinculados ao serviço: Não

Uma função vinculada ao serviço é um tipo de função de serviço vinculada a um. AWS service (Serviço da AWS) O serviço pode presumir o perfil para executar uma ação em seu nome. As

funções vinculadas ao serviço aparecem em você Conta da AWS e são de propriedade do serviço. Um administrador do IAM pode visualizar, mas não editar as permissões para funções vinculadas ao serviço.

Para obter detalhes sobre como criar ou gerenciar perfis vinculados a serviços, consulte [Serviços da AWS que funcionam com o IAM](#). Encontre um serviço na tabela que inclua um Yes na coluna Perfil vinculado ao serviço. Escolha o link Sim para visualizar a documentação do perfil vinculado a serviço desse serviço.

## Exemplos de políticas baseadas em identidade para AWS IoT TwinMaker

Por padrão, usuários e perfis não têm permissão para criar ou modificar recursos do AWS IoT TwinMaker. Eles também não podem realizar tarefas usando a AWS API AWS Management Console, AWS Command Line Interface (AWS CLI) ou. Para conceder permissão aos usuários para executar ações nos recursos que eles precisam, um administrador do IAM pode criar políticas do IAM. O administrador pode então adicionar as políticas do IAM aos perfis e os usuários podem assumir os perfis.

Para aprender a criar uma política baseada em identidade do IAM ao usar esses documentos de política em JSON de exemplo, consulte [Criar políticas do IAM \(console\)](#) no Guia do usuário do IAM.

Para obter detalhes sobre ações e tipos de recursos definidos por AWS IoT TwinMaker, incluindo o formato do ARNs para cada um dos tipos de recursos, consulte [Ações, recursos e chaves de condição AWS IoT TwinMaker na Referência de Autorização de Serviço](#).

### Tópicos

- [Práticas recomendadas de política](#)
- [Usar o console do AWS IoT TwinMaker](#)
- [Permitir que os usuários visualizem suas próprias permissões](#)

### Práticas recomendadas de política

As políticas baseadas em identidade determinam se alguém pode criar, acessar ou excluir AWS IoT TwinMaker recursos em sua conta. Essas ações podem incorrer em custos para sua Conta da AWS. Ao criar ou editar políticas baseadas em identidade, siga estas diretrizes e recomendações:

- Comece com as políticas AWS gerenciadas e avance para as permissões de privilégios mínimos — Para começar a conceder permissões aos seus usuários e cargas de trabalho, use as políticas

AWS gerenciadas que concedem permissões para muitos casos de uso comuns. Eles estão disponíveis no seu Conta da AWS. Recomendamos que você reduza ainda mais as permissões definindo políticas gerenciadas pelo AWS cliente que sejam específicas para seus casos de uso. Para obter mais informações, consulte [Políticas gerenciadas pela AWS](#) ou [Políticas gerenciadas pela AWS para funções de trabalho](#) no Guia do usuário do IAM.

- Aplique permissões de privilégio mínimo: ao definir permissões com as políticas do IAM, conceda apenas as permissões necessárias para executar uma tarefa. Você faz isso definindo as ações que podem ser executadas em recursos específicos sob condições específicas, também conhecidas como permissões de privilégio mínimo. Para obter mais informações sobre como usar o IAM para aplicar permissões, consulte [Políticas e permissões no IAM](#) no Guia do usuário do IAM.
- Use condições nas políticas do IAM para restringir ainda mais o acesso: você pode adicionar uma condição às políticas para limitar o acesso a ações e recursos. Por exemplo, você pode escrever uma condição de política para especificar que todas as solicitações devem ser enviadas usando SSL. Você também pode usar condições para conceder acesso às ações de serviço se elas forem usadas por meio de uma ação específica AWS service (Serviço da AWS), como AWS CloudFormation. Para obter mais informações, consulte [Elementos da política JSON do IAM: condição](#) no Guia do usuário do IAM.
- Use o IAM Access Analyzer para validar suas políticas do IAM a fim de garantir permissões seguras e funcionais: o IAM Access Analyzer valida as políticas novas e existentes para que elas sigam a linguagem de política do IAM (JSON) e as práticas recomendadas do IAM. O IAM Access Analyzer oferece mais de cem verificações de política e recomendações práticas para ajudar a criar políticas seguras e funcionais. Para obter mais informações, consulte [Validação de políticas do IAM Access Analyzer](#) no Guia do Usuário do IAM.
- Exigir autenticação multifator (MFA) — Se você tiver um cenário que exija usuários do IAM ou um usuário root, ative Conta da AWS a MFA para obter segurança adicional. Para exigir MFA quando as operações de API forem chamadas, adicione condições de MFA às suas políticas. Para obter mais informações, consulte [Configuração de acesso à API protegido por MFA](#) no Guia do Usuário do IAM.

Para obter mais informações sobre as práticas recomendadas do IAM, consulte [Práticas recomendadas de segurança no IAM](#) no Guia do usuário do IAM.

## Usar o console do AWS IoT TwinMaker

Para acessar o AWS IoT TwinMaker console, você deve ter um conjunto mínimo de permissões. Essas permissões devem permitir que você liste e visualize detalhes sobre os AWS IoT TwinMaker

recursos em seu Conta da AWS. Caso crie uma política baseada em identidade mais restritiva que as permissões mínimas necessárias, o console não funcionará como pretendido para entidades (usuários ou perfis) com essa política.

Você não precisa permitir permissões mínimas do console para usuários que estão fazendo chamadas somente para a API AWS CLI ou para a AWS API. Em vez disso, permita o acesso somente a ações que correspondam à operação de API que estiverem tentando executar.

Para garantir que usuários e funções ainda possam usar o AWS IoT TwinMaker console, anexe também a política AWS IoT TwinMaker ConsoleAccess ou a política ReadOnly AWS gerenciada às entidades. Para obter mais informações, consulte [Como adicionar permissões a um usuário](#) no Guia do usuário do AWS IAM Identity Center .

## Permitir que os usuários visualizem suas próprias permissões

Este exemplo mostra como criar uma política que permita que os usuários do IAM visualizem as políticas gerenciadas e em linha anexadas a sua identidade de usuário. Essa política inclui permissões para concluir essa ação no console ou programaticamente usando a API AWS CLI ou AWS .

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ViewOwnUserInfo",
      "Effect": "Allow",
      "Action": [
        "iam:GetUserPolicy",
        "iam:ListGroupsWithUser",
        "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": "*"
}
]
```

## Solução de problemas AWS IoT TwinMaker de identidade e acesso

Use as informações a seguir para ajudá-lo a diagnosticar e corrigir problemas comuns que você pode encontrar ao trabalhar com AWS IoT TwinMaker um IAM.

### Tópicos

- [Não estou autorizado a realizar uma ação em AWS IoT TwinMaker](#)
- [Não estou autorizado a realizar iam: PassRole](#)
- [Quero permitir que pessoas fora da minha Conta da AWS acessem meus AWS IoT TwinMaker recursos](#)

### Não estou autorizado a realizar uma ação em AWS IoT TwinMaker

Se você receber uma mensagem de erro informando que não tem autorização para executar uma ação, suas políticas deverão ser atualizadas para permitir que você realize a ação.

O erro do exemplo a seguir ocorre quando o usuário do IAM mateojackson tenta usar o console para visualizar detalhes sobre um atributo *my-example-widget* fictício, mas não tem as permissões `iottwinmaker:GetWidget` fictícias.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
iottwinmaker:GetWidget on resource: my-example-widget
```

Nesse caso, a política do usuário mateojackson deve ser atualizada para permitir o acesso ao recurso *my-example-widget* usando a ação `iottwinmaker:GetWidget`.

Se precisar de ajuda, entre em contato com seu AWS administrador. Seu administrador é a pessoa que forneceu suas credenciais de login.

## Não estou autorizado a realizar iam: PassRole

Se você receber uma mensagem de erro informando que não está autorizado a executar a ação `iam:PassRole`, as suas políticas devem ser atualizadas para permitir que você passe uma função para o AWS IoT TwinMaker.

Alguns Serviços da AWS permitem que você passe uma função existente para esse serviço em vez de criar uma nova função de serviço ou uma função vinculada ao serviço. Para fazê-lo, você deve ter permissões para passar o perfil para o serviço.

O exemplo de erro a seguir ocorre quando uma usuária do IAM chamada `marymajor` tenta utilizar o console para executar uma ação no AWS IoT TwinMaker. No entanto, a ação exige que o serviço tenha permissões concedidas por um perfil de serviço. Mary não tem permissões para passar o perfil para o serviço.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
iam:PassRole
```

Nesse caso, as políticas de Mary devem ser atualizadas para permitir que ela realize a ação `iam:PassRole`.

Se precisar de ajuda, entre em contato com seu AWS administrador. Seu administrador é a pessoa que forneceu suas credenciais de login.

## Quero permitir que pessoas fora da minha Conta da AWS acessem meus AWS IoT TwinMaker recursos

Você pode criar um perfil que os usuários de outras contas ou pessoas fora da organização podem usar para acessar seus recursos. É possível especificar quem é confiável para assumir o perfil. Para serviços que oferecem suporte a políticas baseadas em recursos ou listas de controle de acesso (ACLs), você pode usar essas políticas para conceder às pessoas acesso aos seus recursos.

Para saber mais, consulte:

- Para saber se é AWS IoT TwinMaker compatível com esses recursos, consulte [Como AWS IoT TwinMaker funciona com o IAM](#).

- Para saber como fornecer acesso aos seus recursos em todos os Contas da AWS que você possui, consulte Como [fornecer acesso a um usuário do IAM em outro Conta da AWS que você possui](#) no Guia do usuário do IAM.
- Para saber como fornecer acesso aos seus recursos a terceiros Contas da AWS, consulte Como [fornecer acesso Contas da AWS a terceiros](#) no Guia do usuário do IAM.
- Para saber como conceder acesso por meio da federação de identidades, consulte [Conceder acesso a usuários autenticados externamente \(federação de identidades\)](#) no Guia do usuário do IAM.
- Para saber a diferença entre perfis e políticas baseadas em recurso para acesso entre contas, consulte [Acesso a recursos entre contas no IAM](#) no Guia do usuário do IAM.

## Usar perfis vinculados a serviço do AWS IoT TwinMaker

AWS IoT TwinMaker usa funções [vinculadas ao serviço AWS Identity and Access Management](#) (IAM). Uma função vinculada ao serviço é um tipo exclusivo de função do IAM vinculada diretamente a. AWS IoT TwinMaker As funções vinculadas ao serviço são predefinidas AWS IoT TwinMaker e incluem todas as permissões que o serviço exige para chamar outros AWS serviços em seu nome.

Uma função vinculada ao serviço facilita a configuração AWS IoT TwinMaker porque você não precisa adicionar manualmente as permissões necessárias. AWS IoT TwinMaker define as permissões de suas funções vinculadas ao serviço e, a menos que seja definido de outra forma, só AWS IoT TwinMaker pode assumir suas funções. As permissões definidas incluem a política de confiança e a política de permissões, que não pode ser anexada a nenhuma outra entidade do IAM.

Um perfil vinculado ao serviço poderá ser excluído somente após excluir seus atributos relacionados. Isso protege seus AWS IoT TwinMaker recursos porque você não pode remover inadvertidamente a permissão para acessar os recursos.

Para obter informações sobre outros serviços que oferecem suporte a funções vinculadas a serviços, consulte [AWS Serviços que funcionam com IAM](#) e procure os serviços que têm Sim na coluna Funções vinculadas ao serviço. Escolha um Sim com um link para visualizar a documentação do perfil vinculado a esse serviço.

## Permissões de função vinculadas ao serviço para AWS IoT TwinMaker

AWS IoT TwinMaker usa a função vinculada ao serviço chamada `AWSServiceRoleForIoT` — Permite AWS IoT TwinMaker chamar outros AWS serviços e sincronizar seus recursos em seu nome.

A função vinculada ao serviço AWSService RoleForlo TTwin Maker confia nos seguintes serviços para assumir a função:

- `iottwinmaker.amazonaws.com`

A política de permissões de função nomeada AWSIoTTwinMakerServiceRolePolicy AWS IoT TwinMaker permite concluir as seguintes ações nos recursos especificados:

- Ação: `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` em `all your iotsitewise asset and asset-model resources`

Você deve configurar permissões para permitir que seus usuários, grupos ou perfis criem, editem ou excluam um perfil vinculado ao serviço. Para obter mais informações, consulte [Service-linked role permissions](#) (Permissões de nível vinculado a serviços) no Guia do usuário do IAM.

## Criação de uma função vinculada ao serviço para AWS IoT TwinMaker

Não é necessário criar manualmente um perfil vinculado ao serviço. Quando você sincroniza seus AWS IoT SiteWise ativos e modelos de ativos (sincronização de ativos) na AWS Management Console, na ou na AWS API AWS CLI, AWS IoT TwinMaker cria a função vinculada ao serviço para você.

Se excluir esse perfil vinculado ao serviço e precisar criá-lo novamente, será possível usar esse mesmo processo para recriar o perfil em sua conta. Quando você sincroniza seus AWS IoT SiteWise ativos e modelos de ativos (sincronização de ativos), AWS IoT TwinMaker cria a função vinculada ao serviço para você novamente.

Você também pode usar o console do IAM para criar uma função vinculada ao serviço com o caso de uso “IoT TwinMaker - Função gerenciada”. Na AWS CLI ou na AWS API, crie uma função vinculada ao serviço com o nome do `iottwinmaker.amazonaws.com` serviço. Para obter mais informações, consulte [Criar um perfil vinculado a serviço](#) no Guia do usuário do IAM. Se você excluir essa função vinculada ao serviço, será possível usar esse mesmo processo para criar a função novamente.

## Editando uma função vinculada ao serviço para AWS IoT TwinMaker

AWS IoT TwinMaker não permite que você edite a função vinculada ao serviço AWSService RoleForlo TTwin Maker. Depois que criar um perfil vinculado ao serviço, você não poderá alterar o nome do perfil, pois várias entidades podem fazer referência a ele. No entanto, será possível editar a descrição do perfil usando o IAM. Para obter mais informações, consulte [Editar um perfil vinculado ao serviço](#) no Guia do usuário do IAM.

## Excluindo uma função vinculada ao serviço para AWS IoT TwinMaker

Se você não precisar mais usar um recurso ou serviço que requer um perfil vinculado ao serviço, é recomendável excluí-lo. Dessa forma, você não tem uma entidade não utilizada que não seja monitorada ativamente ou mantida. No entanto, você deve limpar todos os espaços de trabalho vinculados a serviços que ainda estão usando sua função vinculada ao serviço antes de excluir manualmente a função.

### Note

Se o AWS IoT TwinMaker serviço estiver usando a função quando você tentar excluir os recursos, a exclusão poderá falhar. Se isso acontecer, espere alguns minutos e tente a operação novamente.

Como excluir manualmente o perfil vinculado ao serviço usando o IAM

Use o console do IAM AWS CLI, o ou a AWS API para excluir a função vinculada ao serviço AWSService RoleForlo TTwin Maker. Para obter mais informações, consulte [Excluir um perfil vinculado ao serviço](#) no Guia do usuário do IAM.

## Regiões suportadas para funções vinculadas a AWS IoT TwinMaker serviços

AWS IoT TwinMaker suporta o uso de funções vinculadas ao serviço em todas as regiões em que o serviço está disponível. Para obter mais informações, consulte [Regiões e endpoints da AWS](#).

## AWS políticas gerenciadas para AWS IoT TwinMaker

Para adicionar permissões a usuários, grupos e funções, é mais fácil usar políticas AWS gerenciadas do que escrever políticas você mesmo. É necessário tempo e experiência para criar [políticas](#)

[gerenciadas pelo cliente do IAM](#) que fornecem à sua equipe apenas as permissões de que precisam. Para começar rapidamente, você pode usar nossas políticas AWS gerenciadas. Essas políticas abrangem casos de uso comuns e estão disponíveis na sua Conta da AWS. Para obter mais informações sobre políticas AWS gerenciadas, consulte [políticas AWS gerenciadas](#) no Guia do usuário do IAM.

AWS os serviços mantêm e atualizam as políticas AWS gerenciadas. Você não pode alterar as permissões nas políticas AWS gerenciadas. Os serviços ocasionalmente acrescentam permissões adicionais a uma política gerenciada pela AWS para oferecer suporte a novos recursos. Esse tipo de atualização afeta todas as identidades (usuários, grupos e funções) em que a política está anexada. É mais provável que os serviços atualizem uma política gerenciada pela AWS quando um novo recurso for iniciado ou novas operações se tornarem disponíveis. Os serviços não removem as permissões de uma política AWS gerenciada, portanto, as atualizações de políticas não violarão suas permissões existentes.

Além disso, AWS oferece suporte a políticas gerenciadas para funções de trabalho que abrangem vários serviços. Por exemplo, a política ReadOnlyAccess AWS gerenciada fornece acesso somente de leitura a todos os AWS serviços e recursos. Quando um serviço lança um novo recurso, AWS adiciona permissões somente de leitura para novas operações e recursos. Para obter uma lista e descrições das políticas de perfis de trabalho, consulte [Políticas gerenciadas pela AWS para perfis de trabalho](#) no Guia do usuário do IAM.

## AWS política gerenciada: AWSIoTTwinMakerServiceRolePolicy

Você não pode se vincular AWSIoTTwinMakerServiceRolePolicy às suas entidades do IAM. Essa política é anexada a uma função vinculada ao serviço que permite que o realize ações em seu nome. Para obter mais informações, consulte [Permissões de função vinculadas ao serviço para AWS IoT TwinMaker](#).

A política de permissões de função nomeada AWSIoTTwinMakerServiceRolePolicy AWS IoT TwinMaker permite concluir as seguintes ações nos recursos especificados:

- Ação: `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` em all your `iotsitewise` asset and asset-model resources

```
{
  "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": [
          "IOTSITWISE"
        ]
      }
    }
  }
]
}

```

## AWS IoT TwinMaker atualizações nas políticas AWS gerenciadas

Veja detalhes sobre as atualizações das políticas AWS gerenciadas desde que esse serviço começou a rastrear essas alterações. Para receber alertas automáticos sobre alterações realizadas nesta página, assine o feed RSS na página de histórico do documento do .

Alteração	Descrição	Data
<p><a href="#">AWSIoT TwinMakerServiceRolePolicy</a>— Adicionou uma política</p>	<p>AWS IoT TwinMaker adicionou a política de permissões de função chamada AWSIoT TwinMakerServiceRolePolicy que AWS IoT TwinMaker permite concluir as seguintes ações nos recursos especificados:</p> <ul style="list-style-type: none"> <li>Ação: <code>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</code> em</li> </ul>	<p>17 de novembro de 2023</p>

Alteração	Descrição	Data
	<p>all your iotsitewise asset and asset-model resources</p> <p>Para obter mais informações, consulte <a href="#">Permissões de função vinculadas ao serviço para AWS IoT TwinMaker</a>.</p>	
começou a rastrear as alterações	começou a rastrear as mudanças em suas políticas AWS gerenciadas.	11 de maio de 2022

## AWS IoT TwinMaker e endpoints VPC de interface ( )AWS PrivateLink

Você pode estabelecer uma conexão privada entre a nuvem privada virtual (VPC) e o AWS IoT TwinMaker criando um endpoint da VPC de interface. Os endpoints de interface são alimentados por [AWS PrivateLink](#), que você pode usar para acessar de forma privada AWS IoT TwinMaker APIs sem um gateway de internet, dispositivo de tradução de endereços de rede (NAT), conexão VPN ou conexão AWS Direct Connect. As instâncias em sua VPC não precisam de endereços IP públicos para se comunicar. AWS IoT TwinMaker APIs Tráfego entre sua VPC e AWS IoT TwinMaker não sai da rede Amazon.

Cada endpoint de interface é representado por uma ou mais [Interfaces de Rede Elástica](#) nas sub-redes.

Para obter mais informações, consulte [Endpoints da VPC da interface \(AWS PrivateLink\)](#) no Manual do Usuário do Amazon VPC.

### Considerações sobre AWS IoT TwinMaker VPC endpoints

Antes de configurar uma interface para o VPC endpoint AWS IoT TwinMaker, revise as [propriedades e limitações do endpoint da interface no](#) Guia do usuário do Amazon VPC.

AWS IoT TwinMaker suporta fazer chamadas para todas as suas ações de API a partir de sua VPC.

- Para operações de API do plano de dados, use o seguinte endpoint:

```
data.iottwinmaker.region.amazonaws.com
```

As operações da API do plano de dados incluem o seguinte:

- [GetPropertyValue](#)
  - [GetPropertyValueHistory](#)
  - [BatchPutPropertyValues](#)
- Para as operações da API do ambiente de gerenciamento, use o seguinte endpoint:

```
api.iottwinmaker.region.amazonaws.com
```

As operações de API do ambiente de gerenciamento suportadas incluem o seguinte:

- [CreateComponentType](#)
- [CreateEntity](#)
- [CreateScene](#)
- [CreateWorkspace](#)
- [DeleteComponentType](#)
- [DeleteEntity](#)
- [DeleteScene](#)
- [DeleteWorkspace](#)
- [GetComponentType](#)
- [GetEntity](#)
- [GetScene](#)
- [GetWorkspace](#)
- [ListComponentTypes](#)
- [ListComponentTypes](#)
- [ListEntities](#)
- [ListScenes](#)

- [ListWorkspaces](#)
- [TagResource](#)
- [UntagResource](#)
- [UpdateComponentType](#)
- [UpdateEntity](#)
- [UpdateScene](#)
- [UpdateWorkspace](#)

## Criar um endpoint da VPC de interface para o AWS IoT TwinMaker

Você pode criar um VPC endpoint para o AWS IoT TwinMaker serviço usando o console Amazon VPC ou o (). AWS Command Line Interface AWS CLI Para obter mais informações, consulte [Criar um endpoint de interface](#) no Guia do usuário da Amazon VPC.

Crie um VPC endpoint para AWS IoT TwinMaker que use o seguinte nome de serviço.

- Para operações de API do plano de dados, use o seguinte nome de serviço:

```
com.amazonaws.region.iottwinmaker.data
```

- Para operações de API do ambiente de gerenciamento, use o seguinte nome de serviço:

```
com.amazonaws.region.iottwinmaker.api
```

Se você habilitar o DNS privado para o endpoint, poderá fazer solicitações de API AWS IoT TwinMaker usando seu nome DNS padrão para a região, por exemplo, `iottwinmaker.us-east-1.amazonaws.com`

Para mais informações, consulte [Acessar um serviço por um endpoint de interface](#) no Guia do usuário da Amazon VPC.

AWS IoT TwinMaker PrivateLink é compatível com as seguintes regiões:

- us-east-1

O ControlPlane serviço é suportado nas seguintes zonas de disponibilidade: use1-az1use1-az2, use1-az6 e.

O DataPlane serviço é suportado nas seguintes zonas de disponibilidade: use1-az1, use1-az2, use1-az4 e.

- us-west-2

Os DataPlane serviços ControlPlane e são suportados nas seguintes zonas de disponibilidade: usw2-az1, usw2-az2, usw2-az3 e.

- eu-west-1
- eu-central-1
- ap-southeast-1
- ap-southeast-2

Para obter mais informações sobre zonas de disponibilidade, consulte [Zona de disponibilidade IDs para seus AWS recursos - Resource Access Manager](#).

## Acessando AWS IoT TwinMaker por meio de uma interface VPC endpoint

Quando você cria um endpoint de interface, AWS IoT TwinMaker gera nomes de host DNS específicos do endpoint que você pode usar para se comunicar. AWS IoT TwinMaker A opção de DNS privado é habilitada opcionalmente. Para obter mais informações, consulte [Uso de zonas hospedadas privadas](#) no Guia do usuário do Amazon VPC.

Se você habilitar o DNS privado para o endpoint, poderá fazer solicitações de API para o AWS IoT TwinMaker por meio de um dos endpoints da VPC a seguir.

- Para as operações da API do plano de dados, use o seguinte endpoint. Substitua *region* pela sua região da AWS .

```
data.iottwinmaker.region.amazonaws.com
```

- Para as operações da API do ambiente de gerenciamento, use o seguinte endpoint. Substitua *region* pela sua região da AWS .

```
api.iottwinmaker.region.amazonaws.com
```

Se você desabilitar o DNS privado para o endpoint, deverá fazer o seguinte para acessar o AWS IoT TwinMaker por meio do endpoint:

- Especifique o URL do endpoint da VPC nas solicitações de API.
  - Para as operações da API do plano de dados, use o seguinte URL de endpoint. Substitua *vpc-endpoint-id* e *region* por seu ID de endpoint de VPC e sua região.

```
vpc-endpoint-id.data.iottwinmaker.region.vpce.amazonaws.com
```

- Para as operações da API do ambiente de gerenciamento, use o seguinte URL de endpoint. Substitua *vpc-endpoint-id* e *region* por seu ID de endpoint de VPC e sua região.

```
vpc-endpoint-id.api.iottwinmaker.region.vpce.amazonaws.com
```

- Desative a injeção do prefixo do host. O AWS CLI e AWS SDKs precede o endpoint do serviço com vários prefixos de host quando você chama cada operação de API. Isso faz com que o AWS CLI e AWS SDKs produza um valor inválido URLs para AWS IoT TwinMaker quando você especifica um VPC endpoint.

#### Important

Você não pode desativar a injeção do prefixo do host em AWS CLI ou Ferramentas da AWS para PowerShell. Isso significa que, se você desativou o DNS privado, não poderá usar AWS CLI ou Ferramentas da AWS para PowerShell acessar AWS IoT TwinMaker por meio do VPC endpoint. Se você quiser usar essas ferramentas para acessar AWS IoT TwinMaker por meio do endpoint, habilite o DNS privado.

Para obter mais informações sobre como desabilitar a injeção de prefixo de host no AWS SDKs, consulte as seções de documentação a seguir para 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 Ruby](#)

Para mais informações, consulte [Acessar um serviço por um endpoint de interface](#) no Guia do usuário da Amazon VPC.

## Criação de uma política de VPC endpoint para AWS IoT TwinMaker

É possível anexar uma política de endpoint ao endpoint da VPC que controla o acesso ao AWS IoT TwinMaker. Essa política especifica as seguintes informações:

- A entidade principal que pode realizar ações.
- As ações que podem ser realizadas.
- Os recursos aos quais as ações podem ser aplicadas.

Para mais informações, consulte [Controlar o acesso a serviços com VPC endpoints](#) no Guia do usuário da Amazon VPC.

Exemplo: política de VPC endpoint para ações AWS IoT TwinMaker

Veja a seguir um exemplo de uma política de endpoint para AWS IoT TwinMaker. Quando anexada a um endpoint, essa política concede acesso às AWS IoT TwinMaker ações listadas para o usuário do IAM `iottwinmakeradmin` na AWS conta `123456789012` em todos os recursos.

```
{
  "Statement": [
    {
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:user/role"
      },
      "Resource": "*",
      "Effect": "Allow",
      "Action": [
        "iottwinmaker:CreateEntity",
        "iottwinmaker:GetScene",
        "iottwinmaker:ListEntities"
      ]
    }
  ]
}
```

# Validação de conformidade para AWS IoT TwinMaker

Para saber se um AWS service (Serviço da AWS) está dentro do escopo de programas de conformidade específicos, consulte [Serviços da AWS Escopo por Programa de Conformidade](#) [Serviços da AWS](#) e escolha o programa de conformidade em que você está interessado. Para obter informações gerais, consulte Programas de [AWS conformidade Programas AWS](#) de .

Você pode baixar relatórios de auditoria de terceiros usando AWS Artifact. Para obter mais informações, consulte [Baixar relatórios em AWS Artifact](#) .

Sua responsabilidade de conformidade ao usar Serviços da AWS é determinada pela confidencialidade de seus dados, pelos objetivos de conformidade de sua empresa e pelas leis e regulamentações aplicáveis. AWS fornece os seguintes recursos para ajudar na conformidade:

- [Governança e conformidade de segurança](#): esses guias de implementação de solução abordam considerações sobre a arquitetura e fornecem etapas para implantar recursos de segurança e conformidade.
- [Referência de serviços qualificados para HIPAA](#): lista os serviços qualificados para HIPAA. Nem todos Serviços da AWS são elegíveis para a HIPAA.
- AWS Recursos de <https://aws.amazon.com/compliance/resources/> de conformidade — Essa coleção de pastas de trabalho e guias pode ser aplicada ao seu setor e local.
- [AWS Guias de conformidade do cliente](#) — Entenda o modelo de responsabilidade compartilhada sob a ótica da conformidade. Os guias resumem as melhores práticas de proteção Serviços da AWS e mapeiam as diretrizes para controles de segurança em várias estruturas (incluindo o Instituto Nacional de Padrões e Tecnologia (NIST), o Conselho de Padrões de Segurança do Setor de Cartões de Pagamento (PCI) e a Organização Internacional de Padronização (ISO)).
- [Avaliação de recursos com regras](#) no Guia do AWS Config desenvolvedor — O AWS Config serviço avalia o quão bem suas configurações de recursos estão em conformidade com as práticas internas, as diretrizes e os regulamentos do setor.
- [AWS Security Hub](#) — Isso AWS service (Serviço da AWS) fornece uma visão abrangente do seu estado de segurança interno AWS. O Security Hub usa controles de segurança para avaliar os recursos da AWS e verificar a conformidade com os padrões e as práticas recomendadas do setor de segurança. Para obter uma lista dos serviços e controles aceitos, consulte a [Referência de controles do Security Hub](#).
- [Amazon GuardDuty](#) — Isso AWS service (Serviço da AWS) detecta possíveis ameaças às suas cargas de trabalho Contas da AWS, contêineres e dados monitorando seu ambiente em busca

de atividades suspeitas e maliciosas. GuardDuty pode ajudá-lo a atender a vários requisitos de conformidade, como o PCI DSS, atendendo aos requisitos de detecção de intrusões exigidos por determinadas estruturas de conformidade.

- [AWS Audit Manager](#)— Isso AWS service (Serviço da AWS) ajuda você a auditar continuamente seu AWS uso para simplificar a forma como você gerencia o risco e a conformidade com as regulamentações e os padrões do setor.

## Resiliência em AWS IoT TwinMaker

A infraestrutura AWS global é construída em torno Regiões da AWS de zonas de disponibilidade. Regiões da AWS fornecem várias zonas de disponibilidade fisicamente separadas e isoladas, conectadas a redes de baixa latência, alta taxa de transferência e alta redundância. Com as zonas de disponibilidade, é possível projetar e operar aplicações e bancos de dados que automaticamente executam o failover entre as zonas sem interrupção. As zonas de disponibilidade são altamente disponíveis, tolerantes a falhas e escaláveis que uma ou várias infraestruturas de data center tradicionais.

Para obter mais informações sobre zonas de disponibilidade Regiões da AWS e zonas de disponibilidade, consulte [Infraestrutura AWS global](#).

Além da infraestrutura AWS global, AWS IoT TwinMaker oferece vários recursos para ajudar a suportar suas necessidades de resiliência e backup de dados.

## Segurança de infraestrutura em AWS IoT TwinMaker

Como serviço gerenciado, AWS IoT TwinMaker é protegido pelos procedimentos AWS globais de segurança de rede descritos no whitepaper [Amazon Web Services: Visão geral dos processos de segurança](#).

Você usa chamadas de API AWS publicadas para acessar AWS IoT TwinMaker pela rede. Os clientes devem oferecer suporte a Transport Layer Security (TLS) 1.2 ou posterior. Recomendamos usar o TLS 1.3 ou posterior. Os clientes também devem ter compatibilidade com conjuntos de criptografia com perfect forward secrecy (PFS) como Ephemeral Diffie-Hellman (DHE) ou Ephemeral Elliptic Curve Diffie-Hellman (ECDHE). A maioria dos sistemas modernos como Java 7 e versões posteriores oferece compatibilidade com esses modos.

Além disso, as solicitações devem ser assinadas usando um ID da chave de acesso e uma chave de acesso secreta associada a uma entidade principal do IAM. Ou você pode usar o [AWS](#)

---

[Security Token Service](#) (AWS STS) para gerar credenciais de segurança temporárias para assinar solicitações.

# Endpoints e cotas

## AWS IoT TwinMaker endpoints e cotas

Você pode encontrar informações sobre AWS IoT TwinMaker endpoints e cotas na Referência [AWS geral](#).

- Para obter mais informações sobre endpoints de serviço, consulte [Endpoints de serviço AWS IoT TwinMaker](#).
- Para obter informações sobre cotas, consulte [service quotas AWS IoT TwinMaker](#).
- Para obter informações sobre limites de API limitação, consulte limites de [AWS IoT TwinMaker API limitação](#).

## Informações adicionais sobre AWS IoT TwinMaker endpoints

Para se conectar programaticamente a AWS IoT TwinMaker, use um endpoint. Se você usar um HTTP cliente, precisará prefixar o plano de controle e o plano de dados da APIs seguinte forma. No entanto, não é necessário adicionar um prefixo aos AWS Command Line Interface comandos AWS SDK e porque eles adicionam automaticamente o prefixo necessário.

- Use o `api` prefixo para o plano APIs de controle. Por exemplo, `api.iottwinmaker.us-west-1.amazonaws.com`.
- Use o `data` prefixo para o plano APIs de dados. Por exemplo, `data.iottwinmaker.us-west-1.amazonaws.com`.

# Histórico do documento para o AWS IoT TwinMaker Guia do Usuário

A tabela a seguir descreve as versões de documentação para o AWS IoT TwinMaker.

Alteração	Descrição	Data
<a href="#">Novo perfil vinculado ao serviço e nova política do IAM</a>	AWS IoT TwinMaker adicionou uma nova função vinculada ao serviço, chamada. <a href="#">AWSServiceRoleForIoT TwinMaker</a> AWS IoT TwinMaker adicionou essa nova função vinculada ao serviço AWS IoT TwinMaker para permitir chamar outros AWS serviços e sincronizar seus recursos em seu nome. A nova política <code>AWSIoT TwinMakerServiceRolePolicy</code> do IAM está anexada a essa função, e a política concede permissão AWS IoT TwinMaker para chamar outros AWS serviços e sincronizar seus recursos em seu nome.	17 de novembro de 2023
<a href="#">Lançamento inicial</a>	A versão inicial do Guia do usuário AWS IoT TwinMaker	30 de novembro de 2021

As traduções são geradas por tradução automática. Em caso de conflito entre o conteúdo da tradução e da versão original em inglês, a versão em inglês prevalecerá.