



Guida per l'utente

# AWS IoT TwinMaker



# AWS IoT TwinMaker: Guida per l'utente

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

I marchi e l'immagine commerciale di Amazon non possono essere utilizzati in relazione a prodotti o servizi che non siano di Amazon, in una qualsiasi modalità che possa causare confusione tra i clienti o in una qualsiasi modalità che denigri o discrediti Amazon. Tutti gli altri marchi non di proprietà di Amazon sono di proprietà delle rispettive aziende, che possono o meno essere associate, collegate o sponsorizzate da Amazon.

---

# Table of Contents

Che cos'è AWS IoT TwinMaker? .....	1
Come funziona .....	1
Concetti e componenti chiave .....	2
Workspace .....	3
Modello entità-componente .....	3
Visualizzazione .....	5
Guida introduttiva con AWS IoT TwinMaker .....	8
Crea e gestisci un ruolo di servizio per AWS IoT TwinMaker .....	9
Assegna fiducia .....	9
Autorizzazioni di Amazon S3 .....	9
Assegna le autorizzazioni a un bucket Amazon S3 specifico .....	10
Autorizzazioni per i connettori integrati .....	12
Autorizzazioni per un connettore a una fonte di dati esterna .....	15
Modifica il ruolo IAM del tuo spazio di lavoro per utilizzare il connettore dati Athena .....	16
Creare un'area di lavoro .....	18
Crea la tua prima entità .....	20
Configurazione di un AWS account .....	23
Iscriviti per un Account AWS .....	23
Creazione di un utente amministratore .....	24
Utilizzo e creazione di tipi di componenti .....	26
Tipi di componenti incorporati .....	26
Caratteristiche principali dei tipi di AWS IoT TwinMaker componenti .....	27
Creazione di definizioni di proprietà .....	28
Creazione di funzioni .....	30
Esempi di tipi di componenti .....	30
Allarme (astratto) .....	30
Telemetria Timestream .....	32
Allarme (eredita dall'allarme astratto) .....	33
Esempi di apparecchiature .....	34
operazioni in blocco .....	37
Concetti e terminologia chiave .....	37
AWS IoT TwinMaker metadataTransferJob funzionalità .....	38
Esecuzione di operazioni di importazione ed esportazione in blocco .....	39
metadataTransferJob prerequisiti .....	40

Autorizzazioni IAM .....	40
Esegui un'operazione in blocco .....	44
Gestione degli errori .....	47
Importa modelli di metadati .....	48
AWS IoT TwinMaker metadataTransferJob esempi .....	51
AWS IoT TwinMaker schema del processo di trasferimento dei metadati .....	53
connettori dati .....	70
connettori dati .....	70
Connettore di inizializzazione dello schema .....	71
DataReaderByEntity .....	72
DataReaderByComponentType .....	73
DataReader .....	75
AttributePropertyValueReaderByEntity .....	76
DataWriter .....	77
Esempi .....	78
Connettore dati tabulare Athena .....	87
AWS IoT TwinMaker Prerequisiti del connettore dati Athena .....	87
Utilizzo del connettore dati Athena .....	88
Utilizzo del riferimento JSON del connettore dati tabulare Athena .....	92
Utilizzo del connettore dati Athena .....	93
Visualizza i dati tabulari di Athena in Grafana .....	94
AWS IoT TwinMaker connettore dati per serie temporali .....	96
AWS IoT TwinMaker prerequisiti del connettore dati per serie temporali .....	96
Sfondo del connettore dati delle serie temporali .....	97
Sviluppo di un connettore dati per serie temporali .....	98
Miglioramento del connettore dati .....	107
Test del connettore .....	108
Sicurezza .....	108
Creazione di risorse AWS IoT TwinMaker .....	108
Cosa c'è dopo .....	110
AWS IoT TwinMaker connettore dati cookie factory .....	110
Creazione di AWS IoT TwinMaker scene .....	116
Prima di creare scene .....	116
Ottimizza le tue risorse prima di importarle in AWS IoT TwinMaker .....	116
Le migliori pratiche per le prestazioni in AWS IoT TwinMaker .....	117
Ulteriori informazioni .....	117

Caricamento di risorse in AWS IoT TwinMaker .....	118
Carica i file nella Resource Library utilizzando la console .....	118
Crea le tue scene .....	118
Usa la navigazione 3D AWS IoT TwinMaker nelle tue scene .....	120
Aggiungi telecamere fisse .....	122
Modifica migliorata .....	122
Posizionamento mirato degli oggetti della scena .....	123
Selezione del sottomodello .....	123
Modifica le entità nella gerarchia delle scene .....	124
Aggiungere annotazioni alle entità .....	125
Aggiungi sovrapposizioni ai tag .....	129
Modifica le tue scene .....	137
Aggiungi modelli .....	137
Aggiungi widget .....	138
Aggiungere tag .....	142
Ottimizza il tuo modello 3D .....	142
Usare le tessere 3D nella scena .....	142
AWS IoT TwinMaker integrazione del kit di app .....	146
Cambia modalità di AWS IoT TwinMaker prezzo .....	147
grafico della conoscenza .....	149
AWS IoT TwinMaker concetti fondamentali del Knowledge Graph .....	149
Utilizzo del knowledge graph .....	150
Genera un grafico di scena .....	152
AWS IoT TwinMaker prerequisiti del grafico della scena .....	153
Associa i nodi 3D alla tua scena .....	154
Crea un'applicazione web .....	157
Grafico della conoscenza (pannello Grafana) .....	159
AWS IoT TwinMaker prerequisiti dell'editor di query .....	159
Autorizzazioni Knowledge Graph Grafana .....	159
Knowledge Graph, risorse aggiuntive .....	164
Sincronizzazione delle risorse con AWS IoT SiteWise .....	177
Utilizzo della sincronizzazione delle risorse con AWS IoT SiteWise .....	177
Utilizzo di un'area di lavoro personalizzata .....	177
Utilizzo dell'IoT SiteWiseDefaultWorkspace .....	181
Differenze tra aree di lavoro personalizzate e predefinite .....	182
Risorse sincronizzate da AWS IoT SiteWise .....	183

Spazi di lavoro personalizzati e predefiniti .....	183
Solo area di lavoro predefinita .....	184
Risorse non sincronizzate .....	185
Utilizza entità e tipi di componenti sincronizzati in AWS IoT TwinMaker .....	185
Analizza lo stato e gli errori di sincronizzazione .....	186
Sincronizza gli stati dei lavori .....	186
Eliminare un processo di sincronizzazione .....	188
Limiti di sincronizzazione degli asset .....	190
Configurazione delle dashboard Grafana .....	192
Configurazione CORS .....	193
Configurazione dell'ambiente Grafana .....	194
Grafana gestito da Amazon .....	194
Grafana autogestita .....	195
Creazione di un ruolo nella dashboard .....	196
Creazione di una policy IAM .....	196
Carica video dall'edge .....	200
Aggiungi altre autorizzazioni .....	200
Creazione del ruolo IAM di Grafana Dashboard .....	202
Creazione di una politica per i lettori AWS IoT TwinMaker video .....	202
Limita l'accesso alle tue risorse .....	204
Riduci i permessi GET .....	204
Ambita l'autorizzazione AWS IoT SiteWise BatchPutAssetPropertyValue .....	206
Connect Alarms ai dashboard Grafana .....	208
AWS IoT SiteWise prerequisiti per la configurazione degli allarmi .....	208
Definisci il ruolo IAM del componente di AWS IoT SiteWise allarme .....	208
Interroga e aggiorna tramite l'API AWS IoT TwinMaker .....	210
Configura la dashboard Grafana per gli allarmi .....	211
Usa la dashboard Grafana per la visualizzazione degli allarmi .....	213
Integrazione con Matterport .....	216
Panoramica dell'integrazione .....	217
Prerequisiti di integrazione con Matterport .....	218
Credenziali Matterport SDK .....	220
Memorizza le credenziali Matterport in AWS Secrets Manager .....	221
Matterport esegue scansioni nelle scene AWS IoT TwinMaker .....	224
Matterport nella dashboard di Grafana AWS IoT TwinMaker .....	230
Integrazione di Matterport con il kit di app AWS IoT .....	230

Streaming video su AWS IoT TwinMaker .....	231
Usa il connettore edge per Kinesis Video Stream per lo streaming di video AWS IoT TwinMaker .....	231
Prerequisiti .....	231
Crea componenti video per le scene AWS IoT TwinMaker .....	232
Aggiungi video e metadati dallo stream video di Kinesis a una dashboard Grafana .....	233
Usare la AWS IoT TwinMaker libreria Flink .....	234
Registrazione e monitoraggio .....	235
Monitoraggio con CloudWatch metriche Amazon .....	235
Parametri .....	236
Registrazione delle chiamate API con AWS CloudTrail .....	239
AWS IoT TwinMaker informazioni in CloudTrail .....	239
Sicurezza .....	241
Protezione dei dati .....	242
Crittografia dei dati a riposo .....	243
Crittografia in transito .....	243
Identity and Access Management .....	243
Destinatari .....	244
Autenticazione con identità .....	244
Gestione dell'accesso con policy .....	248
Come AWS IoT TwinMaker funziona con IAM .....	251
Esempi di policy basate su identità .....	258
Risoluzione dei problemi .....	261
Uso di ruoli collegati ai servizi .....	263
AWS politiche gestite .....	265
Endpoint VPC (AWS PrivateLink) .....	270
Considerazioni sugli endpoint AWS IoT TwinMaker VPC .....	270
Creazione di un endpoint VPC interfaccia per l' AWS IoT TwinMaker .....	272
Accesso AWS IoT TwinMaker tramite un endpoint VPC di interfaccia .....	273
Creazione di una policy per gli endpoint VPC per AWS IoT TwinMaker .....	275
Convalida della conformità .....	276
Resilienza .....	277
Sicurezza dell'infrastruttura .....	277
Punti finali e quote .....	278
AWS IoT TwinMaker endpoint e quote .....	278
Informazioni aggiuntive sugli endpoint .....	278

---

Cronologia dei documenti ..... 279  
..... cclxxx



# Che cos'è AWS IoT TwinMaker?

AWS IoT TwinMaker è un AWS IoT servizio che puoi utilizzare per creare gemelli digitali operativi di sistemi fisici e digitali. AWS IoT TwinMaker crea visualizzazioni digitali utilizzando misurazioni e analisi da una varietà di sensori, telecamere e applicazioni aziendali del mondo reale per aiutarvi a tenere traccia della vostra fabbrica fisica, edificio o impianto industriale. È possibile utilizzare questi dati del mondo reale per monitorare le operazioni, diagnosticare e correggere errori e ottimizzare le operazioni.

Un gemello digitale è una rappresentazione digitale in tempo reale di un sistema e di tutti i suoi componenti fisici e digitali. Viene aggiornato dinamicamente con dati per imitare la struttura, lo stato e il comportamento reali del sistema. Puoi usarlo per promuovere i risultati aziendali.

Gli utenti finali interagiscono con i dati del tuo digital twin utilizzando un'applicazione di interfaccia utente.

## Come funziona

Per soddisfare i requisiti minimi per la creazione di un gemello digitale, è necessario effettuare le seguenti operazioni.

- Modella dispositivi, apparecchiature, spazi e processi in un luogo fisico.
- Connect questi modelli a fonti di dati che memorizzano importanti informazioni contestuali, come i dati dei sensori, i feed delle telecamere.
- Crea visualizzazioni che aiutino gli utenti a comprendere i dati e le informazioni per prendere decisioni aziendali in modo più efficiente.
- Rendi disponibili i gemelli digitali agli utenti finali per promuovere i risultati aziendali.

AWS IoT TwinMaker affronta queste sfide fornendo le seguenti funzionalità.

- Entity Component System Knowledge Graph: AWS IoT TwinMaker fornisce strumenti per modellare dispositivi, apparecchiature, spazi e processi in un Knowledge Graph.

Questo knowledge graph contiene metadati sul sistema e può connettersi a dati in diverse posizioni. AWS IoT TwinMaker fornisce connettori integrati per i dati archiviati in AWS IoT SiteWise e Kinesis Video Streams. Puoi anche creare connettori personalizzati per i dati archiviati in altre posizioni.

Il knowledge graph e i connettori insieme forniscono un'unica interfaccia per interrogare i dati in posizioni diverse.

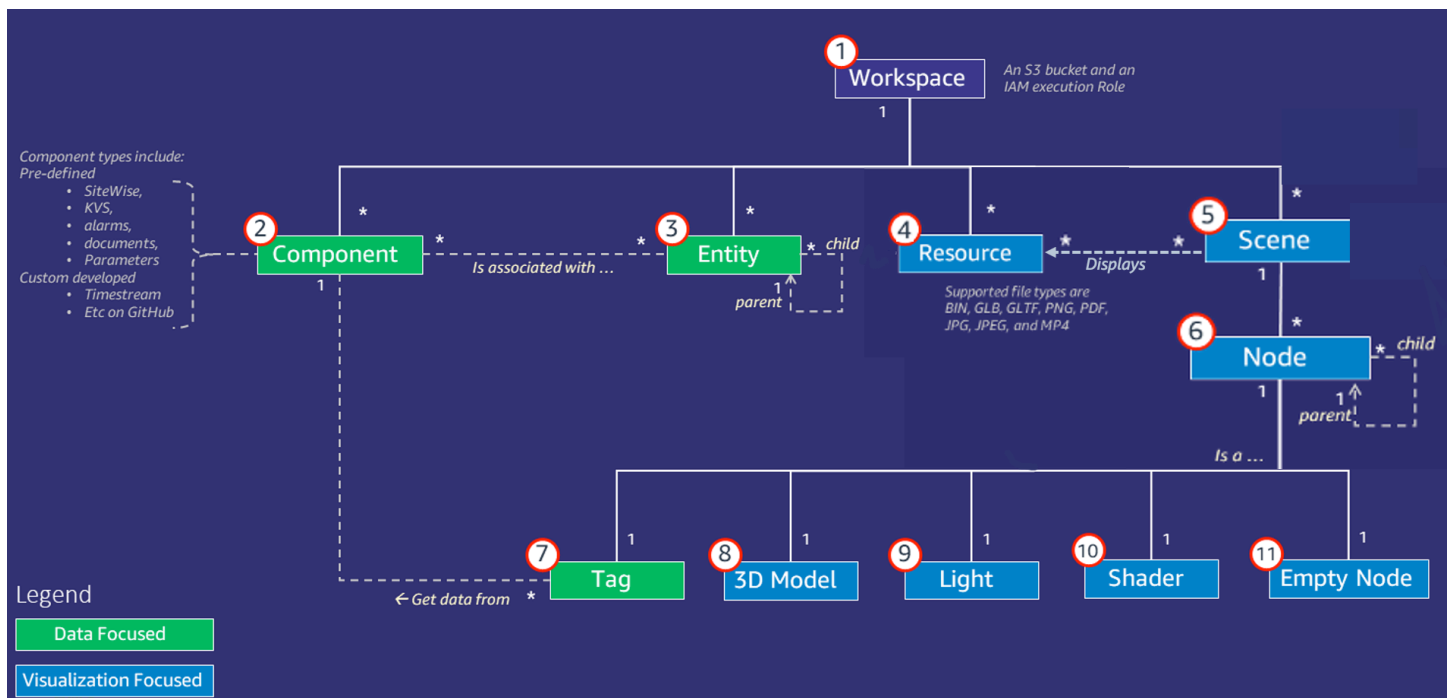
- **Scene Composer:** la AWS IoT TwinMaker console fornisce uno strumento di composizione delle scene per creare scene in 3D. Carichi i modelli 3D/CAD creati in precedenza, ottimizzati per la visualizzazione sul Web e convertiti in formato .gltf o .glb. Utilizzate quindi il compositore di scene per inserire più modelli in un'unica scena, creando rappresentazioni visive delle loro operazioni.

È inoltre possibile sovrapporre i dati nella scena. Ad esempio, è possibile creare un tag in una posizione della scena che si collega ai dati di temperatura di un sensore. Questo associa i dati alla posizione.

- **Applicazioni:** AWS IoT TwinMaker fornisce un plug-in per Grafana e Amazon Managed Grafana che puoi utilizzare per creare applicazioni dashboard per gli utenti finali.
- **Strumenti di terze parti:** Mendix collabora AWS IoT TwinMaker per fornire soluzioni complete per l'IoT industriale. Guarda il workshop [Lean Daily Management Application with Mendix e AWS IoT TwinMaker](#) per iniziare a utilizzare la Mendix Low Code Application Development Platform (LCAP) con AWS servizi come Kinesis AWS IoT TwinMaker Video Streams e. AWS IoT SiteWise

## Concetti e componenti chiave

Il diagramma seguente illustra come i concetti chiave di AWS IoT TwinMaker coincidono.



### Note

Gli asterischi (\*) nel diagramma indicano le relazioni. one-to-many [Per le quote per ciascuna di queste relazioni, vedi AWS IoT TwinMaker endpoint e quote.](#)

Le sezioni seguenti descrivono i concetti illustrati nel diagramma.

## Workspace

Un workspace è un contenitore di primo livello per l'applicazione digital twin. Crei un insieme logico di entità, componenti, risorse di scena e altre risorse per il tuo gemello digitale all'interno di questo spazio di lavoro. Funge anche da limite di sicurezza per gestire l'accesso all'applicazione digital twin e alle risorse in essa contenute. Ogni spazio di lavoro è collegato al bucket Amazon S3 in cui sono archiviati i dati dell'area di lavoro. Utilizzi i ruoli IAM per limitare l'accesso al tuo spazio di lavoro.

Uno spazio di lavoro può contenere più componenti, entità, scene e risorse. Un tipo di componente, un'entità, una scena o una risorsa esiste solo all'interno di un'area di lavoro.

## Modello entità-componente

AWS IoT TwinMaker fornisce strumenti che è possibile utilizzare per modellare il sistema utilizzando un entity-component-based Knowledge Graph. È possibile utilizzare l'architettura entity-component per creare una rappresentazione del sistema fisico. Questo modello di componenti di entità è costituito da entità, componenti e relazioni. Per ulteriori informazioni sui sistemi a componenti di entità, vedere Sistema di [componenti di entità](#).

## Entità

Le entità sono rappresentazioni digitali degli elementi di un gemello digitale che catturano le funzionalità di quell'elemento. Questo elemento può essere un'attrezzatura fisica, un concetto o un processo. Alle entità sono associati dei componenti. Questi componenti forniscono dati e contesto per l'entità associata.

Con AWS IoT TwinMaker, puoi organizzare le entità in gerarchie personalizzate per una gestione più efficiente. La visualizzazione predefinita dell'entità e del sistema di componenti è gerarchica.

## Componente

I componenti forniscono contesto e dati per le entità in una scena. Si aggiungono componenti alle entità. La durata di un componente è legata alla durata di un'entità.

I componenti possono aggiungere dati statici, come un elenco di documenti o le coordinate di una posizione geografica. Possono anche avere funzioni che si collegano ad altri sistemi, inclusi sistemi che contengono dati di serie AWS IoT SiteWise temporali come altre serie storiche del cloud.

I componenti sono definiti da documenti JSON che descrivono la connessione tra un'origine dati e AWS IoT TwinMaker. I componenti possono descrivere fonti di dati esterne o fonti di dati integrate in AWS IoT TwinMaker. Un componente accede a un'origine dati esterna utilizzando una funzione Lambda specificata nel documento JSON. Un'area di lavoro può contenere molti componenti. I componenti forniscono dati ai tag tramite entità associate.

AWS IoT TwinMaker fornisce diversi componenti integrati che è possibile aggiungere dalla console. Puoi anche creare componenti personalizzati per connetterti a fonti di dati come la telemetria del flusso temporale e le coordinate geospaziali. Ne sono un esempio la TimeStream telemetria, i componenti geospaziali e i connettori a fonti di dati di terze parti come Snowflake.

AWS IoT TwinMaker fornisce i seguenti tipi di componenti integrati per casi d'uso comuni:

- Documenti, ad esempio manuali utente o immagini che si trovano in URL specifici.
- Serie temporali, come i dati dei sensori provenienti da AWS IoT SiteWise.
- Allarmi, come allarmi con serie temporali provenienti da fonti di dati esterne.
- Video, da telecamere IP collegate a Kinesis Video Streams.
- Componenti personalizzati per la connessione a fonti di dati aggiuntive. Ad esempio, puoi creare un connettore personalizzato per connettere le tue AWS IoT TwinMaker entità a dati di serie temporali archiviati esternamente.

### Origini dati

Una fonte di dati è la posizione dei dati di origine del tuo gemello digitale. AWS IoT TwinMaker supporta due tipi di fonti di dati:

- Connettori gerarchici, che consentono di sincronizzare continuamente un modello esterno con AWS IoT TwinMaker

- Connettori per serie temporali, che consentono di connettersi a database di serie temporali come AWS IoT SiteWise

## Proprietà

Le proprietà sono i valori, statici e basati su serie temporali, contenuti nei componenti. Quando si aggiungono componenti alle entità, le proprietà del componente descrivono i dettagli sullo stato corrente dell'entità.

AWS IoT TwinMaker supporta tre tipi di proprietà:

- non-time-series Proprietà a valore singolo: queste proprietà sono in genere coppie chiave-valore statiche e vengono archiviate direttamente AWS IoT TwinMaker con i metadati dell'entità associata.
- Proprietà delle serie temporali: AWS IoT TwinMaker memorizza un riferimento all'archivio delle serie temporali per queste proprietà. L'impostazione predefinita è il valore più recente.
- Proprietà di relazione: queste proprietà memorizzano un riferimento a un'altra entità o componente. Ad esempio, `seen_by` è un componente di relazione che potrebbe mettere in relazione un'entità telecamera con un'altra entità che viene visualizzata direttamente da quella telecamera.

È possibile interrogare i valori delle proprietà su fonti di dati eterogenee utilizzando l'interfaccia di interrogazione dei dati unificata.

## Visualizzazione

Lo usi AWS IoT TwinMaker per aumentare una rappresentazione tridimensionale del tuo gemello digitale e poi visualizzarlo in Grafana. Per creare scene, utilizzate file CAD o altri tipi di file 3D esistenti. Puoi quindi utilizzare le sovrapposizioni di dati per aggiungere dati pertinenti per il tuo gemello digitale.

## Scene

Le scene sono rappresentazioni tridimensionali che forniscono un contesto visivo per i dati a cui sono collegati. AWS IoT TwinMaker Le scene possono essere create utilizzando un singolo modello 3D glTF (GL Transmission Format) o glb per l'intero ambiente o utilizzando una composizione di più modelli. Le scene includono anche tag per indicare i punti di interesse della scena.

Le scene sono i contenitori di primo livello per le visualizzazioni. Una scena è composta da uno o più nodi.

Un'area di lavoro può contenere più scene. Ad esempio, un'area di lavoro può contenere una scena per ogni piano di una struttura.

## Risorse

Le scene visualizzano le risorse, che vengono visualizzate come nodi nella AWS IoT TwinMaker console. Una scena può contenere molte risorse.

Le risorse sono immagini e modelli tridimensionali glTF basati su cui si basa la creazione di una scena. Una risorsa può rappresentare una singola apparecchiatura o un sito completo.

Puoi inserire le risorse in una scena caricando un file.gltf o .glb nella libreria di risorse del tuo spazio di lavoro e quindi aggiungendole alla scena.

## Interfaccia utente aumentata

Con AWS IoT TwinMaker puoi ampliare le tue scene con sovrapposizioni di dati che aggiungono contesto e informazioni importanti, come i dati dei sensori, alle posizioni della scena.

**Nodi:** i nodi sono istanze di tag, luci e modelli tridimensionali. Possono anche essere vuoti per aggiungere struttura alla gerarchia delle scene. Ad esempio, è possibile raggruppare più nodi in un unico nodo vuoto.

**Tag:** un tag è un tipo di nodo che rappresenta i dati provenienti da un componente (tramite un'entità). Un tag può essere associato a un solo componente. Un tag è un'annotazione aggiunta a una posizione di x, y, z coordinate specifica di una scena. Il tag collega questa parte della scena al knowledge graph utilizzando una proprietà di entità. È possibile utilizzare un tag per configurare il comportamento o l'aspetto visivo di un elemento nella scena, ad esempio un allarme.

**Luci:** puoi aggiungere luci a una scena per mettere a fuoco determinati oggetti o proiettare ombre sugli oggetti per indicarne la posizione fisica.

**Modelli tridimensionali:** un modello tridimensionale è una rappresentazione visiva di un file.gltf o .glb importato come risorsa.

### Note

AWS IoT TwinMaker non è destinato all'uso in o in associazione con il funzionamento di ambienti pericolosi o sistemi critici che possono causare gravi lesioni personali o morte o causare danni ambientali o materiali.

I dati raccolti tramite l'utilizzo di AWS IoT TwinMaker devono essere valutati per verificarne l'accuratezza in base al caso d'uso. AWS IoT TwinMaker non deve essere usato come sostituto del monitoraggio umano dei sistemi fisici allo scopo di valutare se tali sistemi funzionano in sicurezza.

# Guida introduttiva con AWS IoT TwinMaker

Gli argomenti di questa sezione descrivono come eseguire le seguenti operazioni.

- Crea e configura un nuovo spazio di lavoro.
- Crea un'entità e aggiungi un componente ad essa.

Prerequisiti:

Per creare il primo spazio di lavoro e la prima scena, sono necessarie le seguenti AWS risorse.

- Un [account AWS](#).
- Un ruolo di servizio IAM per AWS IoT TwinMaker. Questo ruolo viene generato automaticamente per impostazione predefinita, quando crei un nuovo AWS IoT TwinMaker spazio di lavoro nella [AWS IoT TwinMaker console](#).

Se scegli di non consentire la creazione AWS IoT TwinMaker automatica di un nuovo ruolo di servizio IAM, devi specificarne uno che hai già creato.

Per istruzioni sulla creazione e la gestione di questo ruolo di servizio, consulta [???](#).

Per ulteriori informazioni sui ruoli di servizio IAM, consulta [Creazione di un ruolo per delegare le autorizzazioni a un. Servizio AWS](#)

## Important

Questo ruolo di servizio deve avere una policy allegata che conceda l'autorizzazione al servizio di leggere e scrivere su un bucket Amazon S3. AWS IoT TwinMaker utilizza questo ruolo per accedere ad altri servizi per tuo conto. Dovrai inoltre assegnare una relazione di fiducia tra questo ruolo e AWS IoT TwinMaker in modo che il servizio possa assumere il ruolo. Se il tuo gemello interagisce con altri AWS servizi, aggiungi anche le autorizzazioni necessarie per tali servizi.

Argomenti

- [Crea e gestisci un ruolo di servizio per AWS IoT TwinMaker](#)
- [Creare un'area di lavoro](#)



- [Crea la tua prima entità](#)
- [Configurazione di un AWS account](#)

## Crea e gestisci un ruolo di servizio per AWS IoT TwinMaker

AWS IoT TwinMaker richiede l'utilizzo di un ruolo di servizio per consentirgli di accedere alle risorse di altri servizi per conto dell'utente. Questo ruolo deve avere un rapporto di fiducia con AWS IoT TwinMaker. Quando crei uno spazio di lavoro, devi assegnare questo ruolo all'area di lavoro. Questo argomento contiene criteri di esempio che mostrano come configurare le autorizzazioni per scenari comuni.

### Assegna fiducia

La seguente politica stabilisce una relazione di fiducia tra il tuo ruolo e AWS IoT TwinMaker. Assegna questa relazione di fiducia al ruolo che utilizzi per il tuo spazio di lavoro.

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

### Autorizzazioni di Amazon S3

La seguente politica consente al tuo ruolo di leggere, eliminare e scrivere in un bucket Amazon S3. Le aree di lavoro archiviano le risorse in Amazon S3, quindi le autorizzazioni di Amazon S3 sono necessarie per tutte le aree di lavoro.

```
{
```

```
"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 crei un'area di lavoro, AWS IoT TwinMaker crea un file nel tuo bucket Amazon S3 che indica che viene utilizzato da un'area di lavoro. Questa policy AWS IoT TwinMaker autorizza l'eliminazione di quel file quando elimini l'area di lavoro.

AWS IoT TwinMaker posiziona altri oggetti correlati al tuo spazio di lavoro. È responsabilità dell'utente eliminare questi oggetti quando si elimina un workspace.

## Assegna le autorizzazioni a un bucket Amazon S3 specifico

Quando crei uno spazio di lavoro nella AWS IoT TwinMaker console, puoi scegliere di AWS IoT TwinMaker creare un bucket Amazon S3 per te. Puoi trovare informazioni su questo bucket usando il seguente comando. AWS CLI

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

L'esempio seguente mostra il formato dell'output di questo 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"
}
```

*Per aggiornare la policy in modo da assegnare le autorizzazioni per uno specifico bucket Amazon S3, utilizza il valore del nome del bucket.*

La seguente politica consente al tuo ruolo di leggere, eliminare e scrivere in un bucket Amazon S3 specifico.

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

```

## Autorizzazioni per i connettori integrati

Se l'area di lavoro interagisce con altri AWS servizi utilizzando connettori integrati, è necessario includere le autorizzazioni per tali servizi in questa politica. Se utilizzi il tipo di componente `com.amazon.iotsitewise.connector`, devi includere le autorizzazioni per AWS IoT SiteWise. Per ulteriori informazioni sui tipi di componenti, [???](#) consulta.

### Note

Se interagisci con altri AWS servizi utilizzando un tipo di componente personalizzato, devi concedere al ruolo l'autorizzazione per eseguire la funzione Lambda che implementa la funzione nel tuo tipo di componente. Per ulteriori informazioni, consulta [???](#).

L'esempio seguente mostra come includere AWS IoT SiteWise nella politica.

```

{
  "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 utilizzi il tipo di componente `com.amazon.iotsitewise.connector` e devi leggere i dati delle proprietà da AWS IoT SiteWise esso, devi includere la seguente autorizzazione nella tua politica.

```

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

```

```

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

```

Se utilizzi il tipo di componente `com.amazon.iotsitewise.connector` e devi scrivere i dati delle proprietà su, devi includere la seguente autorizzazione nella tua politica. AWS IoT SiteWise

```

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

```

Se utilizzi il tipo di componente `com.amazon.iotsitewise.connector.edgevideo`, devi includere le autorizzazioni per Kinesis Video Streams. AWS IoT SiteWise Il seguente esempio di policy mostra come includere AWS IoT SiteWise le autorizzazioni Kinesis Video Streams nella policy.

```

...
{
  "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"
},
...

```

## Autorizzazioni per un connettore a una fonte di dati esterna

Se si crea un tipo di componente che utilizza una funzione che si connette a un'origine dati esterna, è necessario concedere al ruolo di servizio l'autorizzazione a utilizzare la funzione Lambda che implementa la funzione. Per ulteriori informazioni sulla creazione di tipi di componenti e funzioni, vedere. [???](#)

L'esempio seguente autorizza il ruolo di servizio dell'utente a utilizzare una funzione 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_*"
  ]
}
]
```

Per ulteriori informazioni sulla creazione di ruoli e sull'assegnazione di politiche e relazioni di fiducia ad essi utilizzando la console IAM, l'API e l'API IAM AWS CLI, consulta [Creazione di un ruolo per delegare le autorizzazioni](#) a un. Servizio AWS

## Modifica il ruolo IAM del tuo spazio di lavoro per utilizzare il connettore dati Athena

Per utilizzare il [connettore dati tabulare AWS IoT TwinMaker Athena](#), devi aggiornare il ruolo IAM del tuo AWS IoT TwinMaker workspace. Aggiungi le seguenti autorizzazioni al tuo ruolo IAM nel tuo spazio di lavoro:

### Note

Questa modifica IAM funziona solo per i dati tabulari Athena archiviati con Amazon AWS Glue S3. Per utilizzare Athena con altre fonti di dati, devi configurare un ruolo IAM per Athena, vedi [Gestione delle identità e degli accessi in 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
```

Leggi la [gestione delle identità e degli accessi in Athena](#) per ulteriori informazioni sulla configurazione di Athena IAM.

## Creare un'area di lavoro

Per creare e configurare il tuo primo spazio di lavoro, segui i passaggi seguenti.

### Note

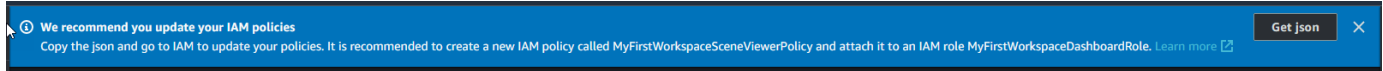
Questo argomento mostra come creare un'area di lavoro semplice con una singola risorsa. Per uno spazio di lavoro completo con più risorse, prova la configurazione di esempio nel repository Github degli [AWS IoT TwinMaker esempi](#).

1. Nella home page della [AWS IoT TwinMaker console](#), scegli Workspaces nel riquadro di navigazione a sinistra.
2. Nella pagina Aree di lavoro, scegli Crea area di lavoro.
3. Nella pagina Crea un'area di lavoro, inserisci un nome per l'area di lavoro.
4. (Facoltativo) Aggiungi una descrizione per il tuo spazio di lavoro.
5. In Risorsa S3, scegli Crea un bucket S3. Questa opzione crea un bucket Amazon S3 in cui AWS IoT TwinMaker archiviare informazioni e risorse relative all'area di lavoro. Ogni area di lavoro richiede il proprio bucket.
6. In Ruolo di esecuzione, scegli Genera automaticamente un nuovo ruolo o il ruolo IAM personalizzato che hai creato per questo spazio di lavoro.

Se scegli Genera automaticamente un nuovo ruolo, AWS IoT TwinMaker applica al ruolo una policy che concede l'autorizzazione al nuovo ruolo di servizio per accedere ad altri AWS servizi,

inclusa l'autorizzazione a leggere e scrivere nel bucket Amazon S3 specificato nel passaggio precedente. Per informazioni sull'assegnazione delle autorizzazioni a questo ruolo, consulta. [???](#)

7. Scegli Create Workspace (Crea WorkSpace). Il seguente banner viene visualizzato nella parte superiore della pagina Workspaces.



8. Scegli Get json. Ti consigliamo di aggiungere la policy IAM che vedi al ruolo IAM AWS IoT TwinMaker creato per gli utenti e gli account che visualizzano la dashboard di Grafana. Il nome di questo ruolo segue questo schema: *workspace-name*DashboardRole. Per istruzioni su come creare una policy e associarla a un ruolo, consulta [Modificare una policy di autorizzazioni di ruolo \(console\)](#).

L'esempio seguente contiene la policy da aggiungere al ruolo del dashboard.

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

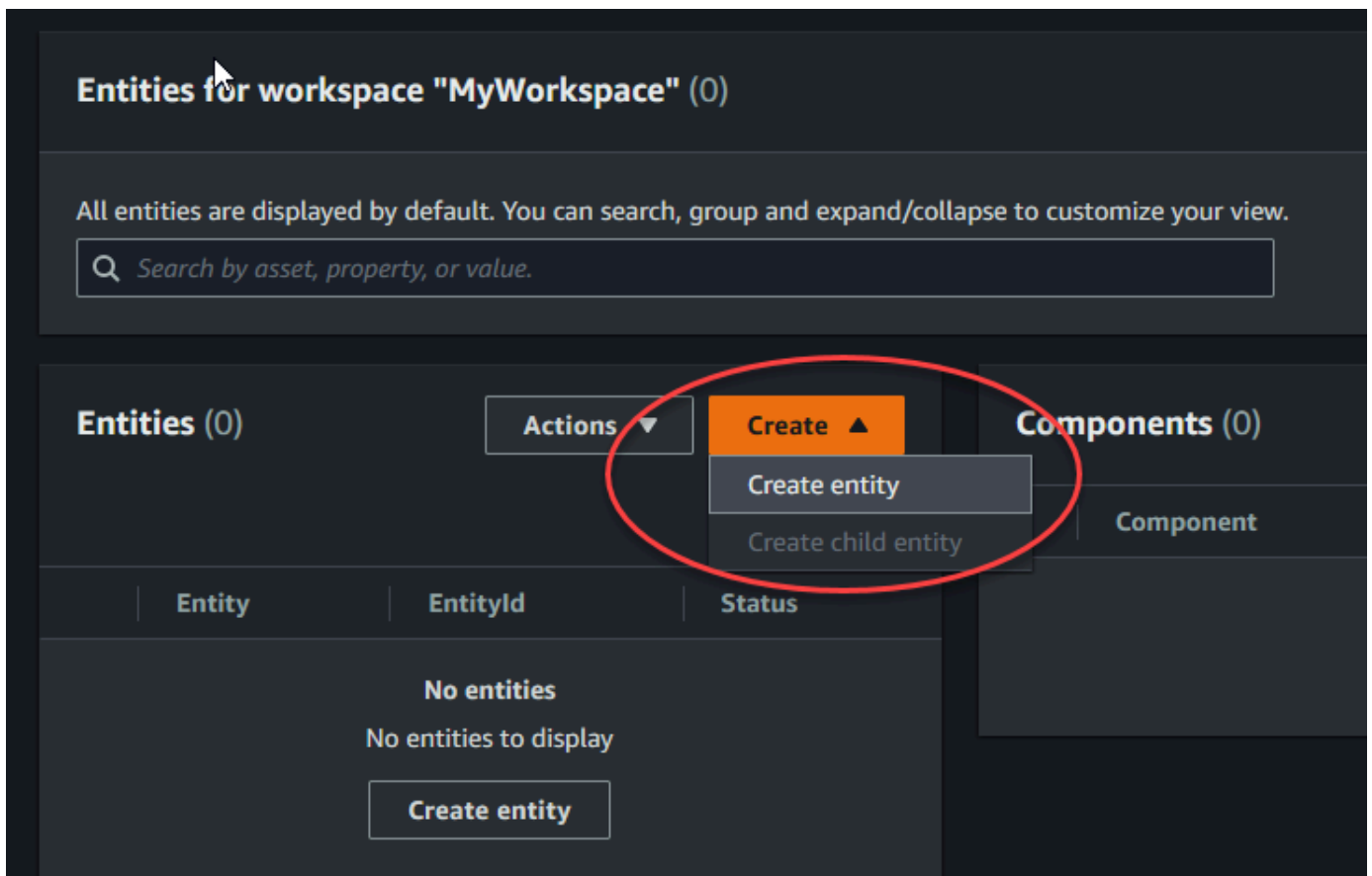
```
"Effect": "Allow",  
"Action": "iottwinmaker:ListWorkspaces",  
"Resource": "*" }  
]  
}
```

Ora sei pronto per iniziare a creare un modello di dati per il tuo spazio di lavoro con la tua prima entità. Per istruzioni su come eseguire questa operazione, consultare [Crea la tua prima entità](#).

## Crea la tua prima entità

Per creare la tua prima entità, segui i passaggi seguenti.

1. Nella pagina Aree di lavoro, scegli l'area di lavoro, quindi nel riquadro a sinistra scegli Entità.
2. Nella pagina Entità, scegli Crea, quindi scegli Crea entità.



3. Nella finestra Crea un'entità, inserisci un nome per la tua entità. Questo esempio utilizza un'**CookieMixer**entità.
4. (Facoltativo) Inserisci una descrizione per l'entità.
5. Scegli Crea entità,

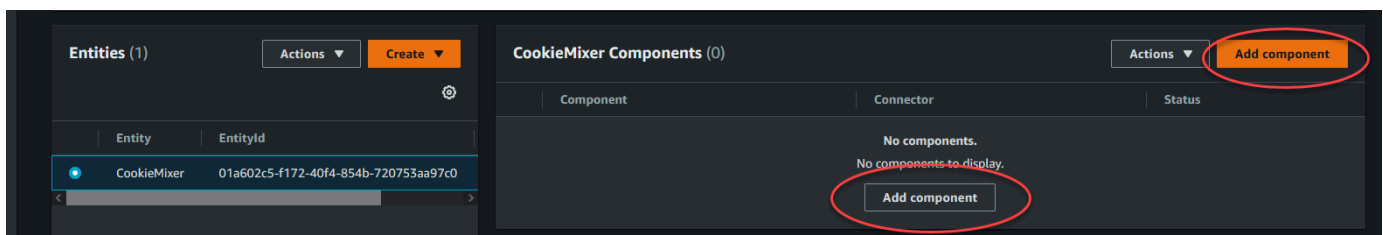
Le entità contengono dati su ogni elemento del tuo spazio di lavoro. I dati vengono inseriti nelle entità aggiungendo componenti. AWS IoT TwinMaker fornisce i seguenti tipi di componenti integrati.

- Parametri: aggiunge un set di proprietà chiave-valore.
- Documento: aggiunge un nome e un URL per un documento che contiene informazioni sull'entità.
- Allarmi: si collega a una fonte di dati della serie temporale degli allarmi.
- SiteWise connettore: richiama le proprietà delle serie temporali definite in una risorsa. AWS IoT SiteWise
- Edge Connector per Kinesis Video Streams: estrae i dati AWS IoT Greengrass video da Edge Connector per KVS. AWS IoT Greengrass Per ulteriori informazioni, consulta [AWS IoT TwinMakerintegrazione video](#).

Puoi vedere questi tipi di componenti e le loro definizioni selezionando Tipi di componenti nel riquadro a sinistra. È inoltre possibile creare un nuovo tipo di componente nella pagina Tipi di componenti. Per ulteriori informazioni sulla creazione di tipi di componenti, vedere [Utilizzo e creazione di tipi di componenti](#).

In questo esempio, creiamo un semplice componente del documento che aggiunge informazioni descrittive sull'entità.

1. Nella pagina Entità, scegli l'entità, quindi scegli aggiungi componente.



2. Nella finestra Aggiungi componente, inserisci un nome per il componente. Poiché questo esempio utilizza un'entità cookie mixer, inseriamo **MixerDescription** nel 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. Scegli Aggiungi un documento, quindi inserisci i valori per il nome del documento e l'URL esterno. Con il componente documenti, puoi memorizzare un elenco di URL esterni che contengono informazioni importanti sull'entità.

#### 4. Scegli Aggiungi componente.

Ora sei pronto per creare la tua prima scena. Per istruzioni su come eseguire questa operazione, consultare [Creazione e modifica di AWS IoT TwinMaker scene](#).

## Configurazione di un AWS account

Se non ne hai uno Account AWS, completa i seguenti passaggi per crearne uno.

Per iscriverti a un Account AWS

1. Apri la pagina all'indirizzo <https://portal.aws.amazon.com/billing/signup>.
2. Segui le istruzioni online.

Nel corso della procedura di registrazione riceverai una telefonata, durante la quale sarà necessario inserire un codice di verifica attraverso la tastiera del telefono.

Quando ti iscrivi a un Account AWS, Utente root dell'account AWS viene creato un. L'utente root dispone dell'accesso a tutte le risorse e tutti i Servizi AWS nell'account. Come best practice di sicurezza, [assegna l'accesso amministrativo a un utente amministrativo](#) e utilizza solo l'utente root per eseguire [attività che richiedono l'accesso di un utente root](#).

## Iscriviti per un Account AWS

Se non ne hai uno Account AWS, completa i seguenti passaggi per crearne uno.

Per iscriverti a un Account AWS

1. Apri la pagina all'indirizzo <https://portal.aws.amazon.com/billing/signup>.
2. Segui le istruzioni online.

Nel corso della procedura di registrazione riceverai una telefonata, durante la quale sarà necessario inserire un codice di verifica attraverso la tastiera del telefono.

Quando ti iscrivi a un Account AWS, Utente root dell'account AWS viene creato un. L'utente root dispone dell'accesso a tutte le risorse e tutti i Servizi AWS nell'account. Come best practice di sicurezza, [assegna l'accesso amministrativo a un utente amministrativo](#) e utilizza solo l'utente root per eseguire [attività che richiedono l'accesso di un utente root](#).

AWS ti invia un'email di conferma dopo il completamento della procedura di registrazione. È possibile visualizzare l'attività corrente dell'account e gestire l'account in qualsiasi momento accedendo all'indirizzo <https://aws.amazon.com/> e selezionando Il mio account.

## Creazione di un utente amministratore

Dopo la registrazione Account AWS, proteggi Utente root dell'account AWS AWS IAM Identity Center, abilita e crea un utente amministrativo in modo da non utilizzare l'utente root per le attività quotidiane.

Proteggi i tuoi Utente root dell'account AWS

1. Accedi [AWS Management Console](#) come proprietario dell'account scegliendo Utente root e inserendo il tuo indirizzo Account AWS email. Nella pagina successiva, inserisci la password.

Per informazioni sull'accesso utilizzando un utente root, consulta la pagina [Signing in as the root user](#) della Guida per l'utente di Accedi ad AWS .

2. Abilita l'autenticazione a più fattori (MFA) per l'utente root.

Per istruzioni, consulta [Abilitare un dispositivo MFA virtuale per l'utente Account AWS root \(console\)](#) nella Guida per l'utente IAM.

Creazione di un utente amministratore

1. Abilita Centro identità IAM.

Per istruzioni, consulta [Abilitazione di AWS IAM Identity Center](#) nella Guida per l'utente di AWS IAM Identity Center .

2. In IAM Identity Center, assegna l'accesso amministrativo a un utente amministratore.

Per un tutorial sull'utilizzo di IAM Identity Center directory come fonte di identità, consulta [Configurare l'accesso utente con l'impostazione predefinita IAM Identity Center directory](#) nella Guida per l'AWS IAM Identity Center utente.

Accesso come utente amministratore

- Per accedere con l'utente IAM Identity Center, utilizza l'URL di accesso che è stato inviato al tuo indirizzo e-mail quando hai creato l'utente IAM Identity Center.



Per informazioni sull'accesso utilizzando un utente IAM Identity Center, consulta [AWS Accedere al portale di accesso](#) nella Guida per l'Accedi ad AWS utente.

# Utilizzo e creazione di tipi di componenti

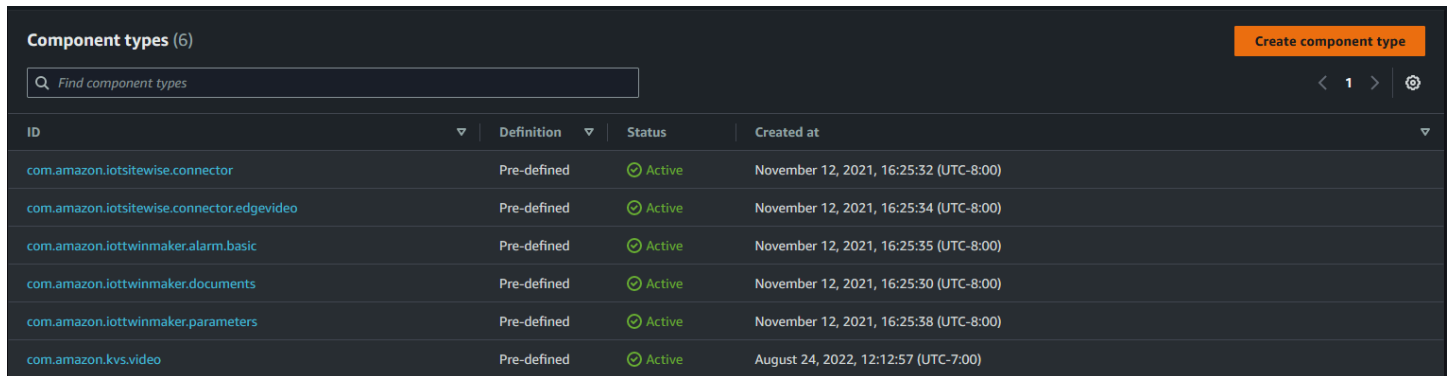
Questo argomento illustra i valori e le strutture utilizzati per creare un tipo di AWS IoT TwinMaker componente. Viene illustrato come creare un oggetto di richiesta che è possibile passare all'[CreateComponentType](#) API o utilizzando l'editor dei tipi di componenti nella AWS IoT TwinMaker console.

I componenti forniscono il contesto per le proprietà e i dati delle entità associate.

## Tipi di componenti incorporati

Nella AWS IoT TwinMaker console, quando si sceglie un'area di lavoro e quindi si sceglie Tipi di componenti nel riquadro sinistro, vengono visualizzati i seguenti tipi di componenti.

- `com.amazon.iotsitewise.resourcesync`: un tipo di componente che sincronizza automaticamente gli asset e i modelli di asset e li converte in entità, componenti e tipi di componenti. AWS IoT SiteWise AWS IoT TwinMaker Per ulteriori informazioni sull'utilizzo della sincronizzazione delle risorse, AWS IoT SiteWise consulta Sincronizzazione AWS IoT SiteWise delle [risorse](#) con.
- `com.amazon.iottwinmaker.alarm.basic`: un componente di allarme di base che estrae i dati di allarme da una fonte esterna a un'entità. Questo componente non contiene una funzione che si connette a una fonte di dati specifica. Ciò significa che il componente di allarme è astratto e può essere ereditato da un altro tipo di componente che specifica una fonte di dati e una funzione che legge da tale fonte.
- `com.amazon.iottwinmaker.documents`: una semplice mappatura dei titoli agli URL per i documenti che contengono informazioni su un'entità.
- `com.amazon.iotsitewise.connector.edgevideo`: un componente che estrae video da un dispositivo IoT utilizzando il componente Edge Connector for Kinesis Video Streams in un'entità. AWS IoT Greengrass Il componente [Edge Connector for Kinesis AWS IoT Greengrass Video](#) Streams AWS IoT TwinMaker non è un componente, ma piuttosto AWS IoT Greengrass un componente precostruito che viene distribuito localmente sul tuo dispositivo IoT.
- `com.amazon.iotsitewise.connector`: un componente che inserisce i dati in un'entità. AWS IoT SiteWise
- `com.amazon.iottwinmaker.parameters`: un componente che aggiunge coppie chiave-valore statiche a un'entità.
- `com.amazon.kvs.video`: un componente che trasferisce i video da Kinesis Video Streams in un'entità. AWS IoT TwinMaker



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

## Caratteristiche principali dei tipi di AWS IoT TwinMaker componenti

L'elenco seguente descrive le caratteristiche principali dei tipi di componenti.

- Definizioni delle proprietà: l'[PropertyDefinitionRequest](#) oggetto definisce una proprietà che è possibile inserire nel compositore di scene oppure può essere popolata con dati estratti da fonti di dati esterne. Le proprietà statiche impostate vengono memorizzate in AWS IoT TwinMaker. Le proprietà delle serie temporali e altre proprietà estratte dalle fonti di dati vengono archiviate esternamente.

Le definizioni delle proprietà vengono specificate all'interno di una stringa nella mappa.

PropertyDefinitionRequest Ogni stringa deve essere unica per la mappa.

- Funzioni: l'[FunctionRequest](#) oggetto specifica una funzione Lambda che legge e potenzialmente scrive su un'origine dati esterna.

Un tipo di componente che contiene una proprietà con un valore memorizzato esternamente ma che non dispone di una funzione corrispondente per recuperare i valori è un tipo di componente astratto. È possibile estendere tipi di componenti concreti da un tipo di componente astratto. Non è possibile aggiungere tipi di componenti astratti a un'entità. Non compaiono nel compositore di scene.

Si specificano le funzioni all'interno di una stringa da FunctionRequest mappare. La stringa deve specificare uno dei seguenti tipi di funzione predefiniti.

- `dataReader`: Una funzione che estrae dati da una fonte esterna.
- `dataReaderByEntity`: Una funzione che estrae dati da una fonte esterna.

Quando si utilizza questo tipo di lettore di dati, l'operazione [GetPropertyValueHistoryAPI](#) supporta solo query specifiche dell'entità per le proprietà di questo tipo di componente. (Puoi richiedere la cronologia dei valori della proprietà solo per `componentName +.`) `entityId`

- `dataReaderByComponentType`: Una funzione che estrae dati da una fonte esterna.

Quando si utilizza questo tipo di lettore di dati, l'operazione [GetPropertyValueHistoryAPI](#) supporta solo query tra entità per le proprietà di questo tipo di componente. (È possibile richiedere solo la cronologia dei valori della proprietà per `componentTypeId`.)

- `dataWriter`: Una funzione che scrive dati su una fonte esterna.
- `schemaInitializer`: Una funzione che inizializza automaticamente i valori delle proprietà ogni volta che si crea un'entità che contiene il tipo di componente.

Uno dei tre tipi di funzioni di lettura dei dati è richiesto in un tipo di componente non astratto.

[Per un esempio di funzione Lambda che implementa componenti di telemetria del flusso temporale, inclusi gli allarmi, consulta il lettore di dati in Samples.AWS IoT TwinMaker](#)

#### Note

Poiché il connettore di allarme eredita dal tipo di componente di allarme astratto, la funzione Lambda deve restituire `alarm_key` il valore. Se non restituisci questo valore, Grafana non lo riconoscerà come allarme. Questo è necessario per tutti i componenti che restituiscono allarmi.

- Ereditarietà: i tipi di componenti promuovono la riutilizzabilità del codice attraverso l'ereditarietà. Un tipo di componente può ereditare fino a 10 tipi di componenti principali.

Utilizzate il `extendsFrom` parametro per specificare i tipi di componenti da cui il tipo di componente eredita proprietà e funzioni.

- `isSingleton`: alcuni componenti contengono proprietà, come le coordinate di posizione, che non possono essere incluse più di una volta in un'entità. Imposta il valore del `isSingleton` parametro su `true` per indicare che il tipo di componente può essere incluso una sola volta in un'entità.

## Creazione di definizioni di proprietà

La tabella seguente descrive i parametri di `aPropertyDefinitionRequest`.

Parametro	Descrizione
<code>isExternalId</code>	<p>Un valore booleano che specifica se la proprietà è un identificatore univoco (ad esempio un AWS IoT SiteWise asset Id) del valore di una proprietà memorizzato esternamente.</p> <p>Il valore predefinito di questa proprietà è <code>false</code>.</p>
<code>isStoredExternally</code>	<p>Un valore booleano che specifica se il valore della proprietà è memorizzato esternamente.</p> <p>Il valore predefinito di questa proprietà è <code>false</code>.</p>
<code>isTimeSeries</code>	<p>Un valore booleano che specifica se la proprietà memorizza dati di serie temporali.</p> <p>Il valore predefinito di questa proprietà è <code>false</code>.</p>
<code>isRequiredInEntity</code>	<p>Un valore booleano che specifica se la proprietà deve avere un valore in un'entità che utilizza il tipo di componente.</p>
<code>dataType</code>	<p>Un <a href="#">DataType</a> oggetto che specifica il tipo di dati (ad esempio stringa, mappa, elenco e unità di misura) della proprietà.</p>
<code>defaultValue</code>	<p>Un <a href="#">DataValue</a> oggetto che specifica il valore predefinito della proprietà.</p>
<code>configuration</code>	<p>Una string-to-string mappa che specifica le informazioni aggiuntive necessarie per la connessione a un'origine dati esterna.</p>

# Creazione di funzioni

La tabella seguente descrive i parametri di `aFunctionRequest`.

Parametro	Descrizione
<code>implementedBy</code>	Un <a href="#">DataConnector</a> oggetto che specifica la funzione Lambda che si connette all'origine dati esterna.
<code>requiredProperties</code>	Un elenco di proprietà di cui la funzione ha bisogno per leggere e scrivere su un'origine dati esterna.
<code>scope</code>	L'ambito della funzione. Da utilizzare <code>Workspace</code> per funzioni con un ambito che si estende su un intero spazio di lavoro. Utilizzare <code>Entity</code> per funzioni con un ambito limitato all'entità che contiene il componente.

Per esempi che mostrano come creare ed estendere i tipi di componenti, vedere [???](#).

## Esempi di tipi di componenti

Questo argomento contiene esempi che mostrano come implementare i concetti chiave dei tipi di componenti.

### Allarme (astratto)

L'esempio seguente è il tipo di componente di allarme astratto che appare nella AWS IoT TwinMaker console. Contiene un `functions` elenco composto da un `dataReader` che non ha `implementedBy` valore.

```
{
  "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
  }
}
}
```

**Note:**

I valori per `componentTypeId` e `workspaceId` sono obbligatori. Il valore di `componentTypeId` deve essere univoco per l'area di lavoro. Il valore di `alarm_key` è un identificatore univoco che una funzione può utilizzare per recuperare i dati di allarme da una fonte esterna. Il valore della chiave è obbligatorio e memorizzato in AWS IoT TwinMaker. I valori delle serie `alarm_status` temporali vengono memorizzati nella fonte esterna.

Altri esempi sono disponibili nella sezione [AWS IoT TwinMaker Esempi](#).

## Telemetria Timestream

L'esempio seguente è un tipo di componente semplice che recupera i dati di telemetria su un tipo specifico di componente (come un allarme o un cookie mixer) da una fonte esterna. Specifica una funzione Lambda ereditata dai tipi di componenti.

```
{
  "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
    }
  }
}
```



```
}  
}
```

## Allarme (eredita dall'allarme astratto)

L'esempio seguente eredita sia dall'allarme astratto che dal tipo di componente di telemetria timestream. Specifica la propria funzione Lambda che recupera i dati di allarme.

```
{  
  "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

Poiché il connettore di allarme eredita dal tipo di componente di allarme astratto, la funzione Lambda deve restituire `alarm_key` il valore. Se non restituisci questo valore, Grafana non

lo riconoscerà come allarme. Questo è necessario per tutti i componenti che restituiscono allarmi.

## Esempi di apparecchiature

Gli esempi in questa sezione mostrano come modellare potenziali apparecchiature. È possibile utilizzare questi esempi per avere alcune idee su come modellare le apparecchiature nei propri processi.

### Miscelatore per biscotti

L'esempio seguente eredita dal tipo di componente di telemetria timestream. Specifica proprietà di serie temporali aggiuntive per la velocità di rotazione e la temperatura di un mixer per biscotti.

```
{
  "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
    }
  }
}
```

## Serbatoio d'acqua

L'esempio seguente eredita dal tipo di componente di telemetria timestream. Specifica proprietà di serie temporali aggiuntive per il volume e la portata di un serbatoio d'acqua.

```
{
  "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
    }
  }
}
```

## Ubicazione nello spazio

L'esempio seguente contiene proprietà, i cui valori sono memorizzati in AWS IoT TwinMaker. Poiché i valori sono specificati dagli utenti e memorizzati internamente, non è richiesta alcuna funzione per recuperarli. L'esempio utilizza il tipo di RELATIONSHIP dati anche per specificare una relazione con un altro tipo di componente.

Questo componente fornisce un meccanismo leggero per aggiungere contesto a un gemello digitale. Puoi usarlo per aggiungere metadati che indicano dove si trova qualcosa. È inoltre possibile utilizzare queste informazioni nella logica utilizzata per determinare quali telecamere possono vedere un'attrezzatura o uno spazio o per sapere come inviare qualcuno in un luogo.

```
{
  "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 operazioni di massa

Usa `metadataTransferJob` a per trasferire e gestire le tue AWS IoT TwinMaker risorse su larga scala. A `metadataTransferJob` consente di eseguire operazioni di massa e trasferire risorse tra AWS IoT TwinMaker e AWS IoT SiteWise e Amazon S3.

Puoi utilizzare operazioni in blocco nei seguenti scenari:

- Migrazione di massa di risorse e dati tra account, ad esempio migrazione da un account di sviluppo a un account di produzione.
- Gestione delle risorse su larga scala, come il caricamento e la modifica delle AWS IoT risorse su larga scala.
- Importazione di massa delle tue risorse in AWS IoT TwinMaker e AWS IoT SiteWise.
- Importazione in blocco di AWS IoT TwinMaker entità da file ontologici esistenti come file `revit` o `BIM`.

## Argomenti

- [Concetti e terminologia chiave](#)
- [Esecuzione di operazioni di importazione ed esportazione in blocco](#)
- [AWS IoT TwinMaker schema del processo di trasferimento dei metadati](#)

## Concetti e terminologia chiave

AWS IoT TwinMaker le operazioni di massa utilizzano i seguenti concetti e la terminologia seguenti:

- **Importazione:** l'azione di spostare le risorse in uno AWS IoT TwinMaker spazio di lavoro. Ad esempio, da un file locale, un file in un bucket Amazon S3 o da uno spazio AWS IoT SiteWise di lavoro. AWS IoT TwinMaker
- **Esportazione:** l'azione di spostare le risorse da un' AWS IoT TwinMaker area di lavoro a una macchina locale o a un bucket Amazon S3.
- **Fonte:** la posizione di partenza da cui desideri spostare le risorse.

Ad esempio, un bucket Amazon S3 è una fonte di importazione e un AWS IoT TwinMaker workspace è una fonte di esportazione.

- **Destinazione:** la posizione desiderata in cui vuoi spostare le tue risorse.

Ad esempio, un bucket Amazon S3 è una destinazione di esportazione e un AWS IoT TwinMaker workspace è una destinazione di importazione.

- **AWS IoT SiteWise Schema:** schema utilizzato per importare ed esportare risorse da e verso. AWS IoT SiteWise
- **AWS IoT TwinMaker Schema:** schema utilizzato per importare ed esportare risorse da e verso AWS IoT TwinMaker.
- **AWS IoT TwinMaker risorse di primo livello:** risorse utilizzate nelle API esistenti. In particolare, un'entità o un. `ComponentType`
- **AWS IoT TwinMaker risorse di livello secondario:** tipi di risorse annidate utilizzati nelle definizioni dei metadati. In particolare, un componente.
- **Metadati:** informazioni chiave necessarie per importare o esportare correttamente AWS IoT TwinMaker le risorse AWS IoT SiteWise e le risorse.
- `metadataTransferJob`: l'oggetto creato durante l'esecuzione `CreateMetadataTransferJob`.

## AWS IoT TwinMaker metadataTransferJob funzionalità

Questo argomento spiega il comportamento che AWS IoT TwinMaker segue quando si esegue un'operazione in blocco, ovvero come `metadataTransferJob` viene elaborato un. Spiega inoltre come definire uno schema con i metadati necessari per trasferire le risorse. AWS IoT TwinMaker le operazioni in blocco supportano le seguenti funzionalità:

- **Creazione o sostituzione di risorse di primo livello:** AWS IoT TwinMaker creerà nuove risorse o sostituirà tutte le risorse esistenti identificate in modo univoco da un ID di risorsa.

Ad esempio, se nel sistema esiste un'entità, la definizione dell'entità verrà sostituita da quella nuova definita nel modello sotto la `Entity` chiave.

- **Creazione o sostituzione della risorsa secondaria:**

Dal `EntityComponent` livello, puoi solo creare o sostituire un componente. L'entità deve già esistere, altrimenti l'azione produrrà un `ValidationException`.

A livello di proprietà o relazione, è possibile solo creare o sostituire una proprietà o una relazione e il contenitore `EntityComponent` deve già esistere.

- **Eliminazione della risorsa secondaria:**

AWS IoT TwinMaker supporta anche l'eliminazione delle risorse secondarie. Una sottorisorsa può essere un componente, una proprietà o una relazione.

Se desideri eliminare un componente, devi farlo a livello di entità.

Se desideri eliminare una proprietà o una relazione, devi farlo dal livello Entità o dal EntityComponent livello.

Per eliminare una risorsa secondaria, si aggiorna la risorsa di livello superiore e si omette la definizione della risorsa secondaria.

- Nessuna eliminazione delle risorse di primo livello: non AWS IoT TwinMaker eliminerà mai le risorse di primo livello. Una risorsa di primo livello si riferisce a un'entità o. ComponentType
- Nessuna definizione di risorsa secondaria per la stessa risorsa di primo livello in un modello:

Non è possibile fornire la definizione completa dell'entità e la definizione di sottorisorsa (ad esempio la proprietà) della stessa entità nello stesso modello.

Se un EntityID viene utilizzato in Entity, non è possibile utilizzare lo stesso ID in Entity EntityComponent, proprietà o relazione.

Se viene utilizzata una combinazione EntityID o componentName in EntityComponent, non è possibile utilizzare la stessa combinazione in EntityComponent, proprietà o relazione.

Se nella proprietà o nella relazione viene utilizzata una combinazione EntityID, componentName, PropertyName, non è possibile utilizzare la stessa combinazione nella proprietà o nella relazione.

- ExternalId è facoltativo per AWS IoT TwinMaker: ExternalId Può essere utilizzato per aiutarti a identificare le tue risorse.

## Esecuzione di operazioni di importazione ed esportazione in blocco

Questo argomento spiega come eseguire operazioni di importazione ed esportazione in blocco e come gestire gli errori nei processi di trasferimento. Fornisce esempi di processi di trasferimento che utilizzano i comandi CLI.

L' AWS IoT TwinMaker API Reference contiene informazioni sulle [CreateMetadataTransferJob](#) altre azioni API.

### Argomenti

- [metadataTransferJob prerequisiti](#)
- [Autorizzazioni IAM](#)
- [Esegui un'operazione in blocco](#)
- [Gestione degli errori](#)
- [Importa modelli di metadati](#)
- [AWS IoT TwinMaker metadataTransferJob esempi](#)

## metadataTransferJob prerequisiti

Completa i seguenti prerequisiti prima di eseguire un: metadataTransferJob

- Crea uno spazio di AWS IoT TwinMaker lavoro. L'area di lavoro può essere la destinazione di importazione o la fonte di esportazione per un. metadataTransferJob Per informazioni sulla creazione di un'area di lavoro, consulta, [. Creare un'area di lavoro](#)
- Crea un bucket Amazon S3 per archiviare le risorse. Per ulteriori informazioni sull'uso di Amazon S3, consulta [What is Amazon S3?](#)

## Autorizzazioni IAM

Quando esegui operazioni di massa, devi creare una policy IAM con autorizzazioni per consentire lo scambio di AWS risorse tra Amazon S3 AWS IoT TwinMaker, AWS IoT SiteWise, e il tuo computer locale. Per ulteriori informazioni sulla creazione di policy IAM, consulta [Creazione](#) di policy IAM.

Le dichiarazioni politiche per AWS IoT TwinMaker Amazon S3 AWS IoT SiteWise e Amazon sono elencate di seguito:

- AWS IoT TwinMaker politica:

```
{
  "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 politica:

```

{
  "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": "*"
  }
]
}

```

- **Politica di Amazon S3:**

```

{
  "Effect": "Allow",
  "Action": [
    "s3:PutObject",
    "s3:GetObject",
    "s3:GetBucketLocation",
    "s3:ListBucket",
    "s3:AbortMultipartUpload",
    "s3:ListBucketMultipartUploads",
    "s3:ListMultipartUploadParts"
  ],
  "Resource": "*"
}

```

```
}
```

In alternativa, puoi definire l'ambito della tua policy Amazon S3 per accedere a un solo bucket Amazon S3, consulta la seguente policy.

### Policy con ambito a bucket singolo di 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/*"
  ]
}
```

## Imposta il controllo degli accessi per un metadataTransferJob

Per controllare il tipo di lavoro a cui un utente può accedere, aggiungi la seguente politica IAM al ruolo utilizzato per la chiamata AWS IoT TwinMaker.

### Note

Questa politica consente l'accesso solo ai lavori di AWS IoT TwinMaker importazione ed esportazione che trasferiscono risorse da e verso Amazon S3.

```
{
  "Effect": "Allow",
  "Action": [
    "iottwinmaker:*DataTransferJob*"
  ],
}
```

```

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

```

## Esegui un'operazione in blocco

Questa sezione spiega come eseguire operazioni di importazione ed esportazione in blocco.

### Importa dati da Amazon S3 a AWS IoT TwinMaker

1. Specificate le risorse che desiderate trasferire utilizzando lo AWS IoT TwinMaker metadataTransferJob schema. Crea e archivia il tuo file di schema nel tuo bucket Amazon S3.

Per esempi di schemi, vedi. [Importa modelli di metadati](#)

2. Crea un corpo della richiesta e salvalo come file JSON. Il corpo della richiesta specifica l'origine e la destinazione del processo di trasferimento. Assicurati di specificare il tuo bucket Amazon S3 come origine e il tuo AWS IoT TwinMaker spazio di lavoro come destinazione.

Di seguito è riportato un esempio di corpo della richiesta:

```

{
  "metadataTransferJobId": "your-transfer-job-Id",
  "sources": [{
    "type": "s3",
    "s3Configuration": {
      "location": "arn:aws:s3:::your-S3-bucket-name/your_import_data.json"
    }
  }],
  "destination": {
    "type": "iottwinmaker",
    "iotTwinMakerConfiguration": {

```

```

        "workspace": "arn:aws:iottwinmaker:us-
east-1:111122223333:workspace/your-worksapce-name"
      }
    }
  }
}

```

Registra il nome del file che hai assegnato al corpo della richiesta, ne avrai bisogno nel passaggio successivo. In questo esempio il corpo della richiesta è denominato `createMetadataTransferJobImport.json`.

3. Esegui il seguente comando CLI da richiamare `CreateMetadataTransferJob` (sostituisci il nome del file `input-json` con il nome che hai assegnato al corpo della richiesta):

```

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

```

Questo crea un file `metadataTransferJob` e avvia il processo di trasferimento delle risorse selezionate.

## Esportazione di dati AWS IoT TwinMaker da Amazon S3

1. Crea un corpo di richiesta JSON con i filtri appropriati per scegliere le risorse che desideri esportare. Per questo esempio utilizziamo:

```

{
  "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:::your-S3-bucket-location"
  }
}
}
}

```

L'`filtersarray` consente di specificare quali risorse verranno esportate. In questo esempio filtriamo `perentity`, `ecomponentType`.

Assicurati di specificare il tuo AWS IoT TwinMaker spazio di lavoro come origine e il tuo bucket Amazon S3 come destinazione del processo di trasferimento dei metadati.

Salva il corpo della richiesta e registra il nome del file, ti servirà nel passaggio successivo. Per questo esempio, abbiamo chiamato il corpo della nostra richiesta `createMetadataTransferJobExport.json`.

2. Esegui il seguente comando CLI da richiamare `CreateMetadataTransferJob` (sostituisci il nome del file `input-json` con il nome che hai assegnato al corpo della richiesta):

```

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

```

Questo crea un file `metadataTransferJob` e avvia il processo di trasferimento delle risorse selezionate.

Per controllare o aggiornare lo stato di un processo di trasferimento, utilizzate i seguenti comandi:

- Per annullare un lavoro, utilizza l'azione [CancelMetadataTransferJob](#) API. Quando si effettua una chiamata `CancelMetadataTransferJob`, l'API annulla solo una chiamata in esecuzione `metadataTransferJob` e le risorse già esportate o importate non sono interessate da questa chiamata API.
- Per recuperare informazioni su un lavoro specifico, utilizza l' [GetMetadataTransferJob](#) azione API.

In alternativa, puoi richiamare `GetMetadataTransferJob` un processo di trasferimento esistente con il seguente comando CLI:

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

Se richiami `GetMetadataTransferJob` un processo di AWS IoT TwinMaker importazione o esportazione inesistente, ricevi un `ResourceNotFoundException` errore in risposta.

- Per elencare i lavori correnti, utilizza l'azione [ListMetadataTransferJobsAPI](#).

Ecco un esempio di CLI che chiama `ListMetadataTransferJobs` with AWS IoT TwinMaker come `destinationType` e come `SourceType`: `s3`

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

#### Note

È possibile modificare i valori dei parametri `SourceType` e `DestinationType` in modo che corrispondano all'origine e alla destinazione del processo di importazione o esportazione.

Per altri esempi di comandi CLI che richiamano queste azioni API, consulta. [AWS IoT TwinMaker metadataTransferJob esempi](#)

Se si verificano errori durante il processo di trasferimento, consulta. [Gestione degli errori](#)

## Gestione degli errori

Dopo aver creato ed eseguito un processo di trasferimento, puoi chiamare `GetMetadataTransferJob` per diagnosticare eventuali errori che si sono verificati:

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

Una volta visualizzato lo stato del lavoro a cui rivolgerli `COMPLETED`, puoi verificarne i risultati.

`GetMetadataTransferJob` restituisce un oggetto chiamato [MetadataTransferJobProgress](#) che contiene i seguenti campi:

- **FailedCount:** indica il numero di risorse che hanno avuto esito negativo durante il processo di trasferimento.
- **SkippedCount:** indica il numero di risorse che sono state ignorate durante il processo di trasferimento.
- **suceededCount:** indica il numero di risorse che hanno avuto successo durante il processo di trasferimento.
- **totalCount:** indica il numero totale di risorse coinvolte nel processo di trasferimento.

Inoltre, viene restituito un elemento `reportURL` che contiene un URL prefirmato. Se il processo di trasferimento presenta errori che desideri approfondire, puoi scaricare un rapporto di errore completo utilizzando questo URL.

## Importa modelli di metadati

È possibile importare molti componenti, `ComponentType` o entità con un'unica operazione di importazione in blocco. Gli esempi in questa sezione mostrano come eseguire questa operazione.

template: Importing entities

Utilizzate il seguente formato di modello per un lavoro che importa entità:

```
{
  "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

Utilizzate il seguente formato di modello per un job che 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

Utilizzate il seguente formato di modello per un lavoro che importa componenti:

```
{
  "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 esempi

Utilizza i seguenti comandi per gestire i trasferimenti di metadati:

- [CreateMetadataTransferJob](#)Azione API.

### Esempio di comando CLI:

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

- Per annullare un lavoro, utilizza l'azione [CancelMetadataTransferJob](#) API.

### Esempio di comando CLI:

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

Quando si chiama `CancelMetadataTransferJob`, annulla solo un processo di trasferimento di metadati specifico e le risorse già esportate o importate non vengono influenzate.

- Per recuperare informazioni su un lavoro specifico, utilizza l'azione API. [GetMetadataTransferJob](#)

### Esempio di comando CLI:

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

```

- Per elencare i lavori correnti, usa l'azione [ListMetadataTransferJobs](#) API.

Puoi filtrare i risultati restituiti `ListMetadataTransferJobs` utilizzando un file JSON. Consultate la seguente procedura utilizzando la CLI:

1. Crea un file JSON di input CLI per specificare i filtri che desideri utilizzare:

```
{  
  "sourceType": "s3",  
  "destinationType": "iottwinmaker",  
  "filters": [{  
    "workspaceId": "workspaceforbulkimport"  
  }],  
  {  
    "state": "COMPLETED"  
  }  
}]
```

Salvalo e registra il nome del file, ti servirà quando inserisci il comando CLI.

2. Utilizzate il file JSON come argomento per il seguente comando CLI:

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

## AWS IoT TwinMaker schema del processo di trasferimento dei metadati

metadataTransferJob schema di importazione: utilizza questo schema di AWS IoT TwinMaker metadati per convalidare i dati quando li carichi su un bucket Amazon S3:

```
{  
  "$schema": "https://json-schema.org/draft/2020-12/schema",  
  "title": "IoTTwinMaker",  
  "description": "Metadata transfer job resource schema for IoTTwinMaker",  
  "definitions": {  
    "ExternalId": {  
      "type": "string",  
      "minLength": 1,  
      "maxLength": 128,  
      "pattern": "[a-zA-Z0-9][a-zA-Z_\\-0-9.:]*[a-zA-Z0-9]+"  
    },  
    "Description": {  
      "type": "string",  
      "minLength": 0,  
      "maxLength": 512  
    },  
    "DescriptionWithDefault": {  
      "type": "string",  
      "minLength": 0,  
      "maxLength": 512,  
      "default": ""  
    },  
    "ComponentTypeName": {  
      "description": "A friendly name for the component type.",  
      "type": "string",  
      "pattern": ".*[^\\u0000-\\u001F\\u007F]*.*",  
      "minLength": 1,  
    },  
  },  
}
```

```

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

```

```

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

```

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



```

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

```

```
    },
    "nestedType": {
      "description": "The nested type in the data type.",
      "$ref": "#/definitions/DataType"
    },
    },
    "relationship": {
      "description": "A relationship that associates a component with another
component.",
      "$ref": "#/definitions/Relationship"
    },
    },
    "type": {
      "description": "The underlying type of the data type.",
      "type": "string",
      "enum": [
        "RELATIONSHIP",
        "STRING",
        "LONG",
        "BOOLEAN",
        "INTEGER",
        "DOUBLE",
        "LIST",
        "MAP"
      ]
    },
    },
    "unitOfMeasure": {
      "description": "The unit of measure used in this data type.",
      "type": "string",
      "pattern": ".*",
      "minLength": 1,
      "maxLength": 256
    }
  },
  "required": [
    "type"
  ],
  "additionalProperties": false
},
"PropertyDefinition": {
  "description": "An object that specifies information about a property.",
  "type": "object",
  "properties": {
    "configuration": {
      "description": "An object that specifies information about a property.",
      "patternProperties": {
```

```

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

```

```

    }
  },
  "additionalProperties": false
},
"PropertyDefinitions": {
  "type": "object",
  "patternProperties": {
    "[a-zA-Z_\\-0-9]+": {
      "$ref": "#/definitions/PropertyDefinition"
    }
  },
  "additionalProperties": {
    "$ref": "#/definitions/PropertyDefinition"
  }
},
"Property": {
  "type": "object",
  "properties": {
    "definition": {
      "description": "The definition of the property",
      "$ref": "#/definitions/PropertyDefinition"
    },
    "value": {
      "description": "The value of the property.",
      "$ref": "#/definitions/DataValue"
    }
  },
  "additionalProperties": false
},
"Properties": {
  "type": "object",
  "patternProperties": {
    "[a-zA-Z_\\-0-9]+": {
      "$ref": "#/definitions/Property"
    }
  },
  "additionalProperties": {
    "$ref": "#/definitions/Property"
  }
},
"PropertyName": {
  "type": "string",
  "pattern": "[a-zA-Z_\\-0-9]+"
},

```

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

```

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

```

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

```

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

```



```

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

```

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

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

Ecco un esempio che crea un nuovo `ComponentType` chiamato `component.type.initial` e crea un'entità chiamata: `initial`

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

Ecco un esempio che aggiorna le entità esistenti:

```
{
  "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 connettori dati

AWS IoT TwinMaker utilizza un'architettura basata su connettori in modo da poter connettere i dati dal proprio data store a. AWS IoT TwinMaker Ciò significa che non è necessario migrare i dati prima dell'uso. AWS IoT TwinMaker Attualmente, AWS IoT TwinMaker supporta connettori proprietari per. AWS IoT SiteWise Se memorizzi i dati di modellazione e proprietà in AWS IoT SiteWise, non è necessario implementare connettori personalizzati. Se memorizzi i dati di modellazione o proprietà in altri archivi dati, come Timestream, DynamoDB o Snowflake, devi implementare AWS Lambda connettori con l'interfaccia del connettore AWS IoT TwinMaker dati in modo che possa richiamare il connettore quando necessario. AWS IoT TwinMaker

## Argomenti

- [AWS IoT TwinMaker connettori dati](#)
- [AWS IoT TwinMaker Connettore dati tabulare Athena](#)
- [Sviluppo di connettori per dati in serie temporali AWS IoT TwinMaker](#)

# AWS IoT TwinMaker connettori dati

I connettori devono accedere al data store sottostante per risolvere le interrogazioni inviate e restituire risultati o errori.

Per informazioni sui connettori disponibili, le relative interfacce di richiesta e le relative interfacce di risposta, consulta i seguenti argomenti.

Per ulteriori informazioni sulle proprietà utilizzate nelle interfacce dei connettori, consulta l'[GetPropertyValueHistory](#)azione API.

### Note

Alcuni connettori dispongono di due campi timestamp nelle interfacce di richiesta e risposta per le proprietà dell'ora di inizio e dell'ora di fine. Entrambi `startDateTime` `endDateTime` utilizzano un numero lungo per rappresentare epoch second, che non è più supportato. Per mantenere la compatibilità con le versioni precedenti, inviamo comunque un valore di timestamp a quel campo, ma consigliamo di utilizzare `endTime` campi `startTime` e coerenti con il formato di timestamp della nostra API.

## Argomenti

- [Connettore di inizializzazione dello schema](#)
- [DataReaderByEntity](#)
- [DataReaderByComponentType](#)
- [DataReader](#)
- [AttributePropertyValueReaderByEntity](#)
- [DataWriter](#)
- [Esempi](#)

## Connettore di inizializzazione dello schema

È possibile utilizzare l'inizializzatore dello schema nel ciclo di vita del tipo di componente o dell'entità per recuperare il tipo di componente o le proprietà del componente dall'origine dati sottostante. L'inizializzatore dello schema importa automaticamente il tipo di componente o le proprietà del componente senza richiamare esplicitamente un'azione API per la configurazione. `properties`

### SchemaInitializer interfaccia di richiesta

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

#### Note

La mappa delle proprietà in questa interfaccia di richiesta è una `PropertyRequest`. Per ulteriori informazioni, vedere [PropertyRequest](#).

## SchemaInitializer interfaccia di risposta

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

### Note

La mappa delle proprietà in questa interfaccia di richiesta è una `PropertyResponse`. Per ulteriori informazioni, vedere [PropertyResponse](#).

## DataReaderByEntity

`DataReaderByEntity` è un connettore del piano dati utilizzato per ottenere i valori delle serie temporali delle proprietà in un singolo componente.

Per informazioni sui tipi di proprietà, sulla sintassi e sul formato di questo connettore, vedi l'azione [GetPropertyValueHistoryAPI](#).

## DataReaderByEntityinterfaccia di richiesta

```
{
  "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"
}

```

## DataReaderByEntityinterfaccia di risposta

```

{
  "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

Per ottenere i valori delle serie temporali delle proprietà comuni che provengono dallo stesso tipo di componente, utilizzate il connettore `DataReaderByEntity` del piano dati. Ad esempio, se definisci proprietà di serie temporali nel tipo di componente e hai più componenti che utilizzano quel tipo di componente, puoi interrogare tali proprietà su tutti i componenti in un determinato intervallo di tempo. Un caso d'uso comune è quando si desidera interrogare lo stato di allarme di più componenti per una visione globale delle entità.

Per informazioni sui tipi di proprietà, sulla sintassi e sul formato di questo connettore, vedi l'azione [GetPropertyValueHistory](#) API.

## DataReaderByComponentType interfaccia di richiesta

```
{
  "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 interfaccia di risposta

```
{
  "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 è un connettore data plane in grado di gestire sia le maiuscole che le minuscole DataReaderByEntity e le minuscole DataReaderByComponentType.

Per informazioni sui tipi di proprietà, sulla sintassi e sul formato di questo connettore, consulta l'azione [GetPropertyValueHistoryAPI](#).

### DataReader interfaccia di richiesta

EntityId e componentName sono facoltativi.

```
{
  "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 interfaccia di risposta

```
{
  "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 è un connettore del piano dati che è possibile utilizzare per recuperare il valore delle proprietà statiche in una singola entità.

Per informazioni sui tipi di proprietà, sulla sintassi e sul formato di questo connettore, vedi l'azione [GetPropertyValueAPI](#).

### AttributePropertyValueReaderByEntity interfaccia di richiesta

```

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

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

```

### AttributePropertyValueReaderByEntity interfaccia di risposta

```

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

```

```
}
}
```

## DataWriter

DataWriter è un connettore del piano dati che è possibile utilizzare per riscrivere i punti dati di serie temporali nell'archivio dati sottostante per le proprietà di un singolo componente.

Per informazioni sui tipi di proprietà, sulla sintassi e sul formato di questo connettore, consulta l'azione [BatchPutPropertyValuesAPI](#).

### DataWriter interfaccia di richiesta

```
{
  "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 interfaccia di risposta

```
{
```

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

## Esempi

I seguenti esempi JSON sono esempi di sintassi di risposta e richiesta per più connettori.

- SchemaInitializer:

Gli esempi seguenti mostrano l'iniziatore dello schema nel ciclo di vita di un tipo di componente.

Richiesta:

```
{  
  "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"
    }
}
}
}
}

```

Risposta:

```

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

```

```
}  
}
```

- Iniziatore dello schema nel ciclo di vita dell'entità:

Richiesta:

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



```
}
```

Risposta:

```
{
  "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:**

Richiesta:

```
{
  "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
  }
}
}

```

Risposta:

```

{
  "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:

Richiesta:

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

```
}
```

Risposta:

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

• **DataWriter:**

Richiesta:

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

Risposta:

```
{
  "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 Connettore dati tabulare Athena

Con il connettore dati tabulare Athena, puoi accedere e utilizzare i tuoi archivi dati Athena in AWS IoT TwinMaker. Puoi usare i dati di Athena per creare gemelli digitali senza un intenso sforzo di migrazione dei dati. Puoi utilizzare il connettore predefinito o creare un connettore Athena personalizzato per accedere ai dati dalle tue fonti di dati Athena.

## AWS IoT TwinMaker Prerequisiti del connettore dati Athena

Prima di utilizzare il connettore dati tabulare Athena, completa i seguenti prerequisiti:

- Crea tabelle Athena gestite e le relative risorse Amazon S3 associate. [Per informazioni sull'uso di Athena, consulta la documentazione di Athena.](#)
- Crea uno spazio di lavoro AWS IoT TwinMaker. [È possibile creare un'area di lavoro nella AWS IoT TwinMaker console.](#)
- Aggiorna il ruolo IAM del tuo spazio di lavoro con le autorizzazioni Athena. Per ulteriori informazioni, consulta [Modifica il ruolo IAM del tuo spazio di lavoro per utilizzare il connettore dati Athena.](#)
- Acquisisci familiarità con AWS IoT TwinMaker il sistema entità-componente e con come creare entità. Per ulteriori informazioni, consulta [Crea la tua prima entità.](#)

- Acquisisci familiarità con i AWS IoT TwinMaker connettori dati. Per ulteriori informazioni, consulta [AWS IoT TwinMaker connettori dati](#).

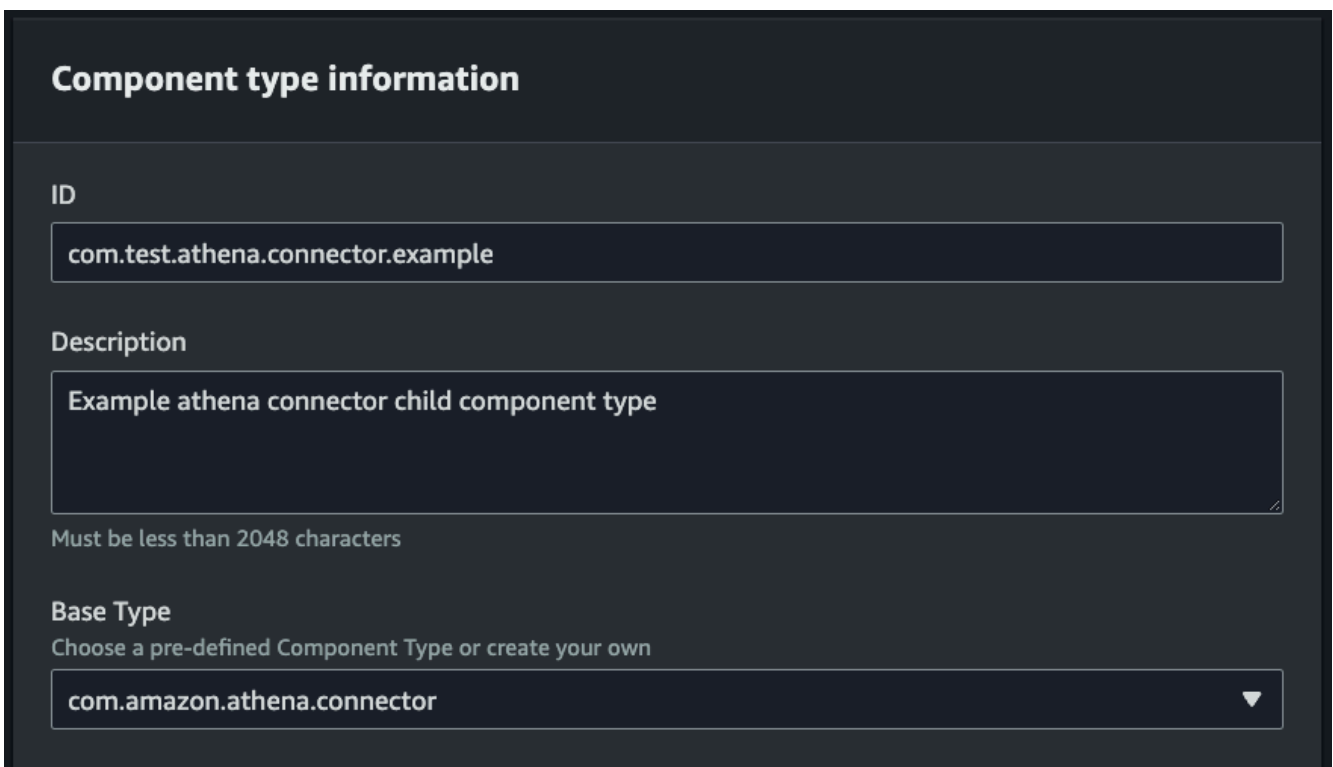
## Utilizzo del connettore dati Athena

Per utilizzare il connettore dati Athena, è necessario creare un componente, utilizzando il connettore Athena come tipo di componente. Quindi colleghi il componente a un'entità all'interno della scena per utilizzarlo in. AWS IoT TwinMaker

### Creare un tipo di componente con il connettore dati Athena

Usa questa procedura per creare un tipo di AWS IoT TwinMaker componente con il connettore dati tabulare Athena:

1. Passare alla [console AWS IoT TwinMaker](#).
2. Apri uno spazio di lavoro esistente o [creane](#) uno nuovo.
3. Dal menu di navigazione a sinistra, scegli Tipi di componenti e seleziona Crea tipo di componente per aprire la pagina di creazione del tipo di componente.
4. Nella pagina Crea tipo di componente, compila il campo ID con un ID che corrisponda al tuo caso d'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. Scegli il tipo di base. Dall'elenco a discesa, seleziona il connettore dati tabulare Athena etichettato come `com.amazon.athena.connector`.
6. Configura l'origine dati del tipo di componente scegliendo le risorse Athena per i seguenti campi:
  - Scegli un'origine dati Athena.
  - Scegliete un database Athena.
  - Scegli un nome per la tabella.
  - Scegli un gruppo di lavoro Athena.
7. Dopo aver selezionato le risorse Athena che desideri utilizzare come origine dati, scegli quali colonne della tabella desideri includere.
8. Seleziona il nome di una colonna con ID esterno. Seleziona una colonna del passaggio precedente da utilizzare come colonna ID esterno. L'ID esterno è l'ID utilizzato per rappresentare una risorsa Athena e mapparla a un' AWS IoT TwinMaker entità.

## 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. (Facoltativo) Aggiungi AWS tag a queste risorse per raggrupparle e organizzarle.
10. Scegliete Crea tipo di componente per completare la creazione del tipo di componente.

Crea un componente con il tipo di connettore dati Athena e collegalo a un'entità

Usa questa procedura per creare un AWS IoT TwinMaker componente con il connettore dati tabulare Athena e collegarlo a un'entità:

 Note

È necessario disporre di un tipo di componente esistente che utilizzi il connettore dati tabulare Athena come origine dati per completare questa procedura. Vedi la procedura precedente [Creare un tipo di componente con il connettore dati Athena](#) prima di iniziare questa procedura dettagliata.

1. Passare alla [console AWS IoT TwinMaker](#).
2. Apri uno spazio di lavoro esistente o [creane](#) uno nuovo.
3. Dal menu di navigazione a sinistra scegli Entità e seleziona l'entità a cui desideri aggiungere il componente o crearne una nuova.
4. [Crea una nuova entità](#).
5. Quindi seleziona Aggiungi componente. , compila il campo Nome componente con un nome che corrisponda al tuo caso d'uso.
6. Dal menu a discesa Tipo di componente, seleziona l'ID del tipo di componente creato nella procedura precedente.
7. Immettete le informazioni sul componente, il nome del componente e selezionate il figlio ComponentType creato in precedenza. Questo è quello ComponentType che hai creato con il connettore dati Athena.
8. Nella sezione Proprietà, inserisci l'athenaComponentExternalID del componente.

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

Add another property

- Scegli Aggiungi componente per completare la creazione del componente.

Ora hai creato con successo un componente con il connettore dati Athena come tipo di componente e lo hai collegato a un'entità.

## Utilizzo del riferimento JSON del connettore dati tabulare Athena

L'esempio seguente è il riferimento JSON completo per il connettore dati tabulare Athena. Utilizzalo come risorsa per creare connettori dati e tipi di componenti personalizzati.

```
{
  "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
      }
    }
  }
}

```

## Utilizzo del connettore dati Athena

È possibile far emergere le entità che utilizzano le tabelle Athena in Grafana. Per ulteriori informazioni, consulta Integrazione della [dashboard AWS IoT TwinMaker Grafana](#).

Leggi la [documentazione di Athena](#) per informazioni sulla creazione e l'utilizzo delle tabelle Athena per archiviare i dati.

## Risoluzione dei problemi del connettore dati Athena

In questo argomento vengono descritti i problemi più comuni che possono verificarsi durante la configurazione del connettore dati Athena.

Sede del gruppo di lavoro Athena:

Quando si crea il ComponentType del connettore Athena, un gruppo di lavoro Athena deve disporre della posizione di uscita configurata. [Vedi Come funzionano i gruppi di lavoro](#).

## Autorizzazioni per i ruoli IAM mancanti:

Al ruolo AWS IoT TwinMaker; workspace potrebbe mancare l'autorizzazione di accesso all'API Athena durante la creazione di un ComponentType, l'aggiunta di un componente Ca a un'entità o l'esecuzione dell'API. GetPropertyValue Per aggiornare le autorizzazioni IAM, consulta [Creare e gestire un ruolo di servizio per. AWS IoT TwinMaker](#)

## Visualizza i dati tabulari di Athena in Grafana

È disponibile anche un plug-in Grafana per visualizzare i dati tabulari su Grafana, un pannello di dashboard con funzionalità aggiuntive come l'ordinamento e il filtraggio in base a proprietà selezionate senza effettuare chiamate API o interazioni con Athena. AWS IoT TwinMaker Questo argomento mostra come configurare Grafana per visualizzare i dati tabulari di Athena.

### Prerequisiti

Prima di configurare un pannello Grafana per la visualizzazione dei dati tabulari Athena, verifica i seguenti prerequisiti:

- Hai creato un ambiente Grafana. Per ulteriori informazioni, vedere Integrazione [AWS IoT TwinMaker Grafana](#).
- Puoi configurare un'origine dati Grafana. Per ulteriori informazioni, vedere [Grafana AWS IoT TwinMaker](#).
- Conosci bene la creazione di una nuova dashboard e l'aggiunta di un nuovo pannello.

## Visualizza i dati tabulari di Athena in Grafana

Questa procedura mostra come configurare un pannello Grafana per visualizzare i dati tabulari di Athena.

1. Apri la dashboard AWS IoT TwinMaker Grafana.
2. Seleziona il pannello Tabella nelle impostazioni del pannello.
3. Seleziona la tua fonte di dati nella configurazione della query.
4. Seleziona la query Get Property Value.
5. Selezionare un'entità.

6. Seleziona un componente con un ComponentType che estende il tipo di componente base Athena.
7. Seleziona il gruppo di proprietà della tua tabella Athena.
8. Seleziona un numero qualsiasi di proprietà dal gruppo di proprietà.
9. Configura le condizioni tabulari tramite un elenco di filtri e ordini di proprietà. Con le seguenti opzioni:
  - Filtro: definisci un'espressione per il valore di una proprietà per filtrare i dati.
  - OrderBy: specifica se i dati devono essere restituiti in ordine crescente o decrescente per una proprietà.

Panel Title					
crit {componentName="	description {component	equipment_type {compo	status {componentNam	total {componentName=	won {componentName=
5	Shutdown valve inspec...	VALVE	COMPLETED	90563	128355
5	Damaged cable on SDV	VALVE	COMPLETED	90041	128461
5	BYTN-04-TV-02385 do...	VALVE	COMPLETED	85611	128361
5	Shutdown vlv inspection	VALVE	COMPLETED	73797	128531
5	RYTN-02-XV-06517 do	VALVE	COMPLETED	71326	128458

**Query** 1 Transform 0

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

# Sviluppo di connettori per dati in serie temporali AWS IoT TwinMaker

Questa sezione spiega come sviluppare un connettore dati per serie temporali in un step-by-step processo. Inoltre, presentiamo un esempio di connettore dati per serie temporali basato sull'intero esempio di Cookie Factory, che include modelli 3D, entità, componenti, allarmi e connettori. La fonte di esempio di cookie factory è disponibile nell'archivio degli [AWS IoT TwinMaker esempi GitHub](#).

## Argomenti

- [AWS IoT TwinMaker prerequisiti del connettore dati per serie temporali](#)
- [Sfondo del connettore dati delle serie temporali](#)
- [Sviluppo di un connettore dati per serie temporali](#)
- [Miglioramento del connettore dati](#)
- [Test del connettore](#)
- [Sicurezza](#)
- [Creazione di risorse AWS IoT TwinMaker](#)
- [Cosa c'è dopo](#)
- [AWS IoT TwinMaker esempio di connettore per serie temporali di cookie factory](#)

## AWS IoT TwinMaker prerequisiti del connettore dati per serie temporali

Prima di sviluppare il connettore dati per serie temporali, ti consigliamo di completare le seguenti attività:

- Crea uno spazio di [AWS IoT TwinMaker lavoro](#).
- Crea [tipi di AWS IoT TwinMaker componenti](#).
- Creare [AWS IoT TwinMaker entità](#).
- (Facoltativo) Leggi [Utilizzo e creazione dei tipi di componenti](#).
- (Facoltativo) Leggi l'[interfaccia del connettore AWS IoT TwinMaker dati](#) per ottenere una comprensione generale dei connettori AWS IoT TwinMaker dati.



## Note

Per un esempio di connettore completamente implementato, consulta il nostro esempio di implementazione di cookie factory.

## Sfondo del connettore dati delle serie temporali

Immagina di lavorare in una fabbrica che ha un set di frullatori per biscotti e un serbatoio d'acqua. Vorresti creare gemelli AWS IoT TwinMaker digitali di queste entità fisiche in modo da poterne monitorare gli stati operativi controllando varie metriche di serie temporali.

Hai configurato i sensori in loco e stai già trasmettendo i dati di misurazione in un database Timestream. Volete essere in grado di visualizzare e organizzare i dati di misurazione AWS IoT TwinMaker con un sovraccarico minimo. È possibile eseguire questa operazione utilizzando un connettore dati per serie temporali. L'immagine seguente mostra un esempio di tabella di telemetria, che viene popolata mediante l'uso di un connettore di serie temporali.

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	-	59.9421195988887
Mixer_24_7ff0b75b-f0fa-43f0-bc89-b96337586d00	Mixer	Temperature	2022-04-19 00:28:00.241000000	-	99.1292877197266
Mixer_25_cf42effc-ba19-48ba-bbc3-d21d2508ce31	Mixer	RPM	2022-04-19 00:28:00.241000000	-	59.8453979492188
Mixer_20_0568f25f-116c-429c-a974-5ceec065a6ac	Mixer	Temperature	2022-04-19 00:28:00.241000000	-	99.1292877197266
Mixer_24_7ff0b75b-f0fa-43f0-bc89-b96337586d00	Mixer	RPM	2022-04-19 00:28:00.241000000	-	60.4532585144043
Mixer_15_0bb566cd-d6f3-4804-9fe1-7d2abca82d00	Mixer	RPM	2022-04-19 00:28:00.241000000	-	58.397144317627
Mixer_2_d8e76844-e739-4845-a748-a83983279376	Mixer	RPM	2022-04-19 00:28:00.241000000	-	60.206958770752
Mixer_6_b66db3d3-c144-47b5-afb9-3a0150c53456	Mixer	RPM	2022-04-19 00:28:00.241000000	-	60.206958770752

[I set di dati e la tabella Timestream utilizzati in questa schermata sono disponibili nell'archivio degli esempi.](#) [AWS IoT TwinMaker GitHub](#) Vedi anche il [connettore di esempio cookie factory](#) per l'implementazione, che produce il risultato mostrato nella schermata precedente.

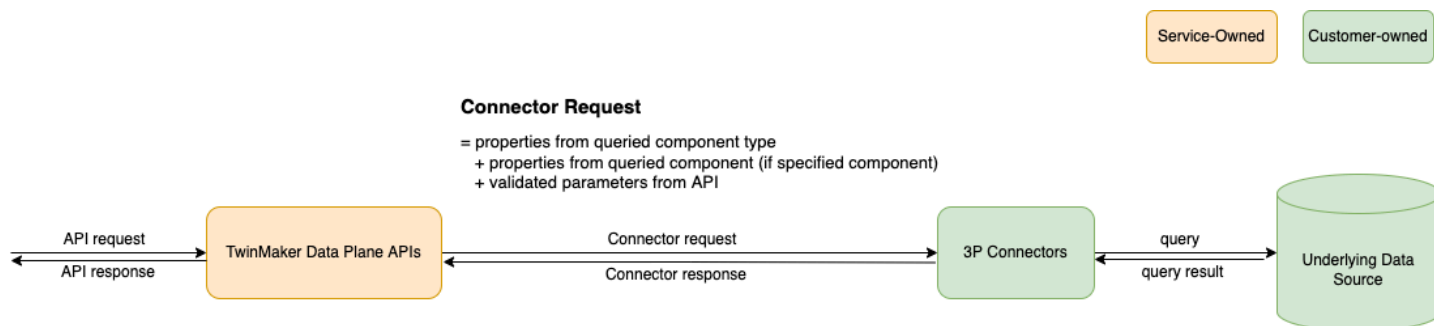
## Flusso di dati del connettore di serie temporali

Per le interrogazioni sul piano dati, AWS IoT TwinMaker recupera le proprietà corrispondenti dei componenti e dei tipi di componenti dalle definizioni dei componenti e dei tipi di componenti. AWS IoT TwinMaker inoltra le proprietà alle AWS Lambda funzioni insieme a tutti i parametri di query API nella query.

AWS IoT TwinMaker utilizza le funzioni Lambda per accedere e risolvere le interrogazioni provenienti da fonti di dati e restituire i risultati di tali interrogazioni. Le funzioni Lambda utilizzano le proprietà del componente e del tipo di componente dal piano dati per risolvere la richiesta iniziale.

I risultati della query Lambda vengono mappati su una risposta API e restituiti all'utente.

AWS IoT TwinMaker definisce l'interfaccia del connettore dati e la utilizza per interagire con le funzioni Lambda. Utilizzando i connettori dati, puoi interrogare la tua fonte di dati dall' AWS IoT TwinMaker API senza alcuna operazione di migrazione dei dati. L'immagine seguente illustra il flusso di dati di base descritto nei paragrafi precedenti.



## Sviluppo di un connettore dati per serie temporali

La procedura seguente delinea un modello di sviluppo che si sviluppa in modo incrementale fino a diventare un connettore dati funzionale per serie temporali. I passaggi di base sono i seguenti:

### 1. Crea un tipo di componente di base valido

In un tipo di componente, si definiscono proprietà comuni condivise tra i componenti. Per ulteriori informazioni sulla definizione dei tipi di componenti, consulta [Utilizzo e creazione di tipi di componenti](#).

AWS IoT TwinMaker utilizza un [modello di modellazione entità-componente](#) in modo che ogni componente sia collegato a un'entità. Si consiglia di modellare ogni elemento fisico come un'entità e di modellare diverse fonti di dati con i propri tipi di componenti.

L'esempio seguente mostra un tipo di componente del modello Timestream con una proprietà:

```

{"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
        }
    }
}
```

Gli elementi chiave del tipo di componente sono i seguenti:

- La `telemetryId` proprietà identifica la chiave univoca dell'elemento fisico nella fonte di dati corrispondente. Il connettore dati utilizza questa proprietà come condizione di filtro per interrogare solo i valori associati all'elemento specificato. Inoltre, se includi il valore della `telemetryId` proprietà nella risposta dell'API del piano dati, il lato client acquisisce l'ID e può eseguire una ricerca inversa, se necessario.
- Il `lambdaArn` campo identifica la funzione Lambda con cui interagisce il tipo di componente.
- Il `isRequiredInEntity` flag impone la creazione dell'ID. Questo flag è necessario in modo che quando viene creato il componente, venga istanziato anche l'ID dell'elemento.

- `TelemetryId`Viene aggiunto al tipo di componente come ID esterno in modo che l'elemento possa essere identificato nella tabella Timestream.

## 2. Crea un componente con il tipo di componente

Per utilizzare il tipo di componente creato, è necessario creare un componente e collegarlo all'entità da cui si desidera recuperare i dati. I passaggi seguenti descrivono in dettaglio il processo di creazione di quel componente:

- Passare alla [console AWS IoT TwinMaker](#).
- Selezionate e aprite lo stesso spazio di lavoro in cui avete creato i tipi di componenti.
- Vai alla pagina dell'entità.
- Crea una nuova entità o seleziona un'entità esistente dalla tabella.
- Dopo aver selezionato l'entità che desideri utilizzare, scegli Aggiungi componente per aprire la pagina Aggiungi componente.
- Assegna un nome al componente e, per il Tipo, scegli il tipo di componente che hai creato con il modello in 1. Crea un tipo di componente di base valido.

## 3. Fai in modo che il tipo di componente chiami un connettore Lambda

Il connettore Lambda deve accedere all'origine dati e generare l'istruzione di query in base all'input e inoltrarla all'origine dati. L'esempio seguente mostra un modello di richiesta JSON che esegue questa operazione.

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

Gli elementi chiave della richiesta:

- `selectedProperties` È un elenco che compili con le proprietà per le quali desideri le misurazioni Timestream.

- I endTime, startTime, startDateTime, startTimeEndDateTime, e specificano un intervallo di tempo per la richiesta. Ciò determina l'intervallo di campionamento per le misurazioni restituite.
- entityId è il nome dell'entità da cui si stanno interrogando i dati.
- componentName è il nome del componente da cui si stanno interrogando i dati.
- Utilizzate il orderByTime campo per organizzare l'ordine in cui vengono visualizzati i risultati.

Nella richiesta di esempio precedente, ci aspetteremmo di ottenere una serie di esempi per le proprietà selezionate durante la finestra temporale specificata per il dato elemento, con l'ordine temporale selezionato. La dichiarazione di risposta può essere riassunta come segue:

```
{
  "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. Aggiorna il tipo di componente per avere due proprietà

Il seguente modello JSON mostra un tipo di componente valido con due proprietà:

```
{  
  "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": false,  
      "isRequiredInEntity": true  
    }  
  }  
}
```

```

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

```

## 5. Aggiorna il connettore Lambda per gestire la seconda proprietà

L'API del piano AWS IoT TwinMaker dati supporta l'interrogazione di più proprietà in un'unica richiesta e AWS IoT TwinMaker segue una singola richiesta a un connettore fornendo un elenco di `selectedProperties`

La seguente richiesta JSON mostra un modello modificato che ora supporta una richiesta per due proprietà.

```

{
  "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
    }
}
}
```

Analogamente, viene aggiornata anche la risposta corrispondente, come illustrato nell'esempio seguente:

```
{
  "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

In termini di impaginazione, in questo caso, la dimensione della pagina nella richiesta si applica a tutte le proprietà. Ciò significa che con cinque proprietà nella query e una dimensione di pagina pari a 100, se ci sono abbastanza punti dati nell'origine, dovresti aspettarti di vedere 100 punti dati per proprietà, con 500 punti dati in totale.

Per un esempio di implementazione, consulta [l'esempio del connettore Snowflake](#) su GitHub

## Miglioramento del connettore dati

### Gestione delle eccezioni

È sicuro che il connettore Lambda generi eccezioni. Nella chiamata all'API del piano dati, il AWS IoT TwinMaker servizio attende che la funzione Lambda restituisca una risposta. Se l'implementazione del connettore genera un'eccezione, AWS IoT TwinMaker traduce il tipo di eccezione in `2ConnectorFailure`, rendendo il client API consapevole che si è verificato un problema all'interno del connettore.

## Gestione dell'impaginazione

Nell'esempio, Timestream fornisce una [funzione di utilità](#) che può aiutare a supportare l'impaginazione in modo nativo. Tuttavia, per alcune altre interfacce di interrogazione, come SQL, potrebbe essere necessario uno sforzo supplementare per implementare un algoritmo di impaginazione efficiente. Esiste un esempio di connettore [Snowflake](#) che gestisce l'impaginazione in un'interfaccia SQL.

Quando il nuovo token viene restituito AWS IoT TwinMaker tramite l'interfaccia di risposta del connettore, il token viene crittografato prima di essere restituito al client API. Quando il token è incluso in un'altra richiesta, lo AWS IoT TwinMaker decripta prima di inoltrarlo al connettore Lambda. Ti consigliamo di evitare di aggiungere informazioni sensibili al token.

## Test del connettore

Sebbene sia ancora possibile aggiornare l'implementazione dopo aver collegato il connettore al tipo di componente, consigliamo vivamente di verificare il connettore Lambda prima dell'integrazione con AWS IoT TwinMaker

Esistono diversi modi per testare il connettore Lambda: puoi testare il connettore Lambda nella console Lambda o localmente in AWS CDK

[Per ulteriori informazioni sul test delle funzioni Lambda, consulta Test delle funzioni Lambda e Test delle applicazioni Lambda a livello locale. AWS CDK](#)

## Sicurezza

[Per la documentazione sulle migliori pratiche di sicurezza con Timestream, consulta Security in Timestream.](#)

Per un esempio di prevenzione dell'iniezione SQL, vedere il seguente [script Python](#) in AWS IoT TwinMaker Samples GitHub Repository.

## Creazione di risorse AWS IoT TwinMaker

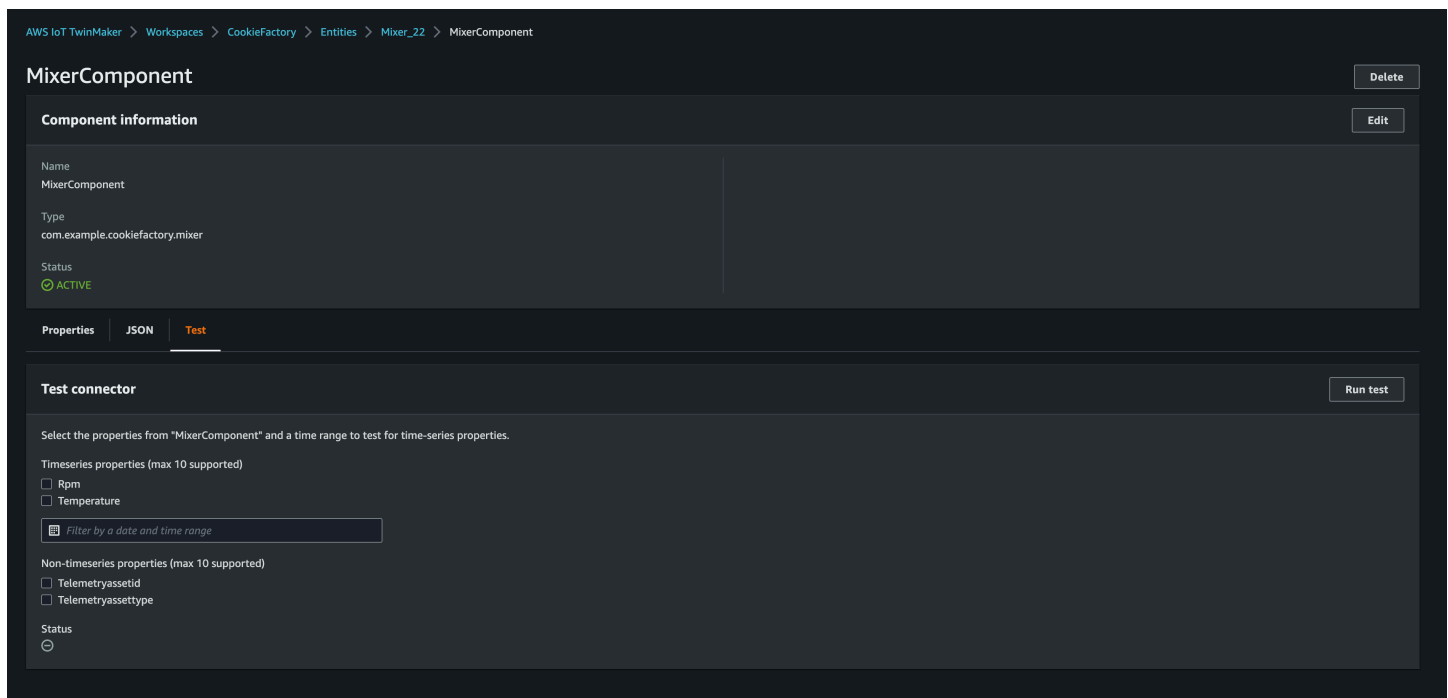
Dopo aver implementato la funzione Lambda, puoi creare AWS IoT TwinMaker risorse come tipi di componenti, entità e componenti tramite la [AWS IoT TwinMaker console o l'API](#).

## Note

Se segui le istruzioni di configurazione nell' [GitHub esempio](#), tutte le AWS IoT TwinMaker risorse sono disponibili automaticamente. È possibile controllare le definizioni dei tipi di componenti nell'[AWS IoT TwinMaker GitHub esempio](#). Una volta che il tipo di componente viene utilizzato da qualsiasi componente, le definizioni delle proprietà e le funzioni del tipo di componente non possono essere aggiornate.

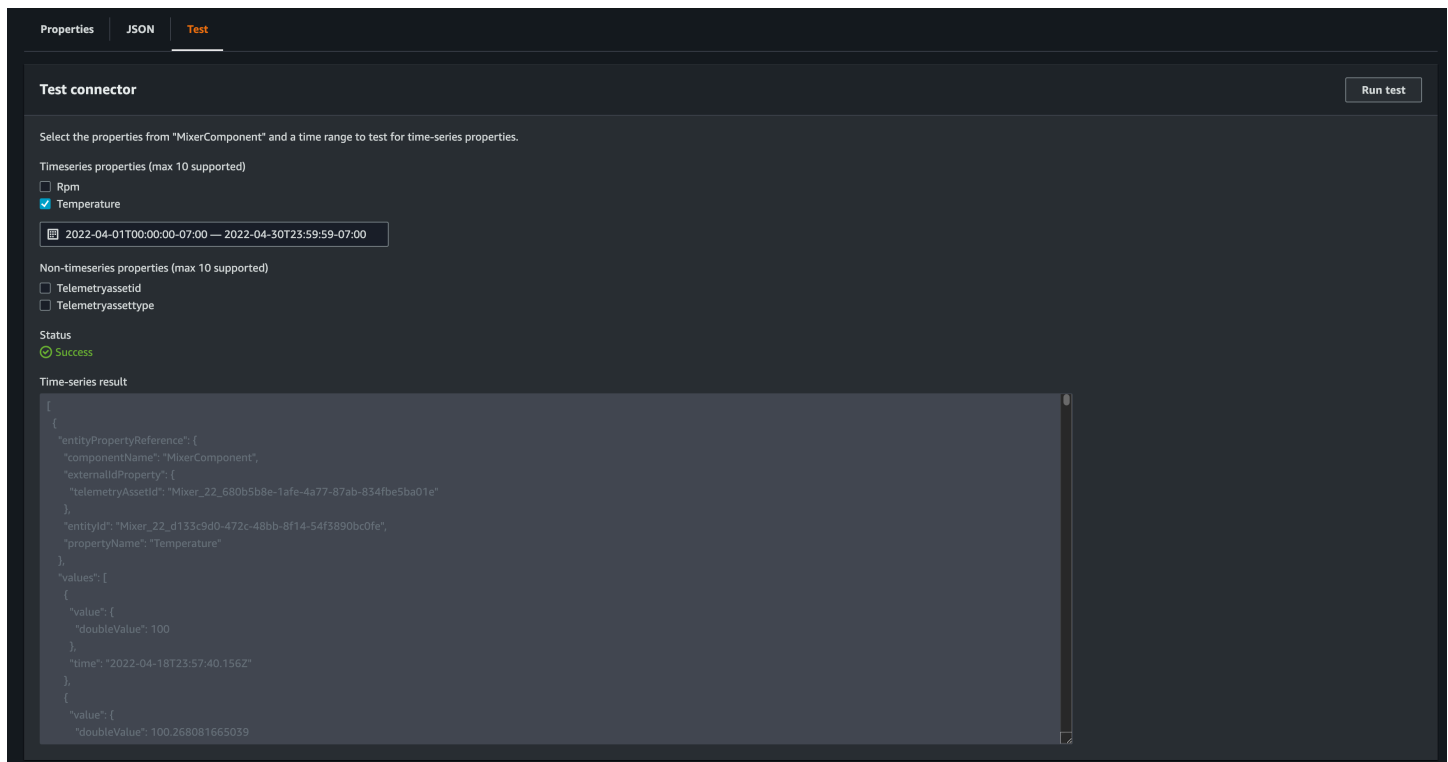
## Test di integrazione

Ti consigliamo di utilizzare un test integrato AWS IoT TwinMaker per verificare che la query sul piano dati funzioni end-to-end. È possibile eseguire questa operazione tramite [GetPropertyValueHistory](#) API o facilmente [AWS IoT TwinMaker dalla console](#).



The screenshot shows the AWS IoT TwinMaker console interface for a component named 'MixerComponent'. The breadcrumb navigation 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, and Status: ACTIVE. There are tabs for 'Properties', 'JSON', and 'Test', with 'Test' being the active tab. The 'Test connector' section is visible, featuring a 'Run test' button and instructions to select properties from 'MixerComponent' and a time range. 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.

Nella AWS IoT TwinMaker console, vai ai dettagli del componente, quindi nella sezione Test, vedrai che tutte le proprietà del componente sono elencate lì. L'area Test della console consente di testare sia le proprietà delle serie temporali che le non-time-series proprietà. Per le proprietà delle serie temporali è inoltre possibile utilizzare l' [GetPropertyValueHistory](#) API e per non-time-series le proprietà utilizzare l' [GetPropertyValue](#) API. Se il connettore Lambda supporta l'interrogazione di più proprietà, puoi scegliere più di una proprietà.



## Cosa c'è dopo

Ora puoi configurare una [dashboard AWS IoT TwinMaker Grafana per visualizzare le metriche](#). Puoi anche esplorare altri esempi di connettori dati nel [GitHub repository degli AWS IoT TwinMaker esempi](#) per vedere se sono adatti al tuo caso d'uso.

## AWS IoT TwinMaker esempio di connettore per serie temporali di cookie factory

Il [codice completo della funzione Lambda di cookie factory](#) è disponibile su GitHub. Sebbene sia ancora possibile aggiornare l'implementazione dopo aver collegato il connettore al tipo di componente, consigliamo vivamente di verificare il connettore Lambda prima dell'integrazione con AWS IoT TwinMaker. Puoi testare la tua funzione Lambda nella console Lambda o localmente in AWS CDK. [Per ulteriori informazioni sul test delle funzioni Lambda, consulta Test delle funzioni Lambda e Test locale delle applicazioni. AWS CDK](#)

## Esempi di tipi di componenti di cookie factory

In un tipo di componente, definiamo proprietà comuni condivise tra i componenti. Per l'esempio di cookie factory, i componenti fisici dello stesso tipo condividono le stesse misurazioni, quindi possiamo

definire lo schema di misurazione nel tipo di componente. Ad esempio, il tipo di mixer è definito nell'esempio seguente.

```
{
  "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
    }
  }
}
```

Ad esempio, un componente fisico potrebbe contenere misurazioni in un database Timestream, registri di manutenzione in un database SQL o dati di allarme in sistemi di allarme. La creazione di più componenti e la loro associazione a un'entità collega diverse fonti di dati all'entità e compila il grafico entità-componente. In questo contesto, ogni componente necessita di una `telemetryId` proprietà per identificare la chiave univoca del componente nella fonte di dati corrispondente. La specificazione della `telemetryId` proprietà presenta due vantaggi: la proprietà può essere utilizzata nel connettore dati come condizione di filtro per interrogare solo i valori del componente specificato e, se si include il valore della `telemetryId` proprietà nella risposta dell'API del piano dati, il lato client acquisisce l'ID e può eseguire una ricerca inversa, se necessario.

Se si aggiunge `TelemetryId` al tipo di componente come ID esterno, il componente viene identificato nella tabella. `TimeStream`

```
{
  "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
    }
}
}
}

```

Allo stesso modo abbiamo il tipo di componente per `WaterTank`, come mostrato nel seguente esempio JSON.

```

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

```



```

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

```

La `TelemetryType` è una proprietà opzionale nel tipo di componente se è finalizzata all'interrogazione dei valori delle proprietà nell'ambito dell'entità. Per un esempio, vedete i tipi di componenti definiti nel [GitHub repository degli AWS IoT TwinMaker esempi](#). Nella stessa tabella sono incorporati anche tipi di allarme, quindi vengono definiti e si estraggono proprietà comuni come il `TelemetryId` tipo di componente principale `TelemetryType` da condividere con altri tipi secondari. `TelemetryType`

## Esempio: Lambda

Il connettore Lambda deve accedere all'origine dati e generare l'istruzione di query in base all'input e inoltrarla all'origine dati. Un esempio di richiesta inviata alla Lambda è illustrato nel seguente esempio JSON.

```

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

```

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

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

L'obiettivo della funzione Lambda è interrogare i dati di misurazione storici per una determinata entità. AWS IoT TwinMaker fornisce una mappa delle proprietà dei componenti ed è necessario specificare un valore istanziato per l'ID del componente. Ad esempio, per gestire la query a livello di tipo di componente (che è comune nei casi d'uso degli allarmi) e restituire lo stato di allarme di tutti i componenti nell'area di lavoro, la mappa delle proprietà contiene le definizioni delle proprietà dei tipi di componente.

Nel caso più semplice, come nella richiesta precedente, vogliamo una serie di campioni di temperatura durante la finestra temporale specificata per il componente specificato, in ordine temporale crescente. L'istruzione di interrogazione può essere riassunta come segue:

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

# Creazione e modifica di AWS IoT TwinMaker scene

Le scene sono visualizzazioni tridimensionali del tuo gemello digitale. Sono il modo principale per modificare il tuo gemello digitale. Scopri come aggiungere allarmi, dati di serie temporali, sovrapposizioni di colori, tag e regole visive alla tua scena per allineare le tue visualizzazioni gemelle digitali al tuo caso d'uso reale.

Questa sezione comprende i seguenti argomenti:

- [Prima di creare la tua prima scena](#)
- [Caricare risorse nella Libreria AWS IoT TwinMaker delle risorse](#)
- [Crea le tue scene](#)
- [Aggiungi telecamere fisse alle entità](#)
- [Modifica migliorata della scena](#)
- [Modifica le tue scene](#)
- [Formato del modello 3D Tiles](#)

## Prima di creare la tua prima scena

Le scene si basano sulle risorse per rappresentare il tuo gemello digitale. Queste risorse sono costituite da modelli 3D, dati o file di texture. Le dimensioni e la complessità delle risorse, gli elementi della scena come l'illuminazione e l'hardware del computer influiscono sulle prestazioni delle AWS IoT TwinMaker scene. Utilizzate le informazioni contenute in questo argomento per ridurre il ritardo, i tempi di caricamento e migliorare la frequenza dei fotogrammi delle scene.

## Ottimizza le tue risorse prima di importarle in AWS IoT TwinMaker

Puoi usarlo AWS IoT TwinMaker per interagire con il tuo gemello digitale in tempo reale. Per un'esperienza ottimale con le scene, consigliamo di ottimizzare le risorse per utilizzarle in un ambiente in tempo reale.

I tuoi modelli 3D possono avere un impatto significativo sulle prestazioni. La geometria e le mesh complesse dei modelli possono ridurre le prestazioni. Ad esempio, i modelli CAD industriali presentano un elevato livello di dettaglio. Consigliamo di comprimere le mesh di questi modelli e di

ridurre il numero di poligoni prima di utilizzarle nelle scene. AWS IoT TwinMaker Se state creando nuovi modelli 3D perAWS IoT TwinMaker, dovrete stabilire un livello di dettaglio e mantenerlo in tutti i modelli. Rimuovi dai modelli i dettagli che non influiscono sulla visualizzazione o sull'interpretazione del tuo caso d'uso.

Per comprimere i modelli e ridurre le dimensioni dei file, utilizzate strumenti di compressione mesh open source, come la compressione dei dati [3D DRACO](#).

Anche le texture non ottimizzate possono influire sulle prestazioni. Se non hai bisogno di alcuna trasparenza nelle tue texture, prendi in considerazione la possibilità di scegliere il formato di immagine PEG rispetto al formato PNG. È possibile comprimere i file di texture utilizzando strumenti di compressione delle texture open source, come [Basis Universal](#).

## Le migliori pratiche per le prestazioni in AWS IoT TwinMaker

Per ottenere prestazioni ottimali conAWS IoT TwinMaker, tieni presente le seguenti limitazioni e procedure consigliate.

- AWS IoT TwinMakerle prestazioni di rendering delle scene dipendono dall'hardware. Le prestazioni variano in base alle diverse configurazioni hardware del computer.
- Consigliamo un numero totale di poligoni inferiore a 1 milione su tutti gli oggetti del tuo. AWS IoT TwinMaker
- Consigliamo un totale di 200 oggetti per scena. L'aumento del numero di oggetti in una scena oltre 200 può ridurre la frequenza dei fotogrammi della scena.
- Consigliamo che la dimensione totale di tutte le risorse 3D uniche presenti nella scena non superi i 100 megabyte. In caso contrario, potreste riscontrare tempi di caricamento lenti o prestazioni ridotte a seconda del browser e dell'hardware.
- Per impostazione predefinita, le scene hanno un'illuminazione ambientale. È possibile aggiungere luci supplementari a una scena per mettere a fuoco determinati oggetti o proiettare ombre sugli oggetti. Ti consigliamo di usare una luce per scena. Usa le luci dove necessario ed evita di replicare le luci del mondo reale all'interno di una scena.

## Ulteriori informazioni

Utilizza queste risorse per saperne di più sulle tecniche di ottimizzazione che puoi utilizzare per migliorare le prestazioni nelle tue scene.

- [Come convertire e comprimere i modelli OBJ in GLTF da utilizzare con AWS IoT TwinMaker](#)
- [Ottimizza i tuoi modelli 3D per i contenuti web](#)
- [Ottimizzazione delle scene per prestazioni WebGL migliori](#)

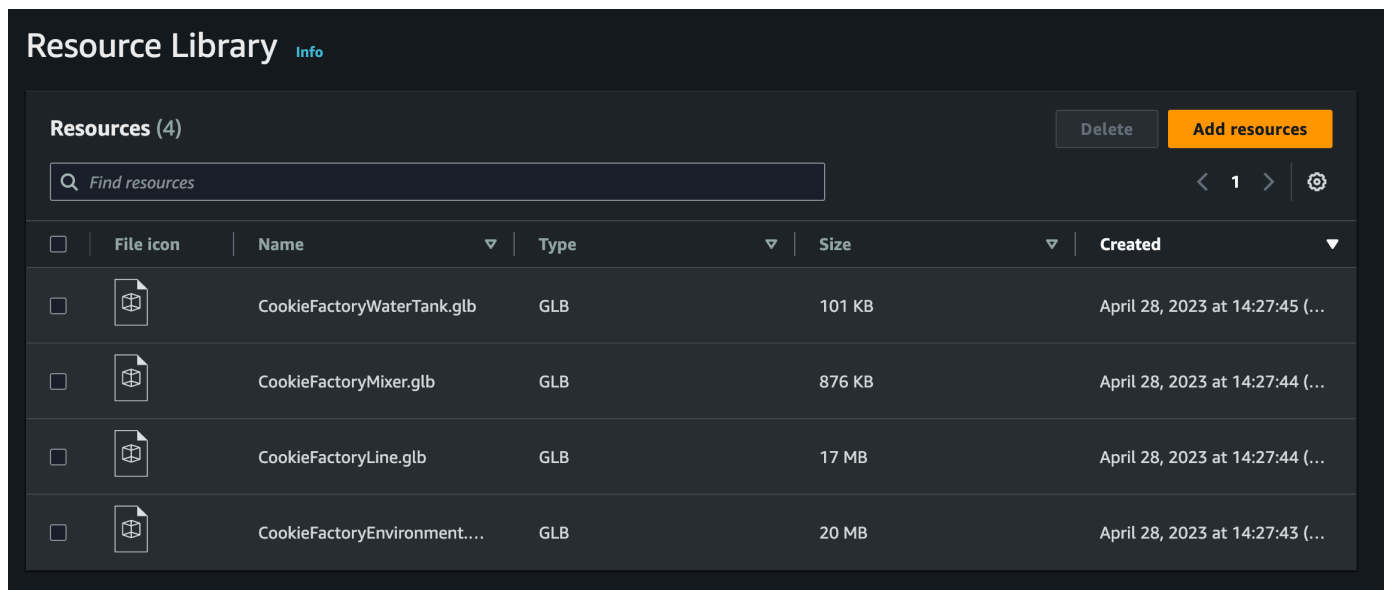
## Caricare risorse nella Libreria AWS IoT TwinMaker delle risorse

Puoi utilizzare la Resource Library per controllare e gestire qualsiasi risorsa che desideri inserire nelle scene della tua applicazione digital twin. Per AWS IoT TwinMaker conoscere le risorse, caricale utilizzando la pagina della console Resource Library.

### Carica i file nella Resource Library utilizzando la console

Segui questi passaggi per aggiungere file alla Libreria di risorse utilizzando la AWS IoT TwinMaker console.

1. Nel menu di navigazione a sinistra, in Spazi di lavoro, seleziona Libreria di risorse.
2. Seleziona Aggiungi risorse e scegli i file che desideri caricare.



## Crea le tue scene

In questa sezione, configurerai una scena in modo da poter modificare il tuo gemello digitale. Dovrai creare una o più scene prima di creare il tuo gemello digitale. Esistono due approcci principali per creare una scena: puoi importare un singolo file glTF contenente l'intero modello del tuo sito.

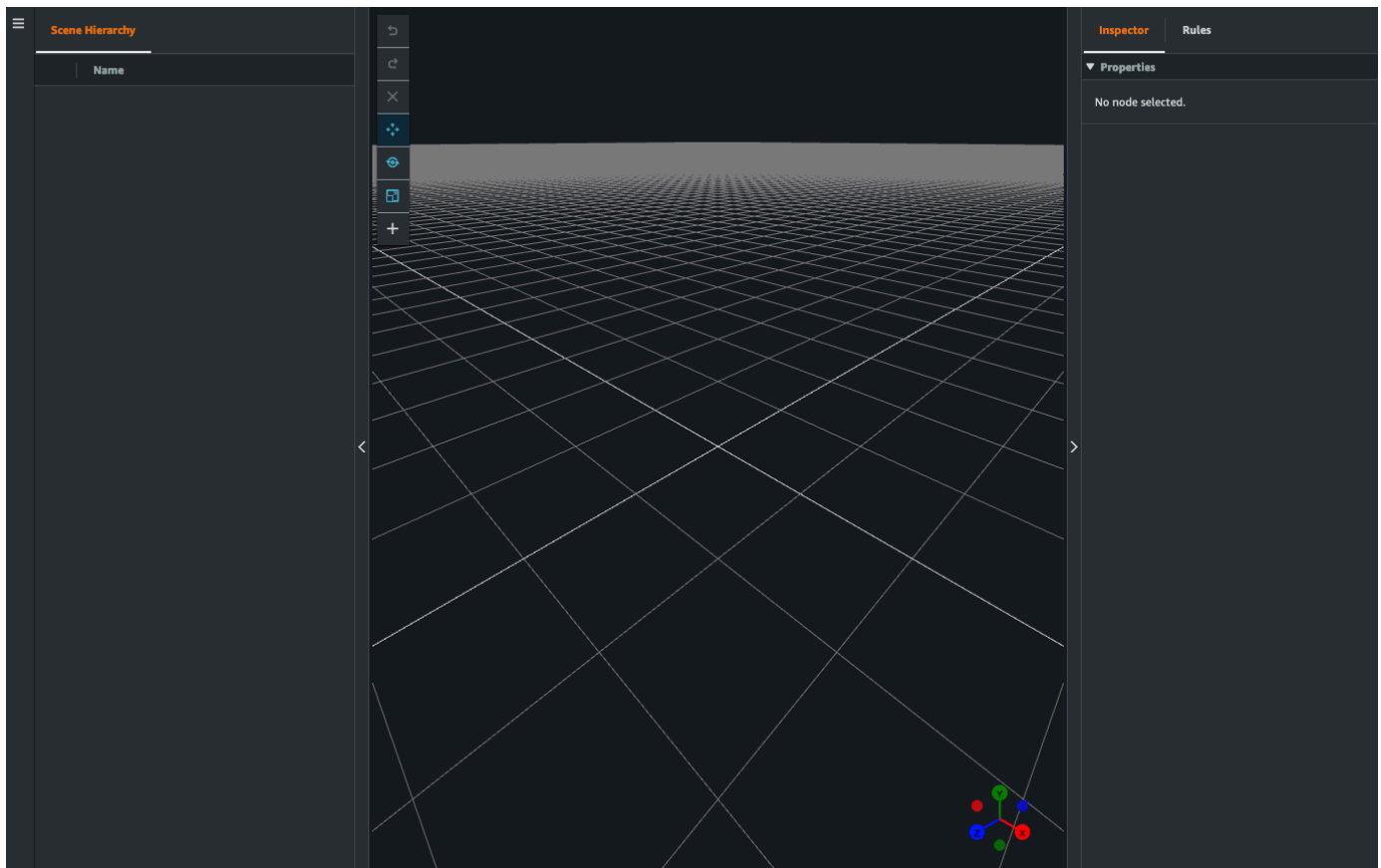
In alternativa, potete importare un modello di ambiente, come un edificio o uno spazio, e quindi importare file glTF separati per ogni apparecchiatura e posizionarli rispetto alla loro posizione fisica.

### Note

Prima di creare una scena, è necessario aver creato un'area di lavoro.

Utilizzate la procedura seguente per creare la scena in AWS IoT TwinMaker.

1. Per aprire il riquadro della scena, nella barra di navigazione a sinistra dell'area di lavoro, scegliete Scene Composer.
2. Scegliere Create scene (Crea scena). Si apre il nuovo riquadro di creazione della scena.
3. Nel riquadro di creazione della scena, inserisci un nome e una descrizione per la nuova scena. Quando sei pronto per creare la scena, scegli Crea scena. La nuova scena si apre ed è pronta per essere utilizzata.

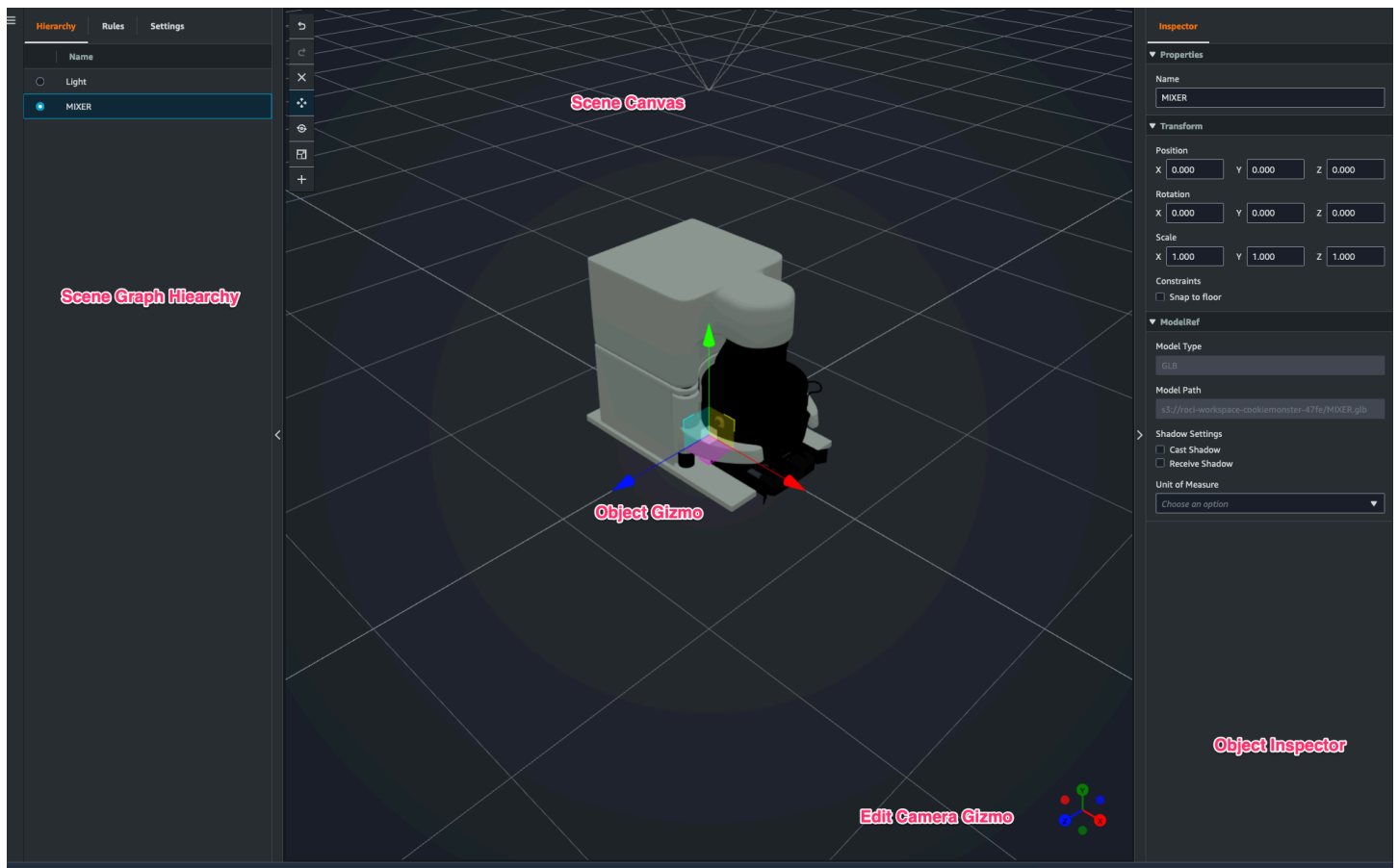


## Usa la navigazione 3D AWS IoT TwinMaker nelle tue scene

La AWS IoT TwinMaker scena dispone di una serie di controlli di navigazione che puoi utilizzare per navigare in modo efficiente nello spazio 3D della scena. Per interagire con lo spazio 3D e gli oggetti rappresentati dalla scena, utilizzate i seguenti widget e opzioni di menu.

- **Inspector:** utilizzate la finestra Inspector per visualizzare e modificare le proprietà e le impostazioni di un'entità o di un componente selezionato nella gerarchia.
- **Scene Canvas:** Scene Canvas è lo spazio 3D in cui è possibile posizionare e orientare tutte le risorse 3D che si desidera utilizzare.
- **Scene Graph Hierarchy:** puoi usare questo pannello per vedere tutte le entità presenti nella tua scena. Viene visualizzato sul lato sinistro della finestra.
- **Oggetto gizmo:** Usa questo aggeggio per spostare oggetti sull'area di disegno. Viene visualizzato al centro di un oggetto 3D selezionato nella Scene Canvas.
- **Modifica gizmo telecamera:** utilizzate il gizmo Edit Camera per visualizzare rapidamente l'orientamento attuale della telecamera di visualizzazione della scena e modificare l'angolo di visione. Puoi trovare questo aggeggio nell'angolo in basso a destra della vista della scena.
- **Controlli dello zoom:** per navigare nella Scene Canvas, fate clic con il pulsante destro del mouse e trascinate nella direzione in cui desiderate muovervi. Per ruotare, fate clic con il pulsante sinistro del mouse e trascinate per ruotare. Per ingrandire, usa la rotellina di scorrimento del mouse oppure pizzica e allontana le dita sul trackpad del laptop.





I pulsanti di scena nel riquadro della gerarchia hanno le seguenti funzioni elencate, in ordine di layout dei pulsanti:

- Annulla: annulla l'ultima modifica apportata alla scena.
- Ripeti: ripeti l'ultima modifica apportata alla scena.
- Plus (+): utilizzate questo pulsante per accedere alle seguenti azioni: Aggiungi nodo vuoto, Aggiungi modello 3D, Aggiungi tag, Aggiungi luce e Aggiungi modello shader.
- Cambia il metodo di navigazione: accedi alle opzioni di navigazione della telecamera di scena, Orbit e Pan.
- Cestino (elimina): utilizzate questo pulsante per eliminare un oggetto selezionato nella scena.
- Strumenti di manipolazione degli oggetti: utilizzate questo pulsante per traslare, ruotare e ridimensionare l'oggetto selezionato.

## Aggiungi telecamere fisse alle entità

Puoi collegare viste fisse con telecamera alle tue entità all'interno delle AWS IoT TwinMaker scene. Queste telecamere offrono una prospettiva fissa su un modello 3D, che consente di spostare rapidamente e facilmente la prospettiva di una scena verso un'entità mirata.

1. Accedi alla scena nella [AWS IoT TwinMaker console](#).
2. Nel menu della gerarchia delle scene, seleziona l'entità a cui desideri collegare la videocamera.
3. Premi il pulsante + e dal menu a discesa seleziona Aggiungi telecamera dalla vista corrente. Per applicare una telecamera con la prospettiva corrente all'entità.
4. Nell'ispettore, è possibile configurare la videocamera e regolare le seguenti impostazioni:
  - Nome di una telecamera
  - La posizione e la rotazione della telecamera
  - La lunghezza focale della fotocamera
  - Il livello di zoom
  - Piani di ritaglio vicini e lontani
5. Per accedere alla fotocamera dopo averla posizionata. Seleziona l'entità a cui hai aggiunto la videocamera nella gerarchia. Cerca il nome della telecamera elencato sotto l'entità.
6. Dopo aver selezionato la telecamera posizionata dall'entità, la vista della telecamera della scena si allineerà alla prospettiva impostata della telecamera posizionata.

## Modifica migliorata della scena

AWS IoT TwinMaker le scene dispongono di una serie di strumenti per il miglioramento, la modifica e la manipolazione delle risorse presenti nella scena.

I seguenti argomenti spiegano come utilizzare le funzionalità di modifica avanzate nelle AWS IoT TwinMaker scene.

- [Posizionamento mirato degli oggetti della scena](#)
- [Selezione del sottomodello](#)
- [Modifica le entità nella gerarchia delle scene](#)

## Posizionamento mirato degli oggetti della scena

AWS IoT TwinMaker consente di posizionare e aggiungere oggetti con precisione nella scena. Questa funzionalità di modifica avanzata offre un maggiore controllo su dove posizionare tag, entità, luci e modelli nella scena.

1. Accedi alla scena nella [AWS IoT TwinMaker console](#).
2. Premi il pulsante + e dal menu a discesa seleziona una delle opzioni. Potrebbe essere un modello, una luce, un tag o qualsiasi altra cosa dal menu +.

Quando sposti il cursore nello spazio 3d della scena, dovresti vedere un bersaglio attorno al cursore.

3. Usa il bersaglio per posizionare con precisione gli elementi nella scena.

## Selezione del sottomodello

AWS IoT TwinMaker consente di selezionare sottomodelli di modelli 3D nelle scene e di applicare loro proprietà standard, come tag, luci o regole.

I formati di file dei modelli 3D contengono metadati che possono specificare sottoaree del modello come sottomodelli all'interno del modello più grande. Ad esempio, un modello potrebbe essere un sistema di filtrazione, le singole parti del sistema come serbatoi, tubi o un motore sono contrassegnate come sottomodelli del modello 3D della filtrazione.

Formati di file 3D supportati nelle scene: GLB e GLTF.

1. [Accedi alla tua scena nella console AWS IoT TwinMaker](#)
2. Se non hai modelli nella tua scena, assicurati di aggiungerne uno selezionando l'opzione dal menu +.
3. Seleziona il modello elencato nella gerarchia della scena. Una volta selezionato, la gerarchia dovrebbe mostrare tutti i sottomodelli al di sotto del modello.

### Note

Se non vedi alcun sottomodello nell'elenco, è probabile che il modello non fosse configurato per avere sottomodelli.

4. Per cambiare la visibilità di un sottomodello, premi l'icona a forma di occhio, situata a destra del nome del sottomodello nella gerarchia.
5. Per modificare i dati del sottomodello, come il nome o la posizione, l'ispettore della scena si apre quando viene selezionato un sottomodello. Utilizzate il menu di ispezione per aggiornare o modificare i dati del sottomodello.
6. Per aggiungere tag, luci, regole o altre proprietà ai sottomodelli, premi +, mentre il sottomodello è selezionato nella gerarchia.

## Modifica le entità nella gerarchia delle scene

AWS IoT TwinMaker le scene consentono di modificare direttamente le proprietà delle entità all'interno della tabella della gerarchia. La procedura seguente mostra quali azioni è possibile eseguire su un'entità tramite il menu della gerarchia.

1. Passa alla scena nella [AWS IoT TwinMaker console](#).
2. Apri la gerarchia delle scene e seleziona un sottoelemento di un'entità che desideri manipolare.
3. Una volta selezionato l'elemento, premi il pulsante + e dal menu a discesa seleziona una delle opzioni:
  - Aggiungi un nodo vuoto
  - Aggiungi modello 3D
  - Aggiungi luce
  - Aggiungi videocamera dalla visualizzazione corrente
  - Aggiungi tag
  - Aggiungi model shader
  - Aggiungi un indicatore di movimento
4. Dopo aver selezionato una delle opzioni dal menu a discesa, la selezione verrà applicata alla scena come elemento secondario dell'elemento selezionato dal passaggio 2.
5. È possibile riordinare gli elementi secondari e gli elementi secondari selezionando un elemento secondario e trascinandolo nella gerarchia fino a un nuovo elemento principale.

## Aggiungere annotazioni alle entità

Il compositore di AWS IoT TwinMaker scene consente di annotare qualsiasi elemento nella gerarchia delle scene. L'annotazione è scritta in markdown.

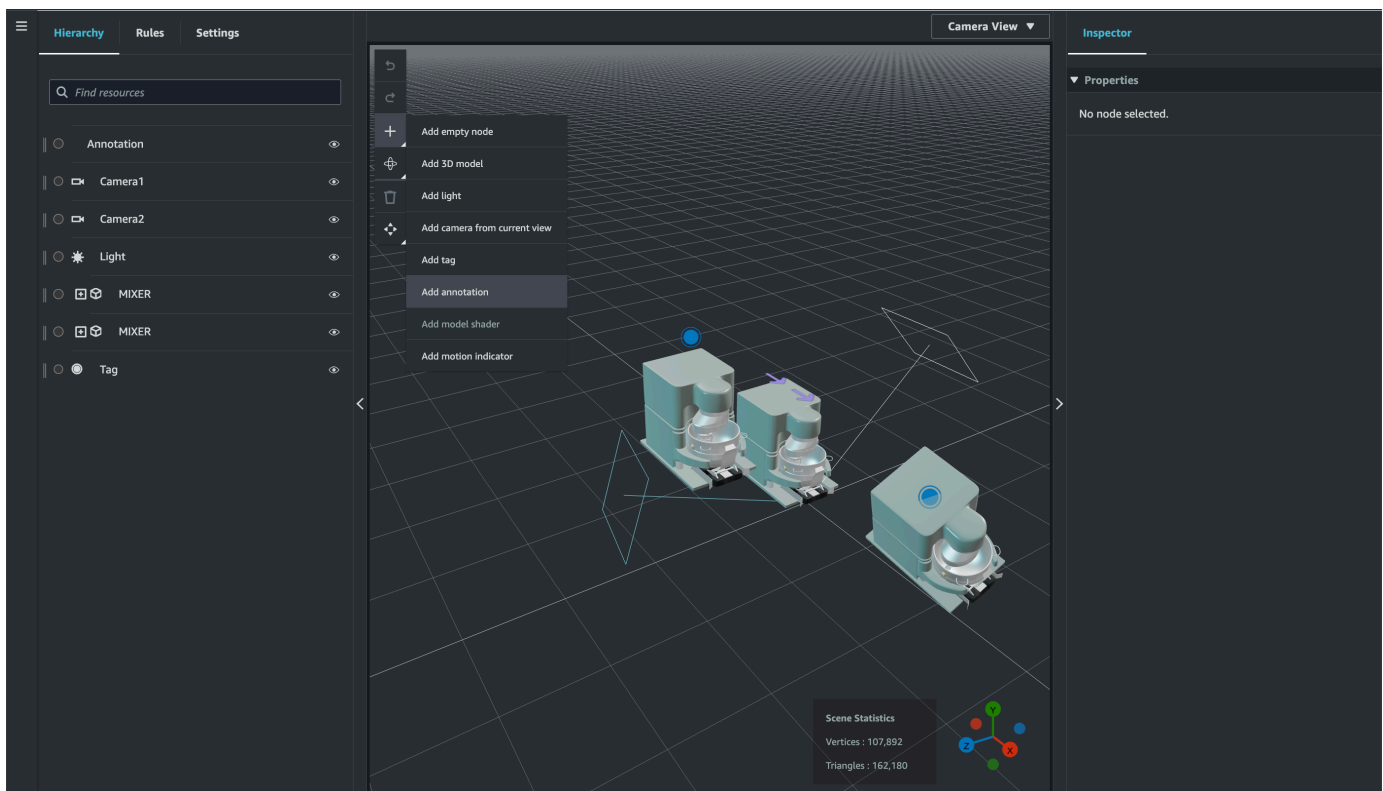
[Per ulteriori informazioni sulla scrittura in Markdown, consulta la documentazione ufficiale sulla sintassi di markdown, Basic Syntax.](#)

### Note

AWS IoT TwinMaker annotazioni e sovrapposizione solo della sintassi Markdown e non dell'HTML.

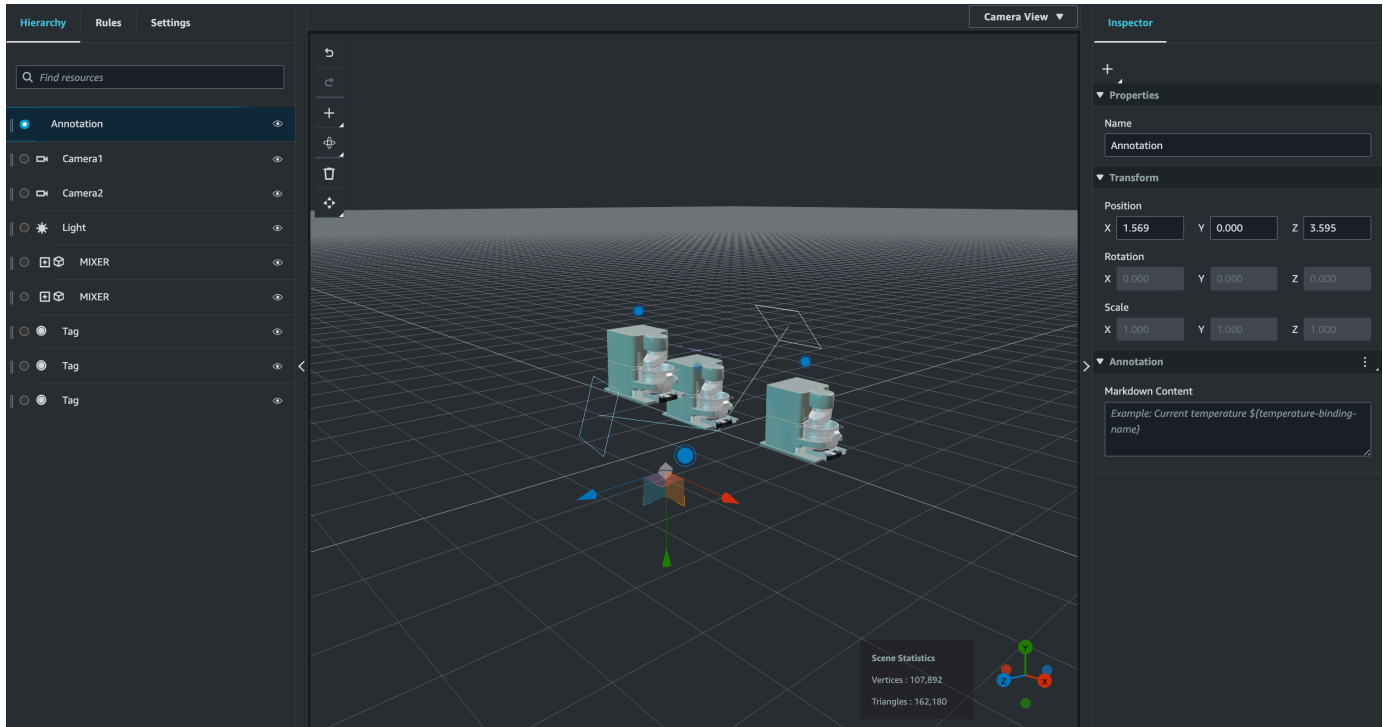
Aggiungi un'annotazione a un'entità

1. Vai alla tua scena nella [AWS IoT TwinMaker console](#).
2. Seleziona un elemento dalla gerarchia delle scene che desideri annotare. Se non è selezionato alcun elemento nella gerarchia, è possibile aggiungere annotazioni alla radice.
3. Premi il pulsante più + e scegli l'opzione Aggiungi annotazione.



4. Nella finestra Inspector a sinistra, scorri verso il basso fino alla sezione delle annotazioni. Usando la sintassi Markdown, scrivi il testo che desideri visualizzare nell'annotazione.

[Per ulteriori informazioni sulla scrittura in Markdown, consulta la documentazione ufficiale sulla sintassi di markdown, Basic Syntax.](#)



5. Per associare i dati della AWS IoT TwinMaker scena a un'annotazione, scegli Aggiungi associazione dati, aggiungi l'ID dell'entità, quindi seleziona il nome del componente e il nome della proprietà dell'entità da cui desideri far emergere i dati. Puoi aggiornare il nome dell'associazione per utilizzarlo come variabile Markdown e far apparire i dati nell'annotazione.

# Inspector



## ▼ Properties

### Name

Annotation

## ▼ Transform

### Position

X 1.569

Y 0.000

Z 3.595

### Rotation

X 0.000

Y 0.000

Z 0.000

### Scale

X 1.000

Y 1.000

Z 1.000

## ▼ Annotation



Add data binding

### Markdown Content

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

## Inspector



### ▼ Properties

#### Name

### ▼ Transform

#### Position

X

Y

Z

#### Rotation

X

Y

Z

#### Scale

X

Y

Z

### ▼ Annotation ⋮

#### Markdown Content

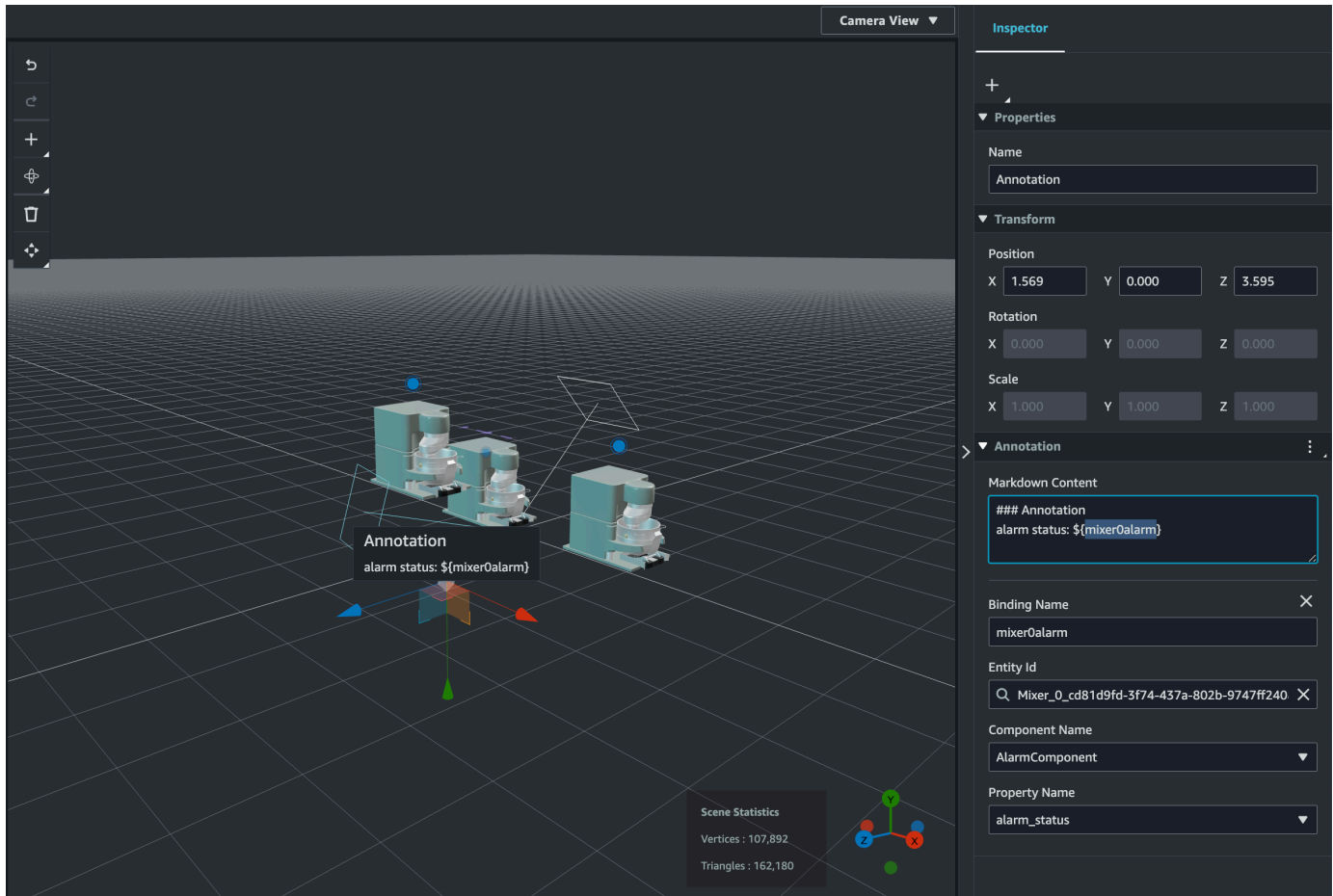
```
Example: Current temperature  $\{temperature-binding-name\}$ 
```



## 6. Il nome dell'associazione viene utilizzato per rappresentare la variabile dell'annotazione.

Inserisci un nome vincolante per visualizzare il valore storico più recente della serie temporale di un'entità nell'annotazione tramite AWS IoT TwinMaker la sintassi variabile: `${variable-name}`

Ad esempio, questa sovrapposizione mostra il valore `dimixer0alarm`, nell'annotazione con la sintassi `${mixer0alarm}`



## Aggiungi sovrapposizioni ai tag

Puoi creare sovrapposizioni per le tue scene. AWS IoT TwinMaker Le sovrapposizioni di scene sono associate ai tag e possono essere utilizzate per far emergere dati critici associati alle entità della scena. L'overlay viene creato e renderizzato in Markdown.

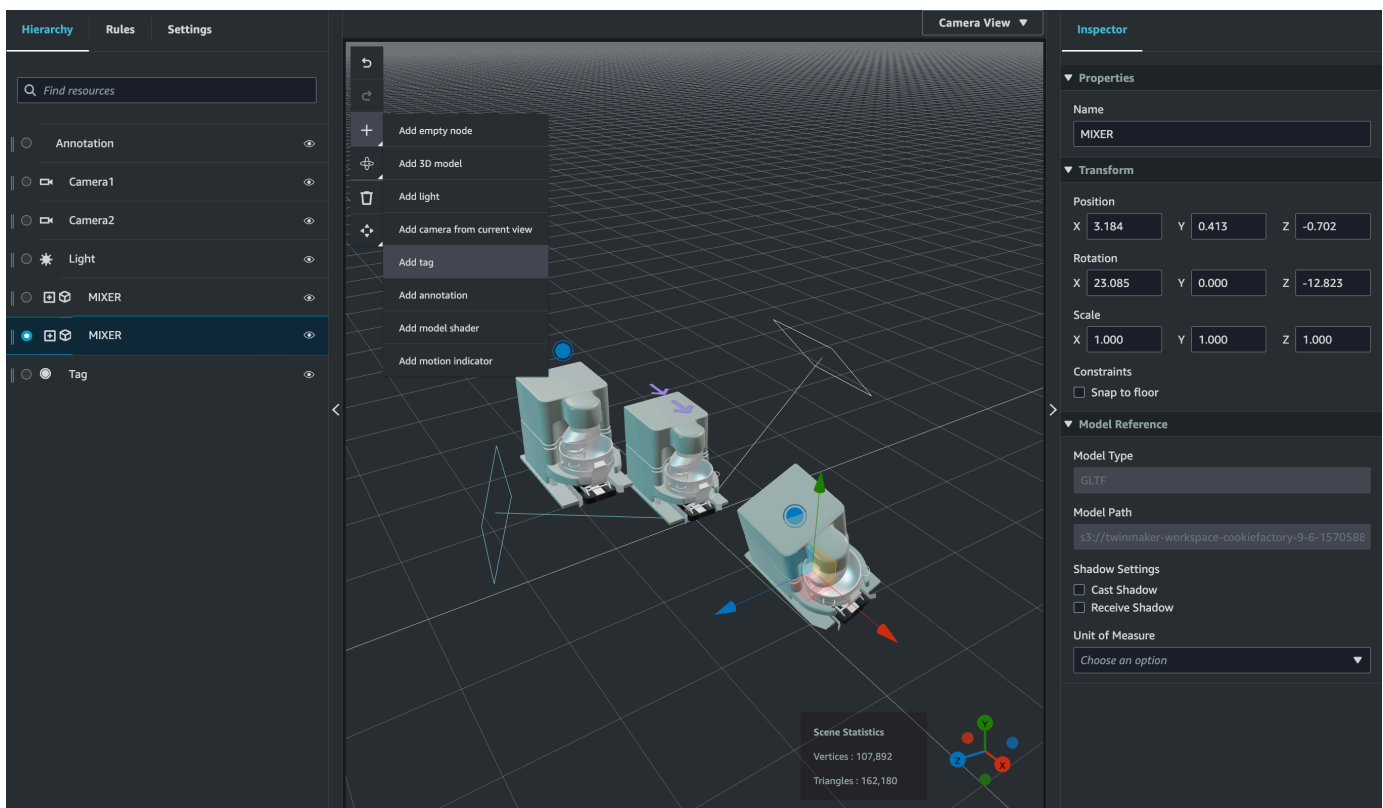
[Per ulteriori informazioni sulla scrittura in Markdown, consulta la documentazione ufficiale sulla sintassi di markdown, Basic Syntax.](#)

**Note**

Per impostazione predefinita, un Overlay è visibile in una scena solo quando è selezionato il tag ad esso associato. Puoi attivarlo nelle Impostazioni della scena in modo che tutte le sovrapposizioni siano visibili contemporaneamente.

1. [Vai alla tua scena nella console AWS IoT TwinMaker](#)
2. L' AWS IoT TwinMaker overlay è associato a una scena di tag, puoi aggiornare un tag esistente o aggiungerne uno nuovo.

Premi il pulsante più + e scegli l'opzione Aggiungi tag.



3. Nel pannello Inspector a destra, seleziona il pulsante + (simbolo più), quindi seleziona Aggiungi sovrapposizione.

## 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*

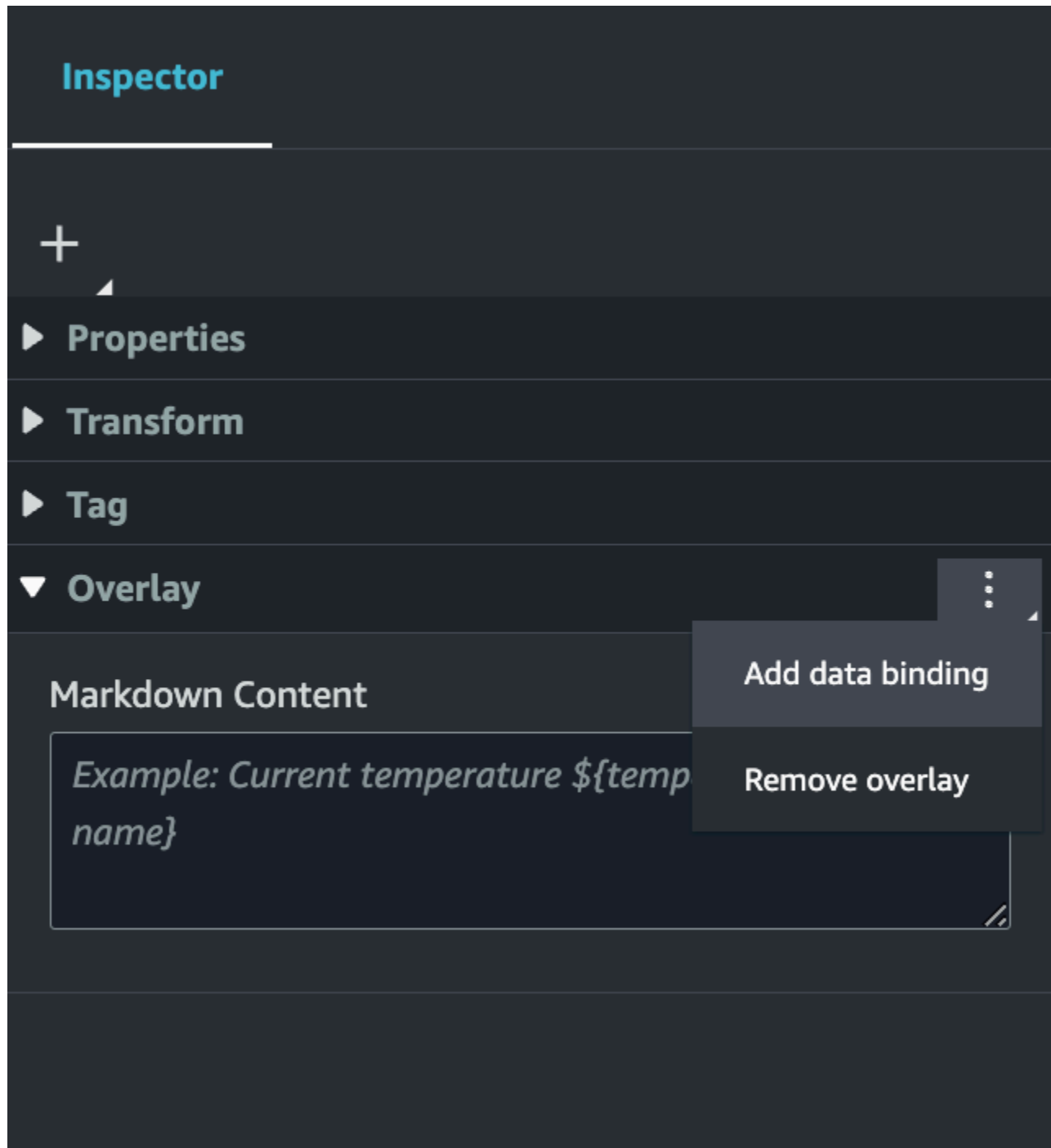


### Link Target

4. Nella sintassi Markdown, scrivi il testo che desideri visualizzare nella sovrapposizione.

[Per ulteriori informazioni sulla scrittura in Markdown, consulta la documentazione ufficiale sulla sintassi di markdown, Basic Syntax.](#)

5. Per associare i dati della AWS IoT TwinMaker scena a una sovrapposizione, seleziona Aggiungi associazione dati.



Aggiungi il nome dell'associazione e l'ID dell'entità, quindi seleziona il nome del componente e il nome della proprietà dell'entità da cui desideri far emergere i dati.

6. È possibile visualizzare il valore storico più recente dei dati della serie temporale di un'entità nella sovrapposizione tramite AWS IoT TwinMaker la sintassi variabile: `${variable-name}`

Ad esempio, questa sovrapposizione mostra il valore `dimixer0alarm`, nella sovrapposizione con la sintassi `${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. Per abilitare la visibilità dell'Overlay, apri la scheda Impostazioni in alto a sinistra e assicurati che l'interruttore per Overlay sia attivato in modo che tutti gli Overlay siano visibili contemporaneamente.

 Note

Per impostazione predefinita, una sovrapposizione è visibile in una scena solo quando è selezionato il tag ad essa associato.

**Hierarchy** | **Rules** | **Settings**

▼ **Current View Setting**

**Toggle visibility**

Motion indicator

Tags

Overlay

Annotation

▼ **Scene Settings**

Environment Preset

*Choose an environment* ▼

▼ **Tag Settings**

Scale

Fixed scaling

**Overlay**

Visibility settings

Always on

▼ **Data Binding Template**

sel\_entity

*Select an option* ▼

sel\_comp



## Modifica le tue scene

Dopo aver creato una scena, puoi aggiungere entità, componenti e configurare widget aumentati nella scena. Usa i componenti e i widget delle entità per modellare il tuo gemello digitale e fornire funzionalità adatte al tuo caso d'uso.

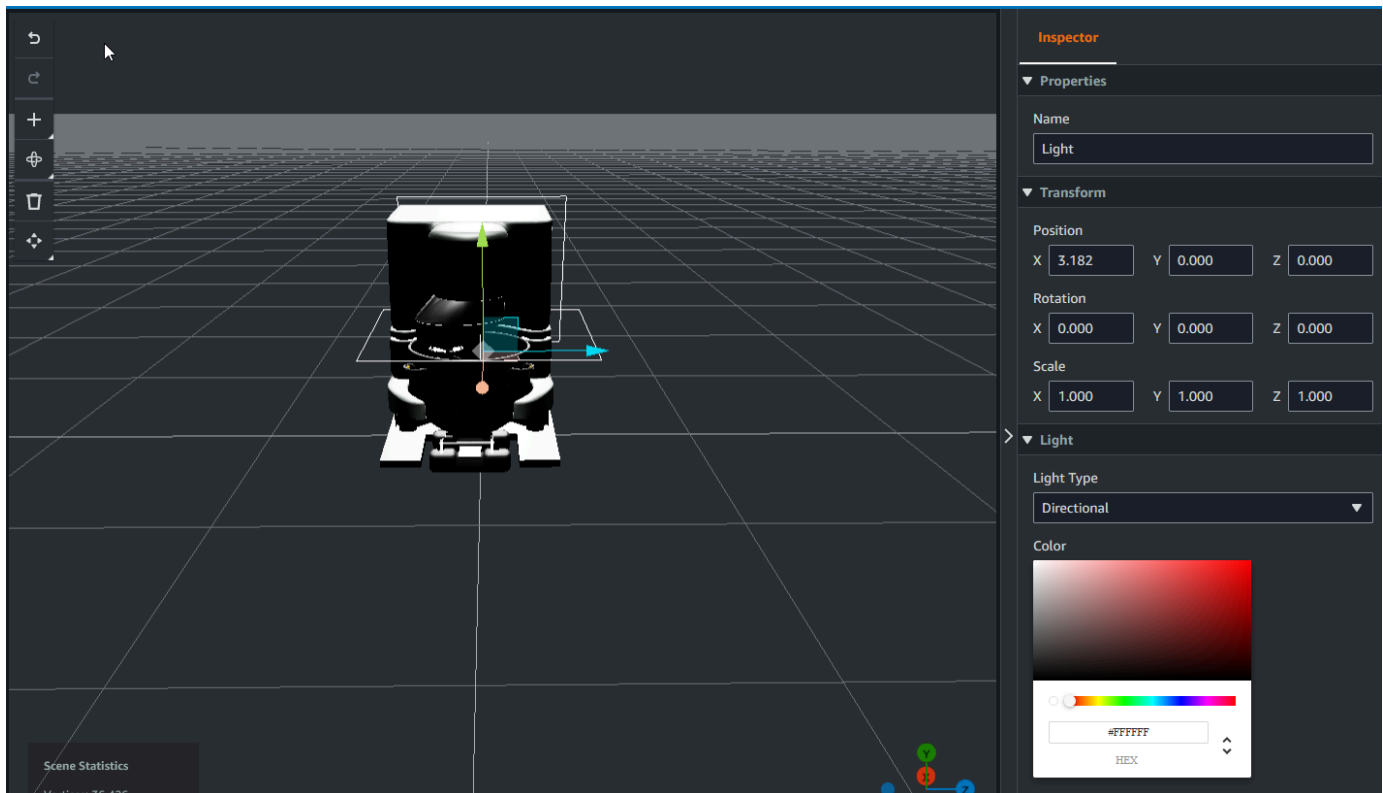
## Aggiungi modelli alle tue scene

Per aggiungere modelli alla scena, utilizzate la procedura seguente.

### Note

Per aggiungere modelli alla scena, è necessario prima caricare i modelli nella Libreria delle AWS IoT TwinMaker risorse. Per ulteriori informazioni, consulta [Caricare risorse nella Libreria AWS IoT TwinMaker delle risorse](#).

1. Nella pagina del compositore della scena, scegliete il segno più (+), quindi scegliete Aggiungi modello 3D.
2. Nella finestra Aggiungi risorsa dalla libreria di risorse, scegli il CookieFactorMixerfile.glb, quindi scegli Aggiungi. Si apre Scene Composer.
3. Facoltativo: scegli il segno più (+), quindi scegli Aggiungi luce.
4. Scegli ciascuna opzione di luce per vedere come influiscono sulla scena.



### Note

Le scene hanno un'illuminazione ambientale predefinita. Per evitare una perdita di frame rate, considerate la possibilità di limitare il numero di luci aggiuntive posizionate nella scena.

## Aggiungi i widget dell'interfaccia utente aumentata Model Shader alla tua scena

I widget Model Shader possono cambiare il colore di un oggetto in base a condizioni da te definite. Ad esempio, potete creare un widget colorato che cambi il colore di un mixer per biscotti nella vostra scena in base ai dati di temperatura del mixer.

Utilizzate la seguente procedura per aggiungere i widget dello shader del modello a un oggetto selezionato.

1. Selezionate un oggetto nella gerarchia a cui desiderate aggiungere un widget. Premi il pulsante +, quindi scegli Model Shader.

2. Per aggiungere un nuovo gruppo di regole visive, segui innanzitutto le istruzioni riportate di seguito per creare il ColorRule, quindi nel pannello Inspector per l'oggetto dell'ID regola, scegli. ColorRule
3. Selezionate EntityID e PropertyName desiderate associare lo shader del modello. ComponentName


## Crea regole visive per le tue scene

Puoi utilizzare le mappe visive delle regole per specificare le condizioni basate sui dati che modificano l'aspetto visivo di un widget dell'interfaccia utente aumentata, come un tag o uno shader del modello. Sono disponibili regole di esempio, ma puoi anche crearne di personalizzate. L'esempio seguente mostra una regola visiva.

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

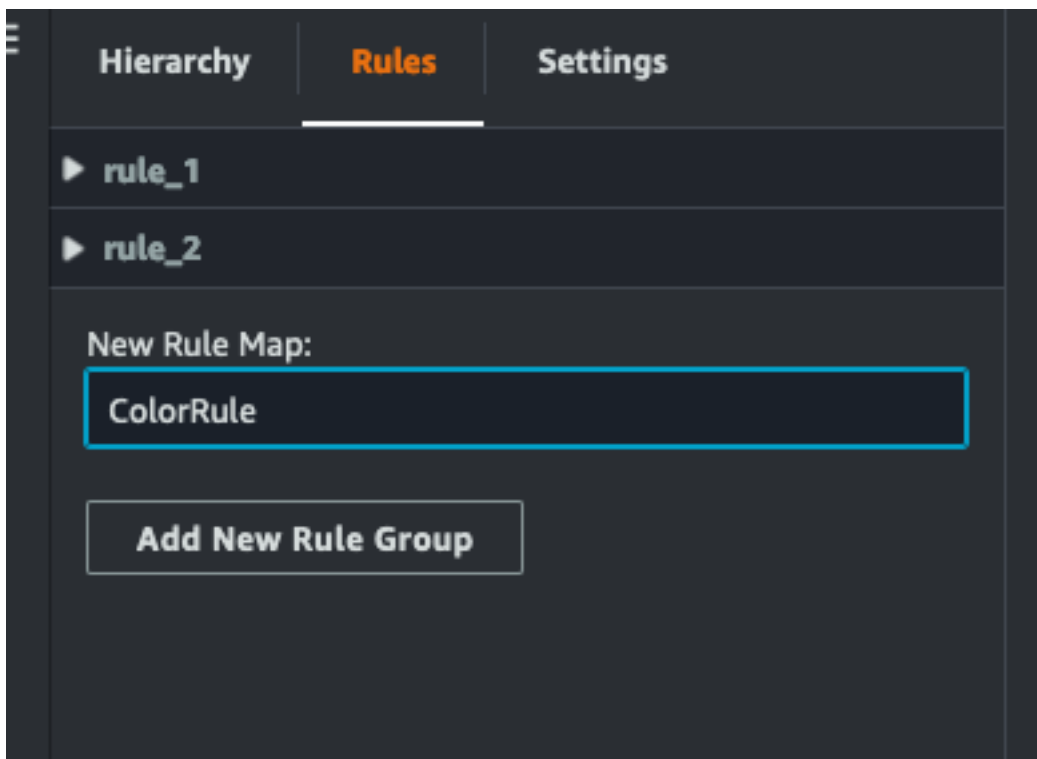
▶ sampleTimeSeriesColorRule

Rule Id

L'immagine sopra mostra una regola che consente di confrontare una proprietà di dati precedentemente definita con ID «temperatura» rispetto a un determinato valore. Ad esempio, se la «temperatura» è maggiore o uguale a 40, lo stato cambierà l'aspetto del tag in un cerchio rosso. La destinazione, se scelta nella dashboard Grafana, popola un pannello di dettaglio configurato per utilizzare la stessa origine dati.

La procedura seguente mostra come aggiungere un nuovo gruppo di regole visive per il livello dell'interfaccia utente con colorazione aumentata della mesh.

1. Nella scheda regole della console, inserisci un nome, ad esempio ColorRule nel campo di testo, e scegli Aggiungi nuovo gruppo di regole.



2. Definisci una nuova regola per il tuo caso d'uso. Ad esempio, puoi crearne uno in base alla proprietà dei dati «temperatura», in cui il valore riportato è inferiore a 20. Utilizzate la seguente sintassi per le espressioni delle regole: Minore di è <, maggiore di è >, minore o uguale è <=, maggiore o uguale è >= e uguale è ==. (Per ulteriori informazioni, consulta la sintassi JEXL di [Apache Commons](#).)
3. Imposta il bersaglio su un colore. Per definire un colore, ad esempio #fcb03, usa valori esadecimali. (Per ulteriori informazioni sui valori esadecimali, consultate [Hexadecimale](#).)

## Creazione di tag per le tue scene

Un tag è un'annotazione aggiunta a una posizione di  $x, y, z$  coordinate specifica di una scena. Il tag utilizza una proprietà di entità per collegare una parte della scena al knowledge graph. È possibile utilizzare un tag per configurare il comportamento o l'aspetto visivo di un elemento nella scena, ad esempio un allarme.

### Note

Per aggiungere funzionalità ai tag, applichi loro delle regole visive.

Utilizzate la seguente procedura per aggiungere tag alla scena.

1. Selezionate un oggetto nella gerarchia, scegliete il pulsante +, quindi scegliete Aggiungi tag.
2. Assegna un nome al tag. Quindi, per applicare una regola visiva, seleziona un ID di gruppo visivo.
3. Negli elenchi a discesa, scegli EntityID e ComponentName. PropertyName
4. Per compilare il campo Percorso dati, scegli Crea. DataFrameLabel

## Formato del modello 3D Tiles

### Usare le tessere 3D nella scena

Se si verificano lunghi tempi di attesa quando si caricano scene 3D AWS IoT TwinMaker o le prestazioni di rendering sono scadenti quando si naviga in un modello 3D complesso, è consigliabile convertire i modelli in riquadri 3D. Questa sezione descrive il formato delle tessere 3D e gli strumenti di terze parti disponibili. Continua a leggere per decidere se le piastrelle 3D sono adatte al tuo caso d'uso e per aiutarti a iniziare.

### Caso d'uso di modelli complessi

Un modello 3D nella AWS IoT TwinMaker scena può causare problemi di prestazioni come tempi di caricamento lenti e ritardo nella navigazione se il modello è:

- Grande: la dimensione del file è superiore a 100 MB.
- Denso: è composto da centinaia o migliaia di mesh distinte.

- **Complessa:** la geometria delle maglie ha milioni di triangoli per formare forme complesse.

## Formato di tessere 3D

[Il formato 3D Tiles](#) è una soluzione per lo streaming della geometria del modello e il miglioramento delle prestazioni di rendering 3D. Consente il caricamento istantaneo di modelli 3D in una AWS IoT TwinMaker scena e ottimizza le interazioni 3D caricando blocchi di un modello in base a ciò che è visibile nella vista della telecamera.

[Il formato 3D Tiles è stato creato da Cesium](#). Cesium dispone di un servizio gestito per convertire modelli 3D in piastrelle 3D chiamato [Cesium Ion](#). Questa è attualmente la soluzione migliore per creare piastrelle 3D e la consigliamo per i modelli complessi nei formati [supportati](#). Puoi registrare Cesium e scegliere il piano di abbonamento appropriato in base alle tue esigenze aziendali nella pagina [dei prezzi di Cesium](#).

Per preparare un modello 3D Tiles da aggiungere a una AWS IoT TwinMaker scena, segui le istruzioni documentate da Cesium Ion:

- [Importa un modello in Cesium Ion](#)

## Carica le tessere Cesium 3D su AWS

Una volta convertito il modello in riquadri 3D, scarica i file del modello e caricali nel bucket Amazon S3 del tuo AWS IoT TwinMaker spazio di lavoro:

1. [Crea e scarica il tuo archivio di modelli 3D Tiles](#).
2. Decomprimi l'archivio in una cartella.
3. Carica l'intera cartella 3D Tiles nel bucket Amazon S3 associato al tuo spazio di lavoro. AWS IoT TwinMaker (Vedi [Caricamento di oggetti](#) nella Guida per l'utente di Amazon S3.)
4. Se il tuo modello 3D Tiles è stato caricato correttamente, vedrai un percorso della cartella Amazon S3 nella tua [libreria di AWS IoT TwinMaker risorse](#) con il tipo. Tiles3D

### Note

La AWS IoT TwinMaker Resource Library non supporta il caricamento diretto di modelli 3D Tiles.

## Utilizzo di riquadri 3D in AWS IoT TwinMaker

AWS IoT TwinMaker è a conoscenza di qualsiasi modello 3D Tiles caricato nel bucket S3 del tuo workspace. Il modello deve avere un file `tileset.json` e tutti i file dipendenti (`.gltf`, `.b3dm`, `.i3dm`, `.cmpt`, `.pntr`) disponibili nella stessa directory Amazon S3. Il percorso della directory Amazon S3 verrà visualizzato nella Resource Library con il tipo `Tiles3D`.

Per aggiungere il modello 3D Tiles alla scena, segui questi passaggi:

1. Nella pagina del compositore di scene, scegli il segno più (+), quindi scegli Aggiungi modello 3D.
2. Nella finestra Aggiungi risorsa dalla libreria di risorse, scegli il percorso del modello 3D Tiles con il tipo `Tiles3D`, quindi scegli Aggiungi.
3. Fai clic sull'area di disegno per posizionare il modello nella scena.

### Differenze tra le tessere 3D

Attualmente 3D Tiles non supporta i metadati geometrici e semantici, il che significa che la gerarchia delle mesh del modello originale non è disponibile per la funzione di selezione dei sottomodelli. È ancora possibile aggiungere widget al modello 3D Tiles, ma non è possibile utilizzare funzionalità ottimizzate per i sottomodelli: shader del modello, trasformazioni 3D separate o associazione di entità per una mesh di sottomodelli.

Si consiglia di utilizzare la conversione 3D Tiles per risorse di grandi dimensioni che fungono da contesto per lo sfondo di una scena. Se desiderate che un sottomodello venga ulteriormente suddiviso e annotato, dovrete estrarlo come risorsa `glTF/GLB` separata e aggiunto direttamente alla scena. [Questo può essere fatto con strumenti 3D gratuiti e comuni come Blender.](#)

Caso d'uso di esempio:

- Hai un modello di fabbrica da 1 GB con sale macchine e pavimenti dettagliati, scatole elettriche e tubi idraulici. Le scatole e le tubazioni elettriche devono diventare rosse quando i dati di proprietà associati superano una soglia.
- Si isolano le maglie della scatola e delle tubazioni nel modello e le si esporta in un `glTF` separato utilizzando Blender.
- Converti la fabbrica senza elementi elettrici e idraulici in un modello 3D Tiles e lo carichi su S3.
- Aggiungi sia il modello 3D Tiles che il modello `glTF` a una AWS IoT TwinMaker scena all'origine (0,0,0).



- Aggiungete i componenti del model shader alla scatola elettrica e ai sottomodelli pipe di gITF per rendere rosse le mesh in base alle regole di proprietà.

# Crea un'applicazione web personalizzata utilizzando AWS IoT TwinMaker i componenti dell'interfaccia utente

AWS IoT TwinMaker fornisce componenti dell'interfaccia utente open source per gli sviluppatori di AWS IoT applicazioni. Utilizzando questi componenti dell'interfaccia utente, gli sviluppatori possono creare applicazioni Web personalizzate con AWS IoT TwinMaker funzionalità abilitate per i loro gemelli digitali.

AWS IoT TwinMaker I componenti dell'interfaccia utente fanno parte dell' AWS IoT Application Kit, una libreria open source lato client che consente agli sviluppatori di applicazioni IoT di semplificare lo sviluppo di applicazioni IoT complesse

AWS IoT TwinMaker I componenti dell'interfaccia utente includono:

- AWS IoT TwinMaker fonte:

Un componente del connettore dati che consente di recuperare dati e interagire con i AWS IoT TwinMaker dati e i gemelli digitali.

Per ulteriori informazioni, consulta la documentazione di [AWS IoT TwinMaker origine](#).

- Visualizzatore di scene:

Un componente di rendering 3D integrato @react-three/fiber che esegue il rendering del gemello digitale e consente di interagire con esso.

Per ulteriori informazioni, consultate la documentazione di [Scene Viewer](#).

- Lettore video:

Un componente del lettore video che consente di trasmettere in streaming un video da Kinesis Video AWS IoT TwinMaker Streams a.

Per ulteriori informazioni, consulta la documentazione di [Video Player](#).

Per saperne di più sull'utilizzo di AWS IoT Application Kit, visita la pagina [Github di AWS IoT Application Kit](#).

Per istruzioni su come avviare una nuova applicazione Web utilizzando AWS IoT Application Kit, visita la pagina della documentazione ufficiale [dell'IoT App Kit](#).

# Cambia modalità di AWS IoT TwinMaker prezzo

AWS IoT TwinMaker attualmente offre tre modalità di prezzo, pacchetto base, standard o a più livelli. La modalità di prezzo standard è impostata come modalità di determinazione del prezzo predefinita.

Puoi passare dalla modalità di tariffazione basata sull'utilizzo a quella basata su più livelli in qualsiasi momento, ma la modifica avrà effetto all'inizio del ciclo di fatturazione successivo. Dopo essere passati dalla modalità di determinazione dei prezzi basata sull'utilizzo a quella basata su livelli, non è possibile tornare alla modalità di determinazione dei prezzi basata sull'utilizzo per i tre cicli di utilizzo successivi. Se si passa dalla modalità base a quella standard, la modifica ha effetto immediato. Per dettagli e informazioni sui costi, consulta la sezione [AWS IoT TwinMaker Prezzi](#)

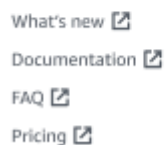
Questa procedura mostra come cambiare la modalità di prezzo nella [AWS IoT TwinMaker console](#):





1. Apri la [AWS IoT TwinMaker console](#).
2. Nel riquadro di navigazione a sinistra, seleziona Impostazioni. Viene visualizzata la pagina dei prezzi.



How it works  
Workspaces  
Workspace  
Component types  
Entities  
Resource library  
Scenes

## Settings



What's new   
Documentation   
FAQ   
Pricing 

3. Scegli la modalità Modifica prezzo.
4. Seleziona la modalità bundle Standard o Tiered, come mostrato nella schermata seguente.

### 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**

- Scegli Salva per confermare la nuova modalità di prezzo.
- Ora hai cambiato la modalità di prezzo.

#### Note

Puoi passare dalla modalità di tariffazione basata sull'utilizzo a quella basata su più livelli in qualsiasi momento, ma la modifica avrà effetto all'inizio del ciclo di fatturazione successivo. Dopo essere passati dalla modalità di determinazione dei prezzi basata sull'utilizzo a quella basata su livelli, non è possibile tornare alla modalità di determinazione dei prezzi basata sull'utilizzo per i tre cicli di utilizzo successivi. Se si passa dalla modalità base a quella standard, la modifica ha effetto immediato.

# AWS IoT TwinMaker grafico della conoscenza

Il AWS IoT TwinMaker knowledge graph organizza tutte le informazioni contenute nei tuoi AWS IoT TwinMaker spazi di lavoro e le presenta in un formato grafico visivo. È possibile eseguire interrogazioni sulle entità, sui componenti e sui tipi di componenti per generare grafici visivi che mostrano le relazioni tra le risorse. AWS IoT TwinMaker

I seguenti argomenti mostrano come utilizzare e integrare il Knowledge Graph.

## Argomenti

- [AWS IoT TwinMaker concetti fondamentali del Knowledge Graph](#)
- [Come eseguire le interrogazioni del AWS IoT TwinMaker Knowledge Graph](#)
- [Integrazione con le scene del Knowledge](#)
- [Come usare il AWS IoT TwinMaker Knowledge Graph con Grafana](#)
- [AWS IoT TwinMaker knowledge graph: risorse aggiuntive](#)

## AWS IoT TwinMaker concetti fondamentali del Knowledge Graph

Questo argomento tratta i concetti chiave e il vocabolario della funzionalità Knowledge Graph.

Come funziona il Knowledge Graph:

Il Knowledge Graph crea relazioni tra le entità e i loro componenti con le [UpdateEntity](#)API [CreateEntity](#)o le esistenti. Una relazione è solo una proprietà di un tipo di dati speciale [RELATIONSHIP](#) definito su un componente di un'entità. AWS IoT TwinMaker knowledge graph chiama l'[ExecuteQuery](#)API per effettuare una query basata su qualsiasi dato nelle entità o sulle relazioni tra di esse. Knowledge Graph utilizza il linguaggio di query PartiQL flessibile (utilizzato da molti AWS servizi) che ha appena aggiunto il supporto per la sintassi Graph Match per aiutarti a scrivere le tue query. Dopo aver effettuato le chiamate, è possibile visualizzare i risultati sotto forma di tabella o visualizzarli come un grafico dei nodi e degli spigoli connessi.

Termini chiave del Knowledge Graph:

- Grafo delle entità: una raccolta di nodi e bordi all'interno di uno spazio di lavoro.
- Nodo: ogni entità nell'area di lavoro diventa un nodo nel grafico delle entità.
- Edge: ogni proprietà di relazione definita su un componente di un'entità diventa un bordo nel grafico dell'entità. Inoltre, una relazione gerarchica padre-figlio definita utilizzando il

parentEntityId campo di un'entità diventa anche un bordo nel grafico dell'entità con un nome di relazione "»isChildOf. Tutti gli spigoli sono bordi direzionali.

- **Relazione:** una AWS IoT TwinMaker relazione è un tipo speciale di proprietà del componente di un'entità. È possibile utilizzare l'[UpdateEntity](#) API AWS IoT TwinMaker [CreateEntity](#) per definire e modificare una relazione. Nel AWS IoT TwinMaker, una relazione deve essere definita in un componente di un'entità. Una relazione non può essere definita come una risorsa isolata. Una relazione deve essere direzionale da un'entità all'altra.

## Come eseguire le interrogazioni del AWS IoT TwinMaker Knowledge Graph

Prima di utilizzare il AWS IoT TwinMaker knowledge graph, assicurati di aver completato i seguenti prerequisiti:

- Crea uno spazio di AWS IoT TwinMaker lavoro. [È possibile creare un'area di lavoro nella AWS IoT TwinMaker console.](#)
- Acquisisci familiarità con AWS IoT TwinMaker il sistema entità-componente e con come creare entità. Per ulteriori informazioni, consulta [Crea la tua prima entità.](#)
- Acquisisci familiarità con i AWS IoT TwinMaker connettori dati. Per ulteriori informazioni, consulta [AWS IoT TwinMaker connettori dati.](#)

### Note

Per utilizzare il AWS IoT TwinMaker Knowledge Graph, è necessario utilizzare la modalità di prezzo in bundle standard o a più livelli. Per ulteriori informazioni, consulta [Cambia modalità di AWS IoT TwinMaker prezzo.](#)

Le procedure seguenti mostrano come scrivere, eseguire, salvare e modificare le interrogazioni.

Aprire l'editor delle interrogazioni

Per accedere all'editor di query del Knowledge Graph

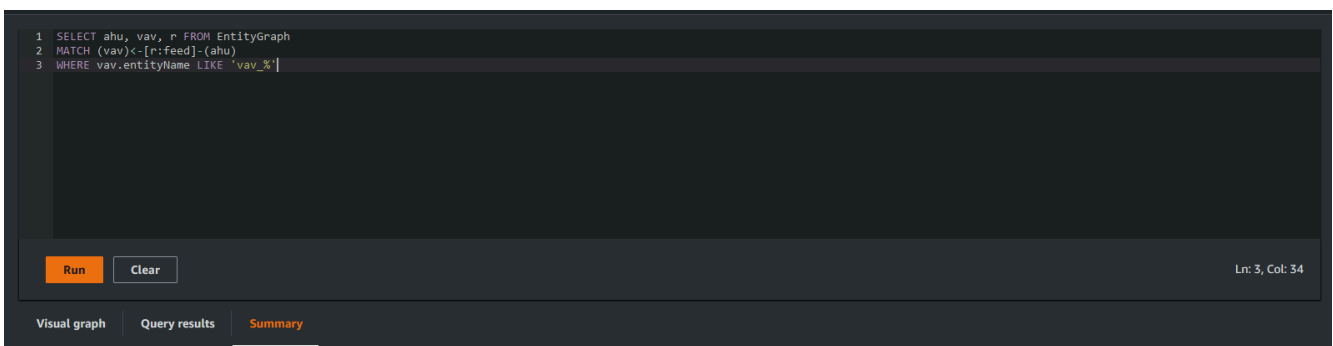
1. Apri la [AWS IoT TwinMaker console.](#)
2. Apri l'area di lavoro in cui desideri utilizzare Knowledge Graph.

3. Nel menu di navigazione a sinistra, scegli Query editor.
4. Si apre l'editor di interrogazioni. Ora sei pronto per eseguire interrogazioni sulle risorse del tuo spazio di lavoro.

## Esegui una query

Per eseguire un'interrogazione e generare un grafico

1. Nell'editor di query, scegliete la scheda Editor per aprire l'editor di sintassi.
2. Nello spazio dell'editor, scrivi la query che desideri eseguire sulle risorse del tuo spazio di lavoro.



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

Nell'esempio mostrato, la richiesta cerca le entità che contengono vav\_% nel loro nome, quindi le organizza in base alla feed relazione tra loro, utilizzando il codice seguente.

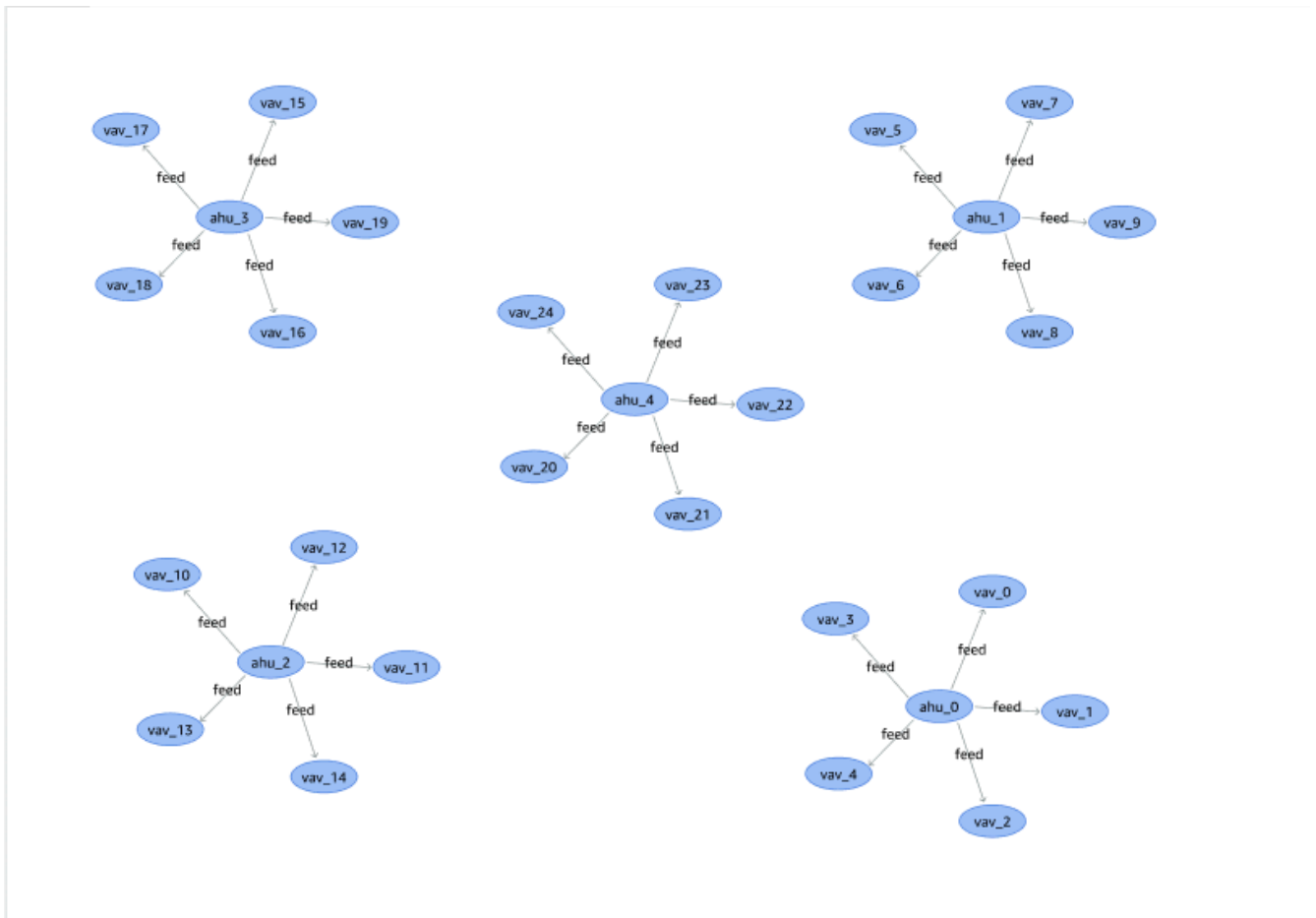
```
SELECT ahu, vav, r FROM EntityGraph
MATCH (vav)<-[r:feed]-(ahu)
WHERE vav.entityName LIKE 'vav_%'
```

### Note

La sintassi del knowledge graph utilizza [PartiQL](#). Per informazioni su questa sintassi, vedere. [AWS IoT TwinMaker knowledge graph: risorse aggiuntive](#)

3. Scegli Esegui query per eseguire la richiesta che hai creato.

Viene generato un grafico in base alla tua richiesta.



Il grafico di esempio mostrato sopra si basa sull'esempio di interrogazione riportato nel passaggio 2.

4. I risultati dell'interrogazione sono inoltre presentati in un elenco. Scegliete i risultati per visualizzare i risultati dell'interrogazione in un elenco.
5. Facoltativamente, scegli Esporta come per esportare i risultati della query in formato JSON o CSV.

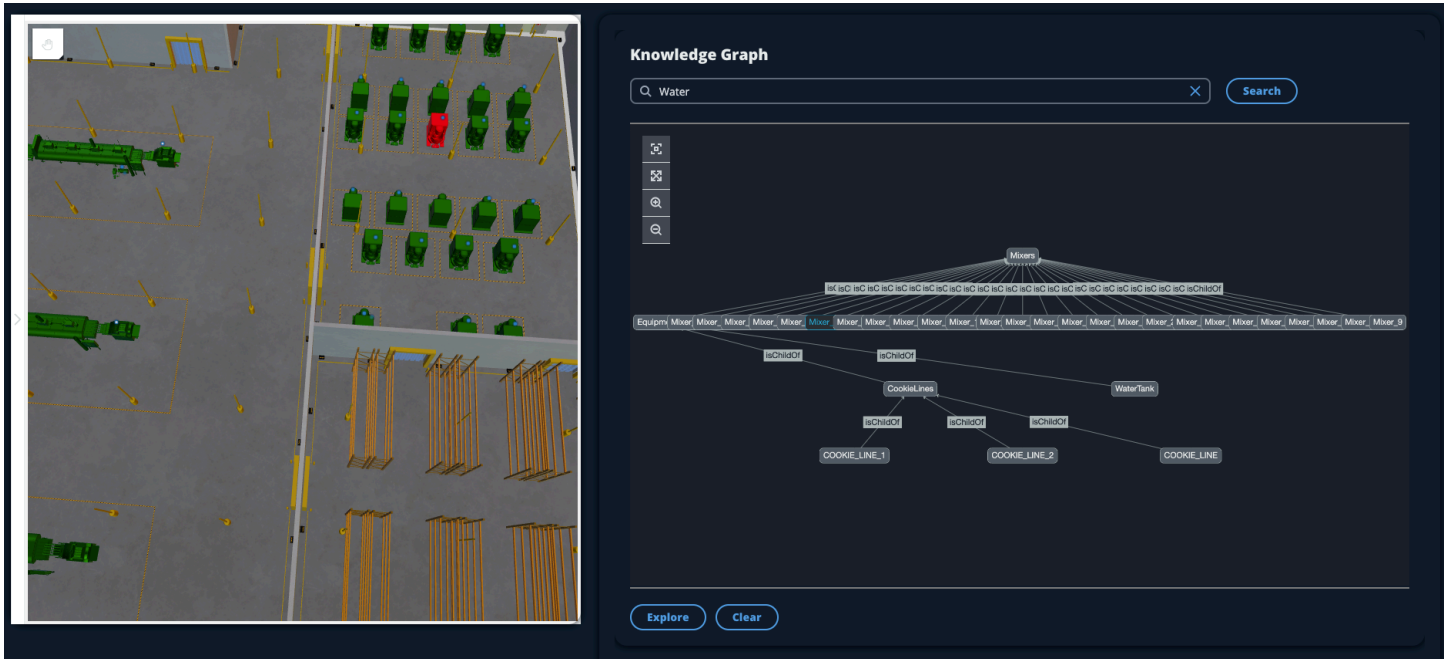
Questo riguarda l'uso di base del knowledge graph nella console. Per ulteriori informazioni ed esempi che dimostrano la sintassi del knowledge graph, vedere. [AWS IoT TwinMaker knowledge graph: risorse aggiuntive](#)

## Integrazione con le scene del Knowledge

Puoi utilizzare i componenti AWS IoT dell'app kit per creare un'applicazione web che integri Knowledge Graph nelle tue AWS IoT TwinMaker scene. Ciò consente di generare grafici basati sui



nodì 3D (i modelli 3D che rappresentano le apparecchiature o i sistemi) presenti all'interno della scena. Per creare un'applicazione che rappresenti graficamente i nodi 3D della scena, associa innanzitutto i nodi 3D alle entità presenti nell'area di lavoro. Con questa mappatura, rappresenta AWS IoT TwinMaker graficamente le relazioni tra i modelli 3D presenti nella scena e le entità nell'area di lavoro. Quindi puoi creare un'applicazione web, selezionare modelli 3D con la tua scena ed esplorare le loro relazioni con altre entità in un formato grafico.



Per un esempio di applicazione web funzionante che utilizza i componenti dell' AWS IoT app kit per generare grafici in una AWS IoT TwinMaker scena, guarda l'[app React di AWS IoT TwinMaker esempio](#) su github.

## AWS IoT TwinMaker prerequisiti del grafico della scena

Prima di creare un'app Web che utilizzi AWS IoT TwinMaker Knowledge Graph nelle tue scene, completa i seguenti prerequisiti:

- Crea uno spazio di AWS IoT TwinMaker lavoro. [È possibile creare un'area di lavoro nella AWS IoT TwinMaker console.](#)
- Acquisisci familiarità con AWS IoT TwinMaker il sistema entità-componente e con come creare entità. Per ulteriori informazioni, consulta [Crea la tua prima entità.](#)
- Crea una AWS IoT TwinMaker scena popolata con modelli 3D.

- Acquisisci familiarità con AWS IoT TwinMaker i componenti dell' AWS IoT app kit. Per ulteriori informazioni sui AWS IoT TwinMaker componenti, consulta [Crea un'applicazione web personalizzata utilizzando AWS IoT TwinMaker i componenti dell'interfaccia utente](#).
- Acquisisci dimestichezza con i concetti e la terminologia chiave del Knowledge Graph. Per informazioni, consulta [AWS IoT TwinMaker concetti fondamentali del Knowledge Graph](#).

#### Note

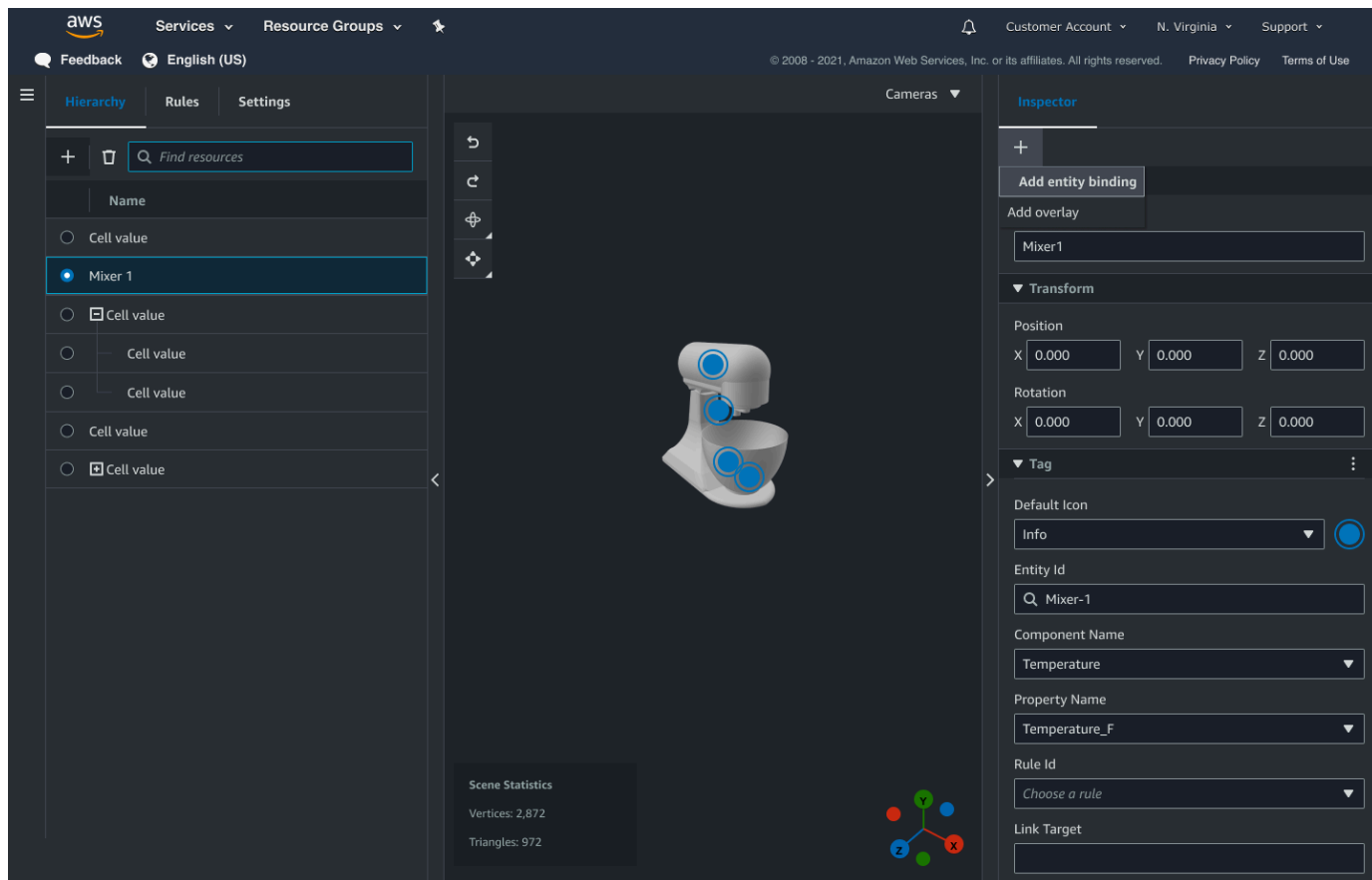
Per utilizzare il AWS IoT TwinMaker Knowledge Graph e tutte le funzionalità correlate, è necessario utilizzare la modalità di prezzo dei pacchetti standard o a più livelli. Per ulteriori informazioni sui AWS IoT TwinMaker prezzi, consulta. [Cambia modalità di AWS IoT TwinMaker prezzo](#)

## Associa i nodi 3D alla tua scena

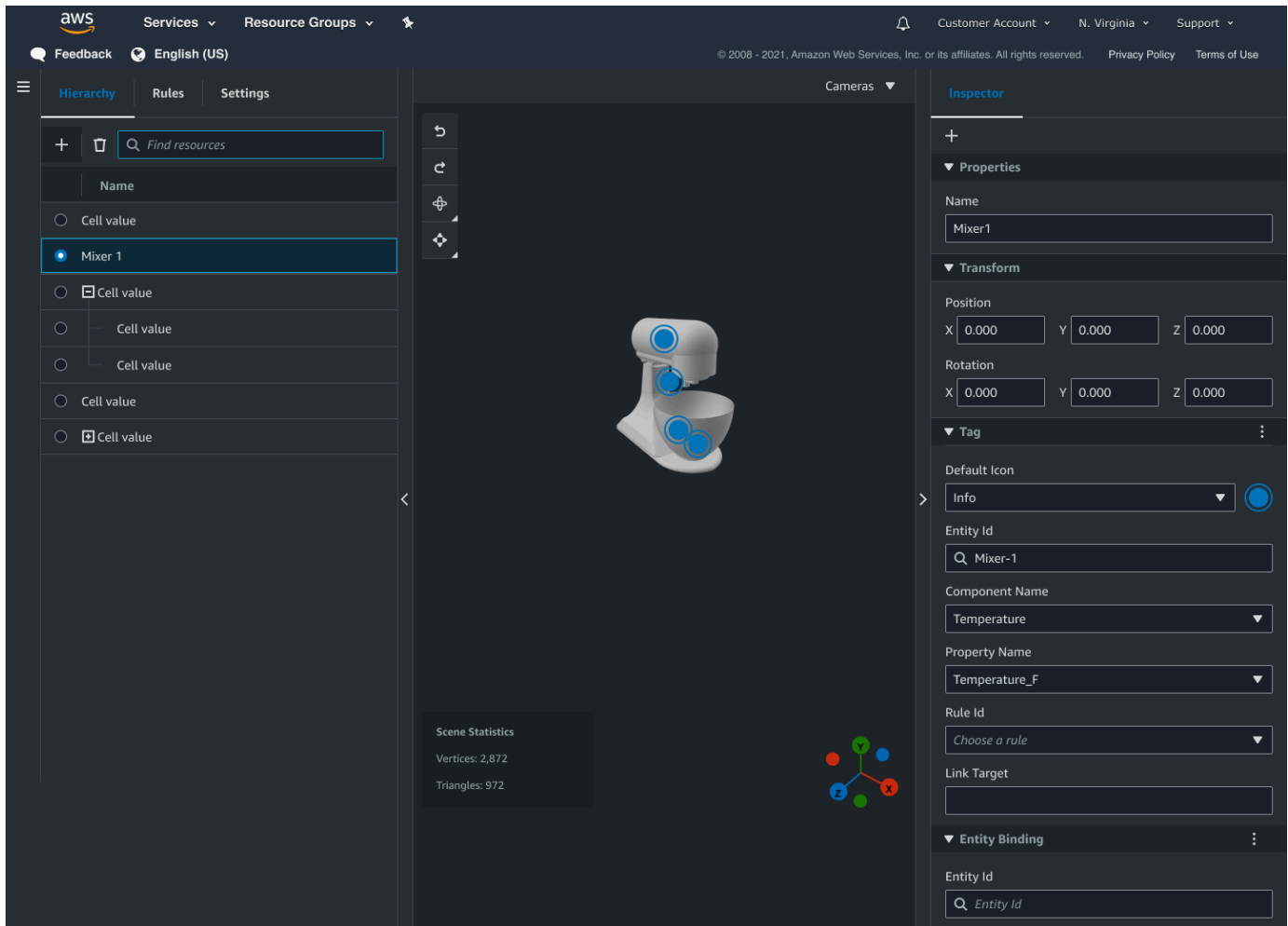
Prima di creare un'app Web che integri Knowledge Graph con la scena, associa i modelli 3D, denominati nodi 3D, presenti nella scena all'entità dell'area di lavoro associata. Ad esempio, se avete un modello di attrezzatura mixer in una scena e un'entità corrispondente denominata `mixer_0`, create un'associazione dati tra il modello del mixer e l'entità che rappresenta il mixer, in modo che il modello e l'entità possano essere rappresentati graficamente.

Per eseguire un'azione di associazione dei dati

1. Accedi alla console di [AWS IoT TwinMaker](#).
2. Apri il tuo spazio di lavoro e seleziona una scena con i nodi 3D che desideri associare.
3. Seleziona un nodo (modello 3D) nel compositore di scene. Quando selezionate un nodo, si aprirà un pannello di ispezione sul lato destro dello schermo.
4. Nel pannello di ispezione, accedi alla parte superiore del pannello e seleziona il pulsante +. Quindi scegli l'opzione Aggiungi associazione di entità. Si aprirà un menu a discesa in cui puoi selezionare un'entità da associare al nodo attualmente selezionato.



5. Dal menu a discesa dell'associazione dati, selezionate l'ID dell'entità che desiderate mappare al modello 3D. Per i campi Nome componente e Nome proprietà, seleziona i componenti e le proprietà che desideri associare.



Dopo aver effettuato le selezioni per i campi Entity Id, Component Name e Property Name, l'associazione è completa.

- Ripetete questo processo per tutti i modelli e le entità che desiderate rappresentare graficamente.

#### Note

La stessa operazione di associazione dati può essere eseguita sui tag della scena, è sufficiente selezionare un tag anziché un'entità e seguire la stessa procedura per associare il tag a un nodo.

## Crea un'applicazione web

Dopo aver associato le entità, utilizza la libreria dell' AWS IoT app kit per creare un'app Web con un widget Knowledge Graph che ti permetta di visualizzare la scena ed esplorare le relazioni tra i nodi e le entità della scena.

Utilizza le seguenti risorse per creare la tua app:

- La documentazione Readme di AWS IoT TwinMaker esempio su github dell'app [React](#).
- Il [sorgente](#) dell'app React AWS IoT TwinMaker di esempio su github.
- Documentazione [introduttiva del kit di AWS IoT](#) app.
- La documentazione del [componente Video Player](#) dell' AWS IoT app kit.
- La documentazione del [componente Scene Viewer dell' AWS IoT](#) app kit.

La procedura seguente dimostra la funzionalità del componente Scene Viewer in un'app Web.

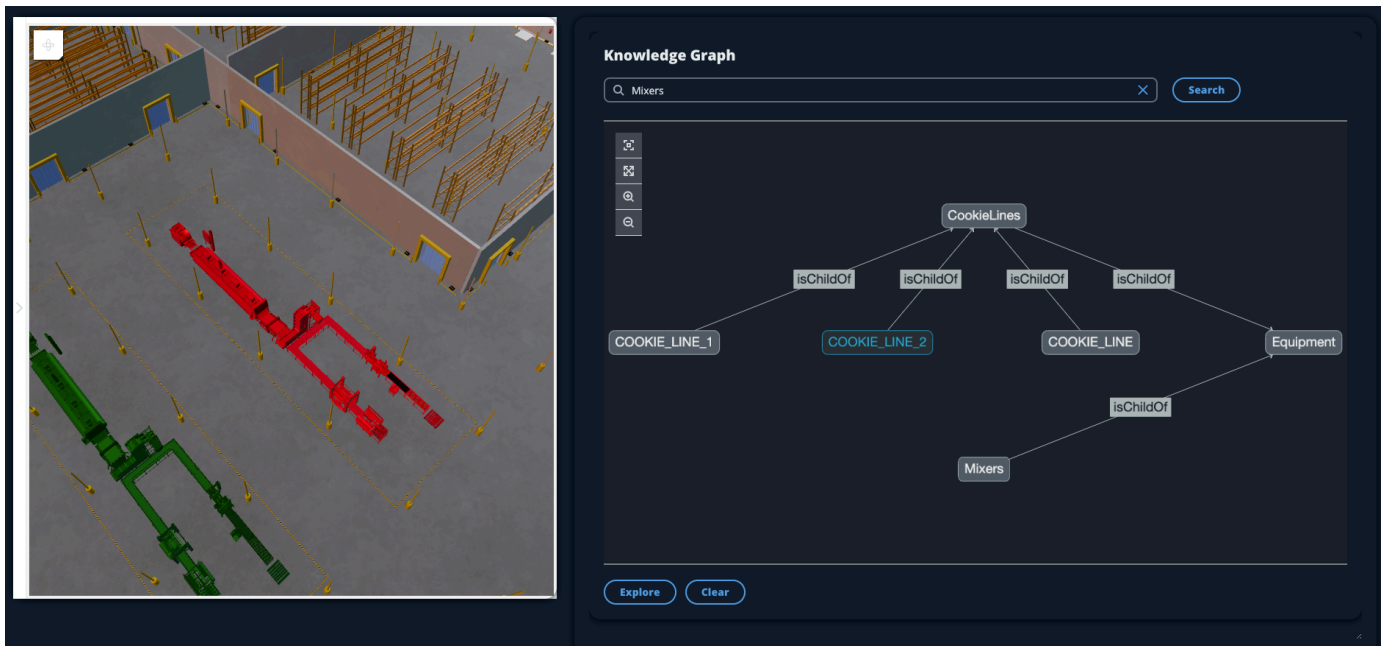
### Note

Questa procedura si basa sull'implementazione del componente AWS IoT app kit scene viewer nell'app React AWS IoT TwinMaker di esempio.

1. Apri il componente scene viewer dell'app AWS IoT TwinMaker Sample React. Nel campo di ricerca digita un nome di entità o un nome di entità parziale (ricerca con distinzione tra maiuscole e minuscole), quindi seleziona il pulsante Cerca. Se un modello è associato all'ID dell'entità, il modello nella scena verrà evidenziato e un nodo dell'entità verrà mostrato nel pannello di visualizzazione della scena.



- Per generare un grafico di tutte le relazioni, selezionate un nodo nel widget Scene Viewer e selezionate il pulsante Esplora.



- Preми il pulsante Cancella per cancellare la selezione attuale del grafico e ricominciare da capo.

# Come usare il AWS IoT TwinMaker Knowledge Graph con Grafana

Questa sezione mostra come aggiungere un pannello di editor di query alla dashboard di AWS IoT TwinMaker Grafana per eseguire e visualizzare le query.

## AWS IoT TwinMaker prerequisiti dell'editor di query

Prima di utilizzare il AWS IoT TwinMaker Knowledge Graph in Grafana, completa i seguenti prerequisiti:

- Crea uno spazio di lavoro AWS IoT TwinMaker . [È possibile creare un'area di lavoro nella AWS IoT TwinMaker console.](#)
- Configura AWS IoT TwinMaker per l'uso con Grafana. Per istruzioni, consulta [AWS IoT TwinMakerIntegrazione della dashboard Grafana.](#)

### Note

Per utilizzare il AWS IoT TwinMaker Knowledge Graph, è necessario utilizzare la modalità di prezzo in bundle standard o a più livelli. Per ulteriori informazioni, consulta [Cambia modalità di AWS IoT TwinMaker prezzo.](#)

## AWS IoT TwinMaker autorizzazioni dell'editor di query

Per utilizzare l'editor di AWS IoT TwinMaker query in Grafana, devi disporre di un ruolo IAM con autorizzazione per l'azione. `iotttwinmaker:ExecuteQuery` Aggiungi tale autorizzazione al ruolo del pannello di controllo del tuo spazio di lavoro, come mostrato in questo esempio:

```
{
  "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

Quando configuri la tua origine dati AWS IoT TwinMaker Grafana, assicurati di utilizzare il ruolo con questa autorizzazione per il campo ARN Assumi ruolo. Dopo averlo aggiunto, puoi selezionare il tuo spazio di lavoro dal menu a discesa accanto a Workspace.

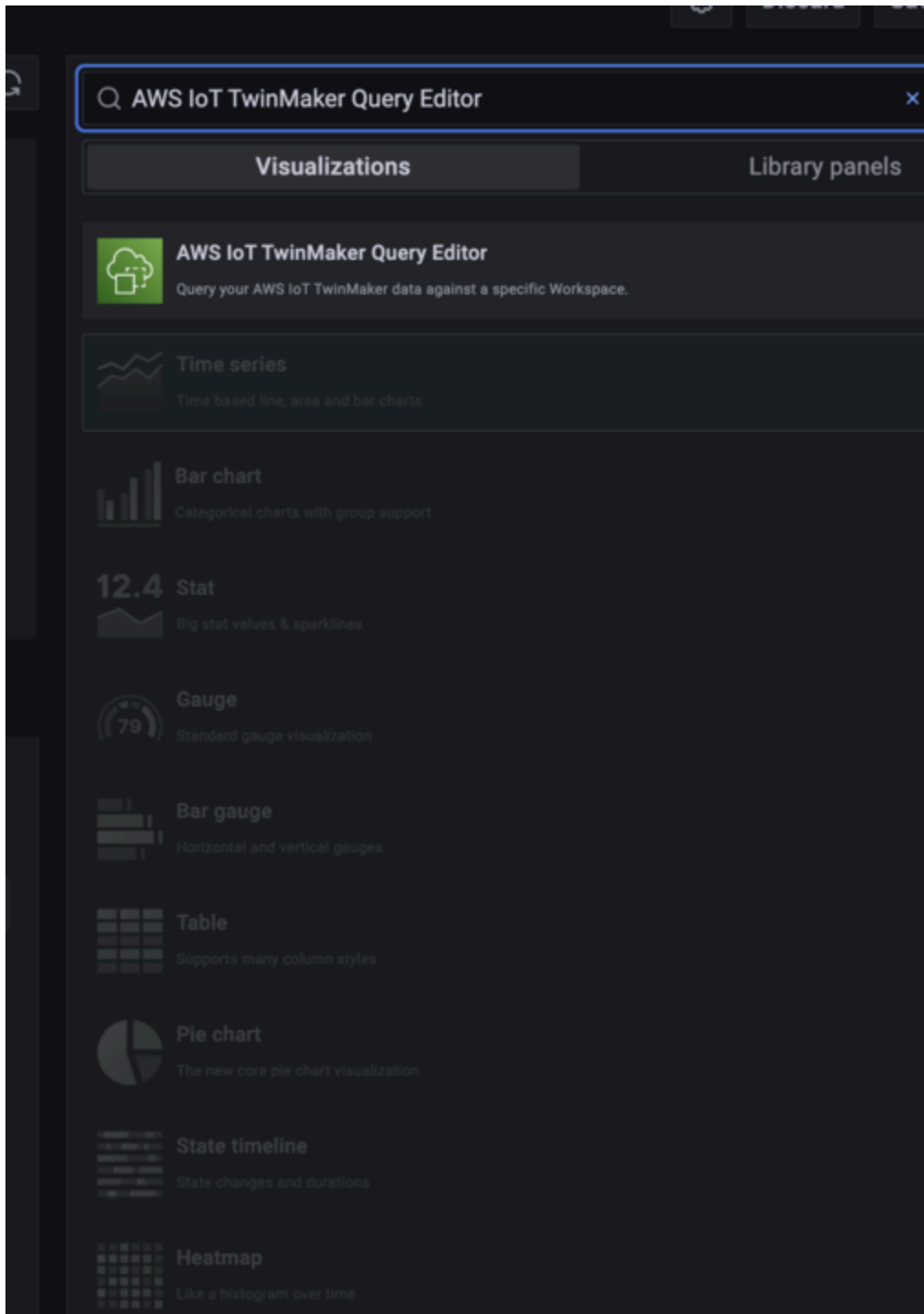
Per ulteriori informazioni, consulta [Creazione di un ruolo IAM nella dashboard](#).

## Configura il pannello dell'editor delle AWS IoT TwinMaker interrogazioni

Per configurare un nuovo pannello dashboard Grafana per Knowledge Graph

1. Apri la dashboard AWS IoT TwinMaker Grafana.
2. Crea un nuovo pannello di controllo. Per i passaggi dettagliati su come creare un pannello, consulta [Creare un pannello di controllo](#) nella documentazione di Grafana.
3. Dall'elenco delle visualizzazioni, seleziona AWS IoT TwinMaker Query Editor.





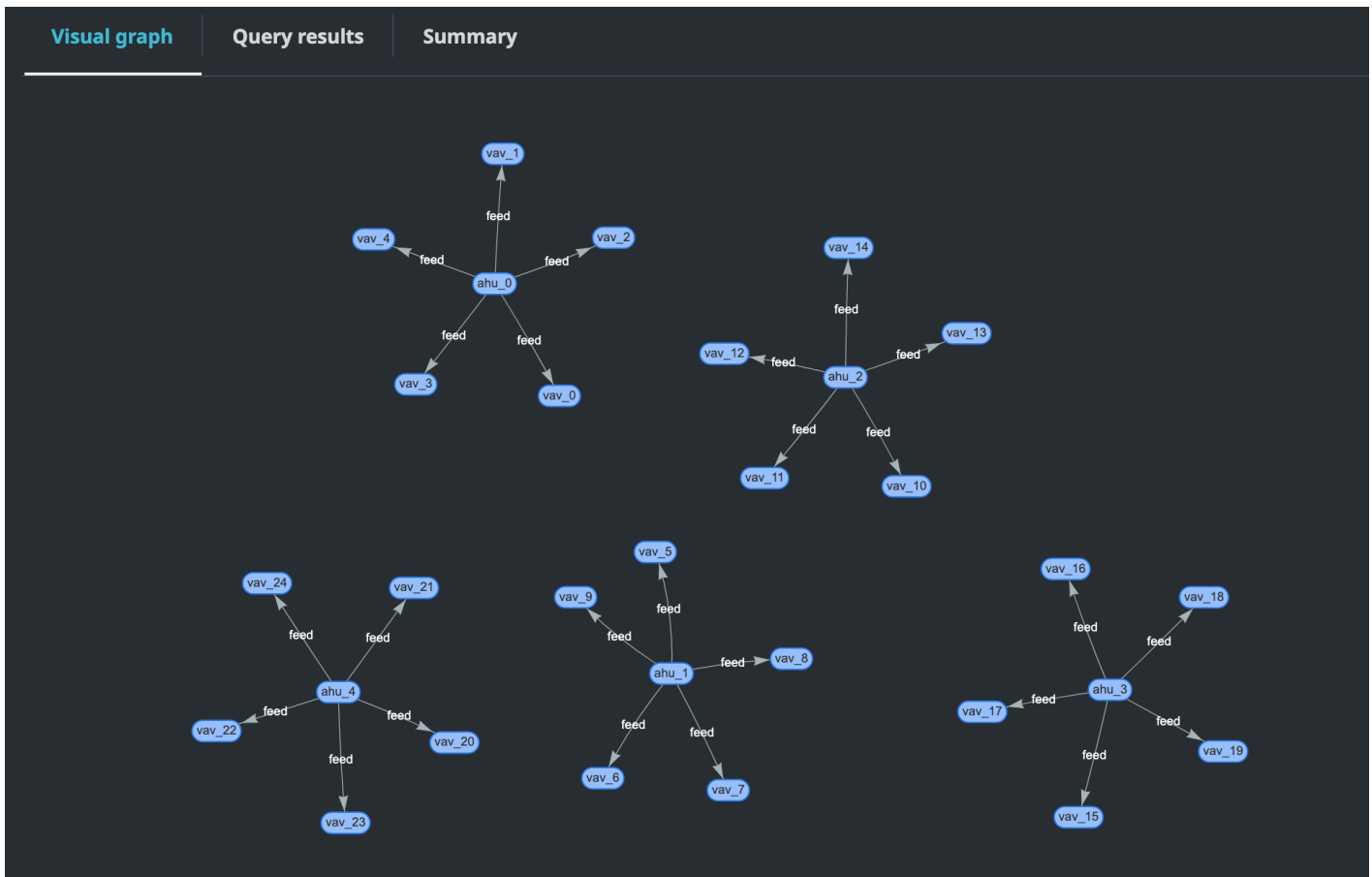
4. Seleziona la fonte di dati su cui eseguire le query.
5. (Facoltativo) Aggiungi un nome per il nuovo pannello nel campo fornito.
6. Seleziona Applica per salvare e confermare il nuovo pannello.

Il pannello Knowledge Graph funziona in modo simile all'editor di query fornito nella AWS IoT TwinMaker console. Puoi eseguire, scrivere e cancellare le interrogazioni che fai nel pannello. Per ulteriori informazioni su come scrivere interrogazioni, consulta. [AWS IoT TwinMaker knowledge graph: risorse aggiuntive](#)

Come usare l'editor di AWS IoT TwinMaker query

I risultati delle interrogazioni vengono visualizzati in tre modi, come illustrato nelle immagini seguenti: visualizzati in un grafico, elencati in una tabella o presentati come riepilogo dell'esecuzione.

- Visualizzazione del grafico:



Il grafico visivo mostra solo i dati per le interrogazioni che hanno almeno una relazione nel risultato. Il grafico mostra le entità come nodi e le relazioni come bordi diretti nel grafico.

- Dati tabulari:

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>

Il formato di dati tabulari visualizza i dati per tutte le interrogazioni. È possibile cercare nella tabella risultati specifici o sottoinsiemi di risultati. I dati possono essere esportati in formato JSON o CSV.

- Riepilogo dell'esecuzione

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

Il riepilogo dell'esecuzione visualizza l'interrogazione e i metadati relativi allo stato dell'interrogazione.

## AWS IoT TwinMaker knowledge graph: risorse aggiuntive

Questa sezione fornisce esempi di base della sintassi PartiQL utilizzata per scrivere query nel knowledge graph, nonché collegamenti alla documentazione PartiQL che fornisce informazioni sul modello di dati del knowledge graph.

- [Documentazione del modello di dati a grafo PartiQL](#)
- [Documentazione sulle interrogazioni grafiche PartiQL](#)

Questa serie di esempi mostra le domande di base con le relative risposte. Utilizzatelo come riferimento per scrivere le vostre domande.

### Interrogazioni di base

- Ottieni tutte le entità con un filtro

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

Questa query restituisce tutte le entità in uno spazio di lavoro con il nome `room_0`.

**FROM**clausola: `EntityGraph` è la raccolta di grafici che contiene tutte le entità e le loro relazioni in uno spazio di lavoro. Questa raccolta viene creata e gestita automaticamente in AWS IoT TwinMaker base alle entità presenti nell'area di lavoro.

**MATCH**clausola: specifica un modello che corrisponde a una parte del grafico. In questo caso, il pattern `(entity)` corrisponde a ogni nodo del grafico ed è associato alla variabile di entità. La **FROM** clausola deve essere seguita dalla **MATCH** clausola.

**WHERE**clausola: specifica un filtro sul `entityName` campo del nodo, dove il valore deve corrispondere. `room_0`

**SELECT**clausola: specifica la `entity` variabile in modo che venga restituito l'intero nodo dell'entità.

Risposta:

```
{
  "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` Restituisce i metadati relativi alla colonna, come il nome e il tipo. Il tipo restituito è `NODE`. Ciò indica che l'intero nodo è stato restituito. Altri valori per il tipo possono

essere quelli EDGE che indicherebbero una relazione o VALUE che indicherebbero un valore scalare come un numero intero o una stringa.

rowsRestituisce un elenco di righe. Poiché è stata trovata una sola entità, ne rowData viene restituita una che contiene tutti i campi di un'entità.

#### Note

A differenza di SQL in cui è possibile restituire solo valori scalari, è possibile restituire un oggetto (come JSON) utilizzando PartiQL.

Ogni nodo contiene tutti i campi a livello di entitàentityId, arn ad esempio i campi a livello di componentecomponents, nonché i campi a livello di proprietà properties come componentTypeId ecomponentName, il tutto come JSON annidato. propertyName propertyValue

- Ottieni tutte le relazioni con un filtro:

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

Questa query restituisce tutte le relazioni in un'area di lavoro con il nome isLocationOf della relazione.

La MATCH clausola: specifica un modello che corrisponde a due nodi (indicati da ( )) che sono collegati da uno spigolo diretto (indicato da - [ ] ->) e legati a una variabile chiamata. relationship

La WHERE clausola: specifica un filtro sul relationshipName campo dello spigolo, dove si trova il valore. isLocationOf

La SELECT clausola: specifica la variabile di relazione in modo che venga restituito l'intero nodo perimetrale.

Risposta

```
{
  "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
]
}

```

Il tipo di colonna in `columnDescriptions` è un. `EDGE`

Ciascuno `rowData` rappresenta un bordo con campi simili a `relationshipName`. È lo stesso nome della proprietà di relazione definita nell'entità. `sourceComponentName` e `sourceComponentTypeId` forniscono informazioni sull'entità e sul componente su cui è stata definita la proprietà della relazione. `sourceEntityId` `targetEntityId` specificano a quale entità è rivolta questa relazione.

- Ottieni tutte le entità con una relazione specifica con un'entità specifica

```

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

```

Questa query restituisce tutti i nomi di entità di tutte le entità che hanno una `isLocationOf` relazione con l'`room_0` entità.

La `MATCH` clausola: specifica un modello che corrisponde a due nodi qualsiasi (`e1`, `e2`) con un bordo diretto (`r`).

La `WHERE` clausola: specifica un filtro sul nome della relazione e sul nome dell'entità di origine.

La `SELECT` clausola: restituisce il `entityName` campo nel nodo. `e2`

## Risposta

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

In ColumnDescriptions, il tipo di colonna è VALUE since entityName è una stringa.

Viene restituita un'entità floor\_0,.

## CORRISPONDENZA

I seguenti modelli sono supportati in una MATCH clausola:

- Abbina il nodo 'b' che punta al nodo 'a':

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

- Abbina il nodo 'a' che punta al nodo 'b':

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

Non esiste alcuna variabile associata a una relazione, presupponendo che non sia necessario specificare un filtro sulla relazione.

- Abbina il nodo 'a' che punta al nodo 'b' e il nodo 'b' che punta al nodo 'a':

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

Ciò restituirà due corrispondenze: una da «a» a «b» e l'altra da «b» a «a», quindi la raccomandazione è di utilizzare bordi diretti ove possibile.



- Il nome della relazione è anche un'etichetta del grafico delle proprietà EntityGraph, quindi puoi semplicemente specificare il nome della relazione dopo i due punti (:) invece di specificare un filtro `rel.relationshipName` nella WHERE clausola.

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

- Concatenamento: i modelli possono essere concatenati in modo da corrispondere su più relazioni.

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

- I pattern di hop variabili possono estendersi anche su più nodi e spigoli:

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

Questa query corrisponde a qualsiasi modello con bordi in uscita dal nodo 'a' entro 1-5 passaggi. I quantificatori consentiti sono:

{m, n}- tra m e n ripetizioni

{m, }- m o più ripetizioni.

DA:

Un nodo di entità può contenere dati nidificati, ad esempio componenti che a loro volta contengono ulteriori dati nidificati come le proprietà. È possibile accedervi annullando il risultato del pattern 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'
```

Accedi ai campi annidati inserendo un puntino in una variabile.. Una virgola (,) viene utilizzata per separare (o unire) le entità con i componenti interni e quindi le proprietà all'interno di tali componenti. AS viene utilizzato per associare una variabile alle variabili non annidate in modo che possano essere utilizzate nelle clausole or. WHERE SELECT Questa query restituisce tutte le entità che contengono una proprietà denominata `roomFunction` con valore `meeting` in un componente con tipo di componente id `com.example.query.construction.room`

Per accedere a più campi annidati di un campo, ad esempio a più componenti di un'entità, usa la notazione virgola per eseguire un'unione.

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

#### SELEZIONA:

- Restituisce un nodo:

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

- Restituisce un bordo:

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

- Restituisce un valore scalare:

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

Formatta il nome del campo di output assegnandogli un alias utilizzando AS. Qui, invece che `propertyValue` come nome di colonna nella risposta, `roomfunction` viene restituito.

- Alias di ritorno:

```
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%'
```

Si consiglia vivamente di utilizzare gli alias per essere espliciti, aumentare la leggibilità ed evitare ambiguità nelle query.

#### DOVE:

- Gli operatori logici supportati sono ANDNOT, eOR.
- Gli operatori di confronto supportati sono <<=,>,>=,>=, e !=.

- Utilizzate la IN parola chiave se desiderate specificare più OR condizioni sullo stesso campo.
- Filtra in base a un'entità, un componente o un campo di proprietà:

```
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'
```

- Filtro sulla configuration proprietà. unitEcco la chiave nella mappa di configurazione ed Celsius è il valore.

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

- Controlla se una proprietà della mappa contiene una determinata chiave e un valore:

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

- Controlla se una proprietà della mappa contiene una determinata chiave:

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

- Controlla se una proprietà dell'elenco contiene un determinato valore:

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

- Usa la lower() funzione per i confronti senza distinzione tra maiuscole e minuscole. Per impostazione predefinita, tutti i confronti fanno distinzione tra maiuscole e minuscole.

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

## AD ESEMPIO:

Utile se non si conosce il valore esatto di un campo e si può eseguire una ricerca a tutto testo nel campo specificato. %rappresenta zero o più.

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

- Ricerca Infix: %room%
- Ricerca con prefisso: room%

- Ricerca con suffissi: %room
- Se hai '%' nei tuoi valori, inserisci un carattere di escape in LIKE e specifica il carattere di escape conESCAPE.

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

## DISTINTO:

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

- La DISTINCT parola chiave elimina i duplicati dal risultato finale.

DISTINCTnon è supportato su tipi di dati complessi.

## CONTARE

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

- La COUNT parola chiave calcola il numero di elementi nel risultato di una query.
- COUNTnon è supportato nei campi complessi annidati e nei campi con pattern grafici.
- COUNTl'aggregazione non è supportata con le query DISTINCT annidate.

Ad esempio, COUNT(DISTINCT e.entityId) non è supportato.

## PERCORSO

Le seguenti proiezioni di pattern sono supportate nelle interrogazioni utilizzando la proiezione del percorso:

- Interrogazioni a hop variabile

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

Questa query abbina e proietta i metadati dei nodi di qualsiasi modello con bordi in uscita dal nodo a entro 1-3 passaggi.

- Interrogazioni a hop fisso

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

Questa query abbina e proietta i metadati delle entità e dei bordi in entrata su b.

- Interrogazioni non indirizzate

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

Questa query abbina e proietta i metadati dei nodi in modelli a 1 hop che collegano a e c tramite 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"
    }
  ]
}
]
}
]
}

```

Questa risposta alla PATH query comprende solo metadati che identificano tutti i nodi e i bordi di ogni percorso/pattern tra a e c tramite b.

#### LIMIT e OFFSET:

```

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

```

LIMIT specifica il numero di risultati da restituire nella query e OFFSET specifica il numero di risultati da ignorare.

#### LIMIT e MaxResults:

L'esempio seguente mostra una query che restituisce 500 risultati in totale, ma ne visualizza solo 50 alla volta per chiamata API. Questo modello può essere utilizzato quando è necessario limitare la quantità di risultati visualizzati, ad esempio se è possibile visualizzare solo 50 risultati in un'interfaccia utente.

```

aws iottwinmaker execute-query \
--workspace-id exampleWorkspace \

```

```
--query-statement "SELECT e FROM EntityGraph MATCH (e) LIMIT 500"\
--max-results 50
```

- La LIMIT parola chiave influisce sulla query e limita le righe risultanti. Se è necessario controllare il numero di risultati restituiti per chiamata API senza limitare il numero totale di risultati restituiti, utilizza LIMIT.
- max-results è un parametro opzionale per l'[azione dell'ExecuteQuery API](#). max-results si applica solo all'API e al modo in cui i risultati vengono letti entro i limiti della query precedente.

L'utilizzo max-results in una query consente di ridurre il numero di risultati visualizzati senza limitare il numero effettivo di risultati restituiti.

La query riportata di seguito scorre nella pagina successiva dei risultati. Questa query utilizza la chiamata ExecuteQuery API per restituire le righe 51-100, dove la pagina successiva dei risultati è specificata da next-token — in questo caso il token è: "H7kyGmvK376L"

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

- La next-token stringa specifica la pagina successiva dei risultati. Per ulteriori informazioni, consulta l'azione [ExecuteQuery API](#).

AWS IoT TwinMaker Knowledge Graph Query presenta i seguenti limiti:

Nome del limite	Quota	Regolabile
Timeout di esecuzione della query	10 secondi	No
Numero massimo di luppoli	10	Sì
Numero massimo di self s JOIN	20	Sì
Numero massimo di campi proiettati	20	Sì

Nome del limite	Quota	Regolabile
Numero massimo di espressioni condizionali (AND,,OR) NOT	10	Sì
Lunghezza massima di un modello di LIKE espressione (inclusi caratteri jolly e escape)	20	Sì
Numero massimo di elementi che possono essere specificati in una clausola IN	10	Sì
Valore massimo per OFFSET	3000	Sì
Valore massimo per LIMIT	3000	Sì
Valore massimo per le traverse (+) OFFSET LIMIT	3000	Sì



# Sincronizzazione delle risorse con AWS IoT SiteWise

AWS IoT TwinMaker supporta la sincronizzazione degli asset (asset sync) per gli AWS IoT SiteWise asset e i modelli di asset. Utilizzando il tipo di AWS IoT SiteWise componente, Asset Sync utilizza AWS IoT SiteWise gli asset e i modelli di asset esistenti e converte tali risorse in AWS IoT TwinMaker entità, componenti e tipi di componenti. Le sezioni seguenti illustrano come configurare la sincronizzazione degli asset e quali AWS IoT SiteWise risorse e modelli di asset possono essere sincronizzati con il tuo AWS IoT TwinMaker spazio di lavoro.

## Argomenti

- [Utilizzo della sincronizzazione delle risorse con AWS IoT SiteWise](#)
- [Differenze tra aree di lavoro personalizzate e predefinite](#)
- [Risorse sincronizzate da AWS IoT SiteWise](#)
- [Analizza lo stato e gli errori di sincronizzazione](#)
- [Eliminare un processo di sincronizzazione](#)
- [Limiti di sincronizzazione degli asset](#)

## Utilizzo della sincronizzazione delle risorse con AWS IoT SiteWise

Questo argomento mostra come attivare e configurare la sincronizzazione AWS IoT SiteWise delle risorse. Segui le procedure appropriate in base al tipo di spazio di lavoro che stai utilizzando.

### Important

[the section called “Differenze tra aree di lavoro personalizzate e predefinite”](#) Per informazioni sulle differenze tra gli spazi di lavoro personalizzati e quelli predefiniti, consulta.

## Argomenti

- [Utilizzo di un'area di lavoro personalizzata](#)
- [Utilizzo dell'loT SiteWiseDefaultWorkspace](#)

## Utilizzo di un'area di lavoro personalizzata

Esamina questi prerequisiti prima di attivare la sincronizzazione delle risorse.

## Prerequisiti

Prima dell'uso AWS IoT SiteWise, assicuratevi di aver fatto quanto segue:

- Hai uno AWS IoT TwinMaker spazio di lavoro.
- Hai creato alcuni asset e modelli di asset in AWS IoT SiteWise. Per ulteriori informazioni, consultate [Creazione di modelli di asset](#).
- Hai creato un ruolo IAM con autorizzazioni di lettura per le seguenti AWS IoT SiteWise azioni: `ListAssets`, `ListAssetModelsDescribeAsset`, e `DescribeAssetModel`.

Il ruolo IAM deve inoltre disporre delle seguenti autorizzazioni di scrittura per AWS IoT TwinMaker: `CreateEntity`, `UpdateEntity`, `DeleteEntity`, `CreateComponentType`, `UpdateComponentType`, e `DeleteComponentType`

Puoi utilizzare il seguente ruolo IAM come modello per il ruolo richiesto:

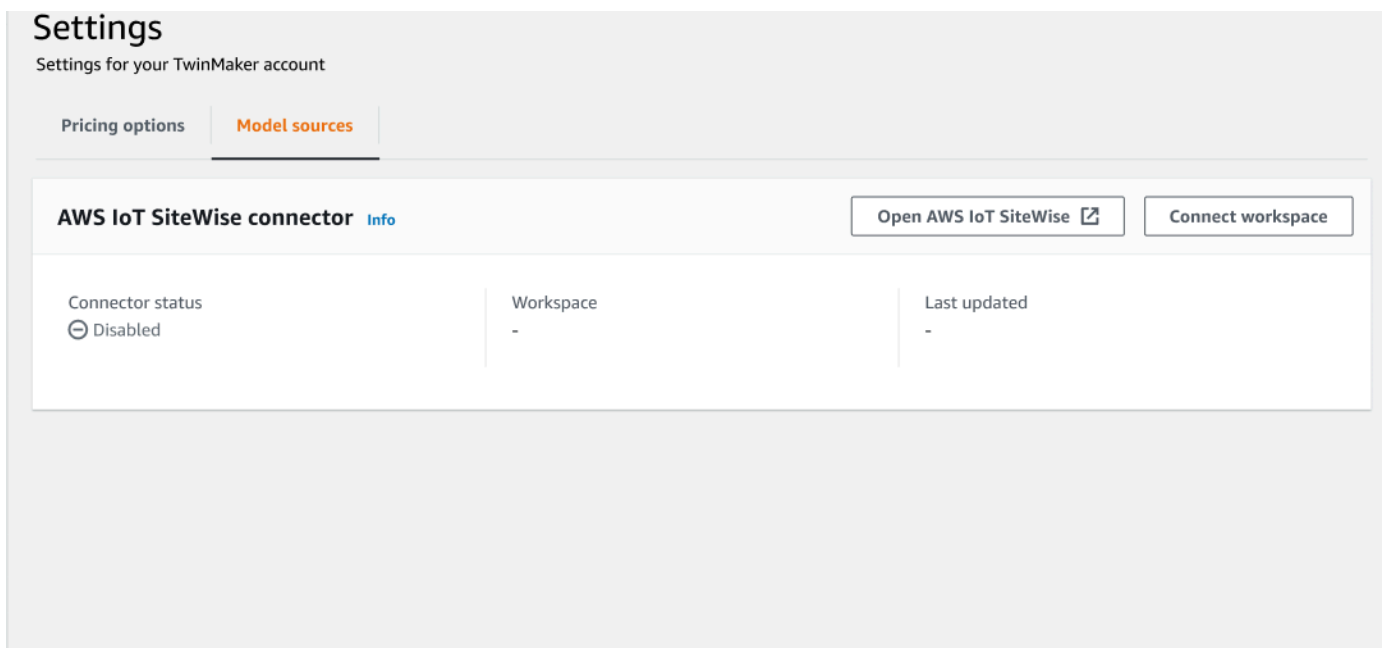
```
// trust relationships
{
  {
    "Version": "2012-10-17",
    "Statement": [
      {
        "Effect": "Allow",
        "Principal": {
          "Service": [
            "iottwinmaker.amazonaws.com",
            "iotsitewise.amazonaws.com"
          ]
        },
        "Action": "sts:AssumeRole"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

// permissions
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "iottwinmaker:*",
      "Resource": "*"
    }
  ]
}
```

```
    },
    {
      "Effect": "Allow",
      "Action": [
        "iotsitewise:Describe*",
        "iotsitewise:List*"
      ],
      "Resource": "*"
    }
  ]
}
```

Utilizzate la seguente procedura per attivare e configurare la sincronizzazione AWS IoT SiteWise degli asset.

1. Nella [AWS IoT TwinMaker console](#), vai alla pagina Impostazioni.
2. Apri la scheda Sorgenti del modello.



**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	-	-

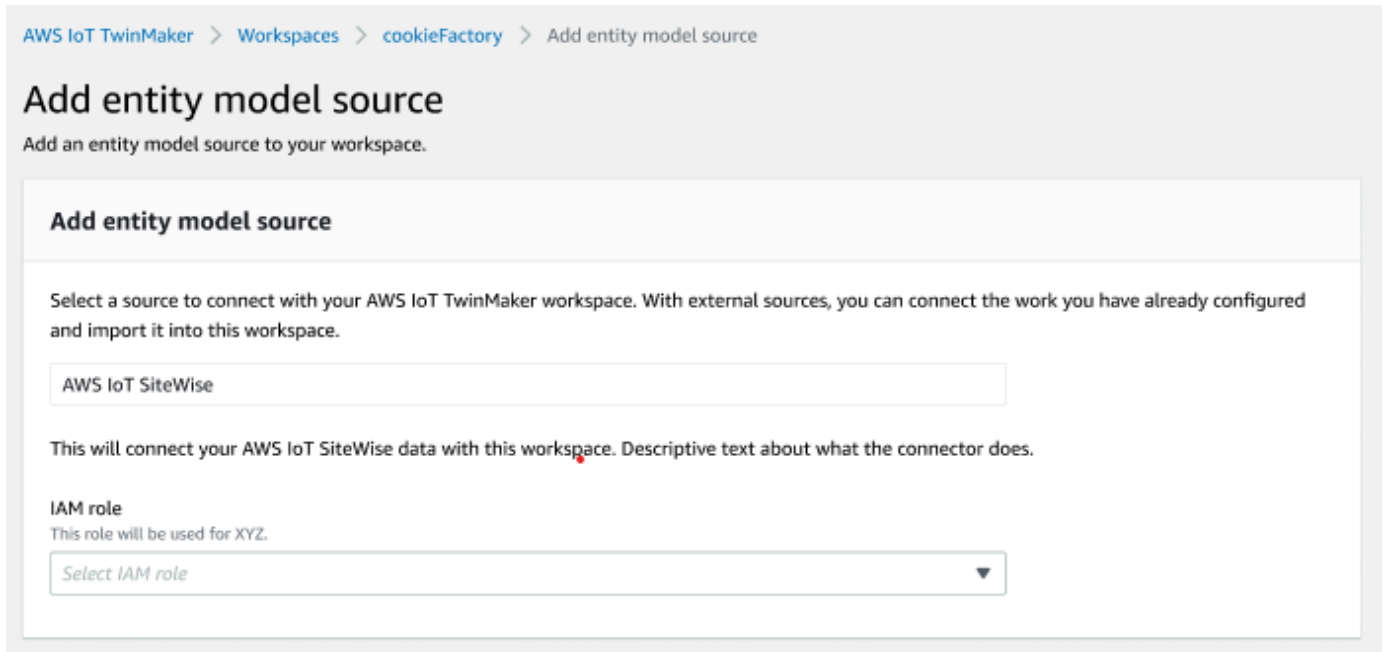
3. Scegli Connect workspace per collegare il tuo AWS IoT TwinMaker spazio di lavoro alle tue AWS IoT SiteWise risorse.

#### Note

Puoi utilizzare la sincronizzazione delle risorse solo con un singolo AWS IoT TwinMaker spazio di lavoro. È necessario disconnettere la sincronizzazione da un'area di lavoro e

connettersi a un'altra area di lavoro se si desidera eseguire la sincronizzazione in un'area di lavoro diversa.

- Successivamente, accedi all'area di lavoro in cui desideri utilizzare la sincronizzazione delle risorse.
- Scegliere Add sources (Aggiungi origini). Si apre la pagina di origine Aggiungi modello di entità.



- Nella pagina di origine Aggiungi modello di entità, verifica che sia visualizzato il campo di origine AWS IoT SiteWise. Seleziona il ruolo IAM che hai creato come prerequisito per il ruolo IAM.
- Ora hai attivato la sincronizzazione AWS IoT SiteWise delle risorse. Dovresti vedere apparire un banner di conferma nella parte superiore della pagina Workspace selezionata che conferma che la sincronizzazione delle risorse è attiva. Ora dovresti anche vedere una fonte di sincronizzazione elencata nella sezione Sorgenti del modello Entity.

The screenshot shows the AWS IoT TwinMaker console interface for a workspace named 'cookieFactory'. At the top right, there are 'View' and 'Delete' buttons. Below the workspace name, there is an 'Info' link and an 'Edit' button. The main content is divided into two sections: 'Workspace information' and 'Entity model sources (1)'. The 'Workspace information' section contains a table with the following data:

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

The 'Entity model sources (1)' section contains a table with one source:

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

## Utilizzo dell'IoT SiteWiseDefaultWorkspace

Quando attivi l'[AWS IoT SiteWiseAWS IoT TwinMaker integrazione](#),

IoTSiteWiseDefaultWorkspace viene creato e sincronizzato automaticamente uno spazio di lavoro predefinito denominato. AWS IoT SiteWise

Puoi anche utilizzare l' AWS IoT TwinMaker CreateWorkspaceAPI per creare uno spazio di lavoro denominato. IoTSiteWiseDefaultWorkspace

### Prerequisiti

Prima di creareIoTSiteWiseDefaultWorkspace, assicurati di aver fatto quanto segue:

- Crea un ruolo AWS IoT TwinMaker collegato al servizio. Per ulteriori informazioni, consulta [Utilizzo di ruoli collegati ai servizi per AWS IoT TwinMaker](#).
- Aprire la console IAM all'indirizzo <https://console.aws.amazon.com/iam/>.

Rivedi il ruolo o l'utente e verifica che disponga dell'autorizzazione a farlo.  
iotsitewise:EnableSiteWiseIntegration

Se necessario, aggiungi l'autorizzazione al ruolo o all'utente:

```
{
```

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

## Differenze tra aree di lavoro personalizzate e predefinite

### Important

AWS IoT SiteWise Le nuove funzionalità, ad esempio [CompositionModel](#), sono disponibili solo in `IoTSiteWiseDefaultWorkspace`. Ti consigliamo di utilizzare un'area di lavoro predefinita anziché un'area di lavoro personalizzata.

Quando si utilizza `IoTSiteWiseDefaultWorkspace`, ci sono alcune differenze notevoli rispetto all'utilizzo di un'area di lavoro personalizzata con sincronizzazione delle risorse.

- Quando crei uno spazio di lavoro predefinito, la posizione Amazon S3 e il ruolo IAM sono opzionali.

### Note

Puoi utilizzarlo `UpdateWorkspace` per fornire la posizione Amazon S3 e il ruolo IAM.

- `IoTSiteWiseDefaultWorkspace` Non ha un limite di numero di risorse con cui sincronizzare AWS IoT SiteWise AWS IoT TwinMaker le risorse.
- Quando sincronizzi le risorse da AWS IoT SiteWise, `SyncSource` lo saranno `SITWISE_MANAGED`. Ciò include `Entities` e `ComponentTypes`.
- Nuove AWS IoT SiteWise funzionalità, ad esempio `CompositionModel` disponibili solo in `IoTSiteWiseDefaultWorkspace`.

Esistono alcune limitazioni specifiche: `IoTSiteWiseDefaultWorkspace`

- L'area di lavoro predefinita non può essere eliminata.

- Per eliminare le risorse, è necessario prima eliminare le AWS IoT SiteWise risorse, quindi AWS IoT TwinMaker vengono eliminate le risorse corrispondenti in.

## Risorse sincronizzate da AWS IoT SiteWise

Questo argomento elenca le risorse che puoi sincronizzare con il tuo AWS IoT SiteWise spazio di lavoro AWS IoT TwinMaker.

### Important

[Differenze tra aree di lavoro personalizzate e predefinite](#) Per informazioni sulle differenze tra gli spazi di lavoro personalizzati e quelli predefiniti, consulta.

## Spazi di lavoro personalizzati e predefiniti

Le seguenti risorse sono sincronizzate e disponibili nelle aree di lavoro personalizzate e predefinite:

### Modelli di asset

AWS IoT TwinMaker crea un nuovo tipo di componente per ogni modello di asset in AWS IoT SiteWise.

- Il componente `TypeId` per il modello di asset utilizzerà uno dei seguenti modelli:
  - Spazio di lavoro personalizzato - `iotsitewise.assetmodel:assetModelId`
  - Spazio di lavoro predefinito - `assetModelId`
- Ogni proprietà nel modello di asset è una nuova proprietà nel tipo di componente, con uno dei seguenti modelli di denominazione:
  - Spazio di lavoro personalizzato - `Property_propertyId`
  - Spazio di lavoro predefinito - `propertyId`

Il nome della proprietà in AWS IoT SiteWise viene memorizzato come `displayName` nella definizione della proprietà.

- Ogni gerarchia nel modello di asset è una nuova proprietà di tipo `LIST` e `nestedType` si trova `RELATIONSHIP` nel tipo di componente. La gerarchia è mappata alla proprietà con un nome preceduto da uno dei seguenti:
  - Spazio di lavoro personalizzato - `Hierarchy_hierarchyId`

- Spazio di lavoro predefinito - *hierarchyId*

## Risorsa

AWS IoT TwinMaker crea una nuova entità per ogni risorsa in AWS IoT SiteWise.

- Il `entityId` è uguale `assetId` a quello in AWS IoT SiteWise.
- Queste entità hanno un unico componente chiamato `sitewiseBase`, che ha il tipo di componente corrispondente al modello di asset per questo asset.
- Qualsiasi modifica a livello di asset, ad esempio l'impostazione dell'`alias` di proprietà o dell'unità di misura, si riflette nell'entità in AWS IoT TwinMaker

## Solo area di lavoro predefinita

Le seguenti risorse sono sincronizzate e disponibili solo nell'area di lavoro predefinita,.

### `IoTSiteWiseDefaultWorkspace`

#### `AssetModelComponents`

AWS IoT TwinMaker crea un nuovo tipo di componente per ogni `AssetModelComponents` ingresso. AWS IoT SiteWise

- Il componente `TypeId` per il modello di asset utilizza il seguente schema:`assetModelId`.
- Ogni proprietà nel modello di asset è una nuova proprietà nel tipo di componente, con il nome della proprietà uguale `propertyId`. Il nome della proprietà in AWS IoT SiteWise viene memorizzato come `displayName` nella definizione della proprietà.
- Ogni gerarchia nel modello di asset è una nuova proprietà di tipo `LIST` e `nestedType` si trova `RELATIONSHIP` nel tipo di componente. La gerarchia è mappata alla proprietà con un nome preceduto da `hierarchyId`

#### `AssetModelCompositeModel`

AWS IoT TwinMaker crea un nuovo tipo di componente per ogni ingresso.

#### `AssetModelCompositeModel` AWS IoT SiteWise

- Il componente `TypeId` per il modello di asset utilizza il seguente schema:`assetModelId_assetModelCompositeModelId`.
- Ogni proprietà nel modello di asset è una nuova proprietà nel tipo di componente, con il nome della proprietà uguale `propertyId`. Il nome della proprietà in AWS IoT SiteWise viene memorizzato come `displayName` nella definizione della proprietà.



## AssetCompositeModels

AWS IoT TwinMaker crea un nuovo componente composito per ogni `AssetCompositeModel` ingresso AWS IoT SiteWise.

- Il `componentName` è uguale `assetModelCompositeModelId` a quello in AWS IoT SiteWise.

## Risorse non sincronizzate

Le seguenti risorse non sono sincronizzate:

### Asset e modelli di asset non sincronizzati

- I modelli di allarme verranno sincronizzati come `CompositeModels`, ma i dati corrispondenti nella risorsa relativi agli allarmi non vengono sincronizzati.
- [AWS IoT SiteWise i flussi di dati non vengono sincronizzati](#). Vengono sincronizzate solo le proprietà modellate nel modello di asset.
- I valori delle proprietà per gli attributi, le misurazioni, le trasformazioni, gli aggregati e il calcolo dei metadati come formula e finestra non vengono sincronizzati. Vengono sincronizzati solo i metadati relativi alle proprietà, come alias, unità di misura e tipo di dati. I valori possono essere interrogati utilizzando la normale AWS IoT TwinMaker API del connettore dati, [GetPropertyValueHistory](#)

## Utilizza entità e tipi di componenti sincronizzati in AWS IoT TwinMaker

Una volta sincronizzate le risorse AWS IoT SiteWise, i tipi di componenti sincronizzati vengono visualizzati in modalità di sola lettura. AWS IoT TwinMaker Qualsiasi azione di aggiornamento o eliminazione deve essere eseguita in AWS IoT SiteWise e tali modifiche vengono sincronizzate AWS IoT TwinMaker se `SyncJob` è ancora attivo.

Le entità sincronizzate e il componente di AWS IoT SiteWise base sono inoltre di sola lettura. AWS IoT TwinMaker È possibile aggiungere componenti non sincronizzati aggiuntivi all'entità sincronizzata, purché non vengano aggiornati attributi a livello di entità come la descrizione o. `entityName`

Alcune restrizioni si applicano al modo in cui è possibile interagire con le entità sincronizzate. Non è possibile creare entità secondarie in un'entità sincronizzata nella gerarchia dell'entità sincronizzata. Inoltre, non è possibile creare tipi di componenti non sincronizzati che si estendono da un tipo di componente sincronizzato.

**Note**

I componenti aggiuntivi vengono eliminati insieme all'entità se la risorsa viene eliminata in AWS IoT SiteWise o se si elimina il processo di sincronizzazione.

Puoi usare queste entità sincronizzate nelle dashboard di Grafana e aggiungerle come tag nel compositore di scene come entità normali. Puoi anche eseguire query sul Knowledge Graph per queste entità sincronizzate.

**Note**

Le entità sincronizzate senza modifiche non vengono addebitati, ma tali entità vengono addebitate all'utente se sono state apportate modifiche. AWS IoT TwinMaker Ad esempio, se aggiungi un componente non sincronizzato a un'entità sincronizzata, ora tale entità viene addebitata. AWS IoT TwinMaker Per ulteriori informazioni, consulta la sezione [Prezzi di AWS IoT TwinMaker](#).

## Analizza lo stato e gli errori di sincronizzazione

Questo argomento fornisce indicazioni su come analizzare gli errori e gli stati di sincronizzazione.

**Important**

[the section called “Differenze tra aree di lavoro personalizzate e predefinite”](#) Per informazioni sulle differenze tra gli spazi di lavoro personalizzati e quelli predefiniti, consulta.

## Sincronizza gli stati dei lavori

Un processo di sincronizzazione ha uno dei seguenti stati a seconda del relativo stato.

- **CREATING** Lo stato del processo di sincronizzazione indica che il processo sta verificando le autorizzazioni e caricando i dati da AWS IoT SiteWise per preparare la sincronizzazione.
- **INITIALIZING** Lo stato del processo di sincronizzazione indica che tutte le risorse esistenti in AWS IoT SiteWise sono sincronizzate con AWS IoT TwinMaker. Il completamento di questo

passaggio può richiedere più tempo se l'utente dispone di un gran numero di risorse e modelli di asset. AWS IoT SiteWise Puoi monitorare il numero di risorse che sono state sincronizzate controllando il processo di sincronizzazione nella [AWS IoT TwinMaker console](#) o chiamando l'`ListSyncResourcesAPI`.

- Lo **ACTIVE** stato del processo di sincronizzazione indica che la fase di inizializzazione è stata completata. Il processo è ora pronto per sincronizzare eventuali nuovi aggiornamenti da AWS IoT SiteWise.
- **ERROR**Lo stato del processo di sincronizzazione indica un errore con uno degli stati precedenti. Controlla il messaggio di errore. Potrebbe esserci un problema con la configurazione del ruolo IAM. Se desideri utilizzare un nuovo ruolo IAM, elimina il processo di sincronizzazione con l'errore e creane uno nuovo con il nuovo ruolo.

Gli errori di sincronizzazione vengono visualizzati nella pagina di origine del modello, a cui si accede dalla tabella delle fonti del modello Entity nell'area di lavoro. La pagina di origine del modello mostra un elenco di risorse che non sono state sincronizzate. La maggior parte degli errori viene ritentata automaticamente dal processo di sincronizzazione, ma se la risorsa richiede un'azione, rimane nello **ERROR** stato. È inoltre possibile ottenere un elenco di errori utilizzando l' [ListSyncResourcesAPI](#).

Per visualizzare tutti gli errori elencati per l'origine corrente, utilizzare la procedura seguente.

1. Accedi al tuo spazio di lavoro nella [AWS IoT TwinMaker console](#).
2. Seleziona la AWS IoT SiteWise fonte elencata nel modale Entity model sources per aprire la pagina dei dettagli di sincronizzazione degli asset.

**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. Come mostrato nella schermata precedente, tutte le risorse con errori persistenti sono elencate nella tabella Errori. È possibile utilizzare questa tabella per rintracciare e correggere errori relativi a risorse specifiche.

Gli errori possibili sono i seguenti:

- Sebbene AWS IoT SiteWise supporti nomi di risorse duplicati, li supporta AWS IoT TwinMaker solo a ROOT livello, non sotto la stessa entità principale. Se avete due risorse con lo stesso nome sotto un'entità principale in AWS IoT SiteWise, una di esse non riesce a sincronizzarsi. Per correggere questo errore, elimina una delle risorse o spostane una sotto un'altra risorsa principale AWS IoT SiteWise prima della sincronizzazione.
- Se disponi già di un'entità con lo stesso ID dell' AWS IoT SiteWise asset, tale risorsa non viene sincronizzata finché non elimini l'entità esistente.

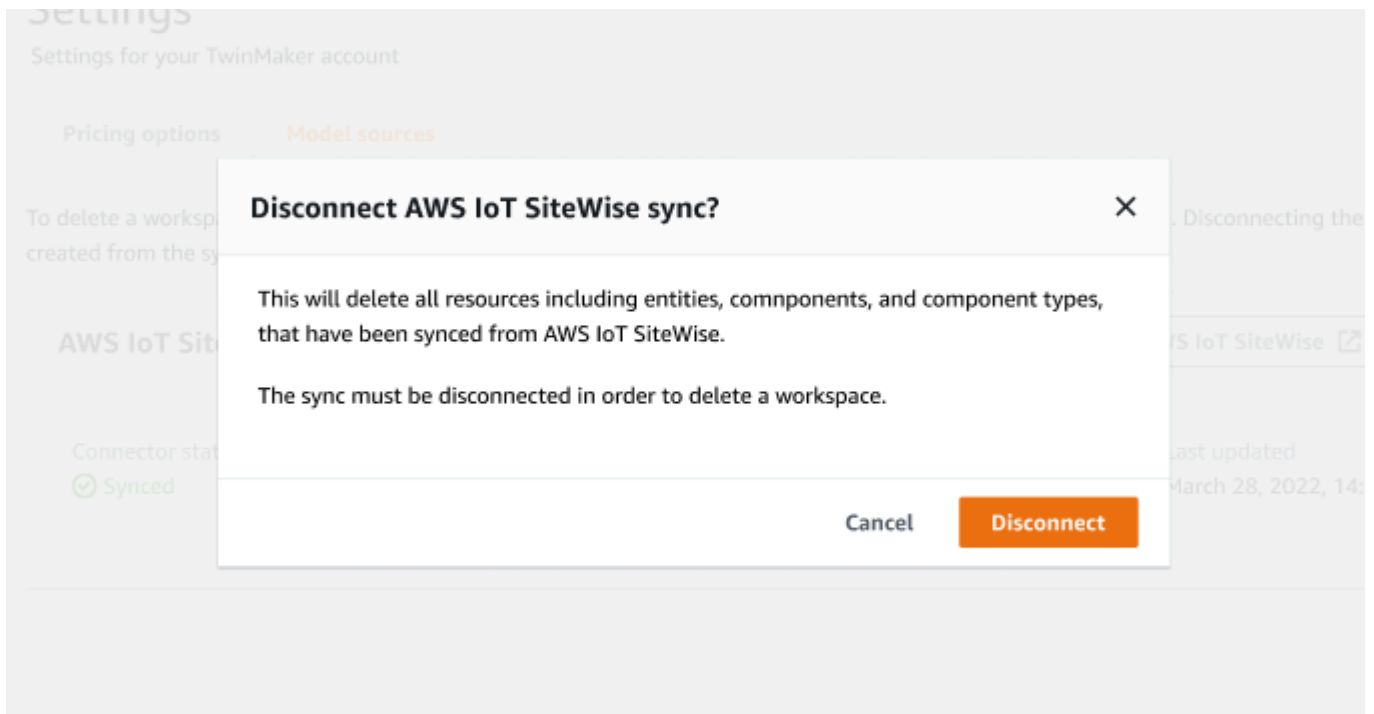
## Eliminare un processo di sincronizzazione

Utilizzare la procedura seguente per eliminare un processo di sincronizzazione.

**⚠ Important**

[the section called “Differenze tra aree di lavoro personalizzate e predefinite”](#) Per informazioni sulle differenze tra gli spazi di lavoro personalizzati e quelli predefiniti, vedere.

1. Passare alla [console AWS IoT TwinMaker](#).
2. Apri l'area di lavoro da cui desideri eliminare il processo di sincronizzazione.
3. In Entity model sources, seleziona la AWS IoT SiteWise fonte per aprire la pagina dei dettagli della fonte.
4. Per interrompere il processo di sincronizzazione, scegli Disconnetti. Conferma la tua scelta di eliminare completamente il processo di sincronizzazione.



Una volta eliminato un processo di sincronizzazione, è possibile crearlo nuovamente nella stessa area di lavoro o in un'area di lavoro diversa.

Non puoi eliminare un'area di lavoro se in quell'area di lavoro sono presenti lavori di sincronizzazione. Elimina i lavori di sincronizzazione prima di eliminare un'area di lavoro.

Se si verificano errori durante l'eliminazione del processo di sincronizzazione, il processo di sincronizzazione rimane nello DELETING stato e viene riprovato automaticamente. È ora possibile

eliminare manualmente qualsiasi entità o tipo di componente sincronizzato in caso di errori relativi all'eliminazione di una risorsa.

#### Note

Tutte le risorse da cui sono state sincronizzate AWS IoT SiteWise vengono prima eliminate, quindi viene eliminato il processo di sincronizzazione stesso.

## Limiti di sincronizzazione degli asset

#### Important

[the section called “Differenze tra aree di lavoro personalizzate e predefinite”](#) Per informazioni sulle differenze tra le aree di lavoro personalizzate e quelle predefinite, consulta.


Poiché le [AWS IoT SiteWise quote](#) sono superiori alle [AWS IoT TwinMaker quote](#) predefinite, stiamo aumentando i seguenti limiti per le entità e i tipi di componenti da cui è stata sincronizzata. AWS IoT SiteWise

- 1000 tipi di componenti sincronizzati in un'area di lavoro, poiché può sincronizzare solo 1000 modelli di asset da. AWS IoT SiteWise
- 100.000 entità sincronizzate in un'area di lavoro, poiché può sincronizzare solo 100.000 risorse da. AWS IoT SiteWise
- Un massimo di 2000 entità figlie per entità principale. Sincronizza 2000 risorse secondarie per singola risorsa principale.

#### Note

L'[GetEntity](#)API restituisce solo le prime 50 entità figlie per una proprietà gerarchica, ma è possibile utilizzare l'[GetPropertyValue](#)API per impaginare e recuperare l'elenco di tutte le entità secondarie.

- 600 proprietà per componente sincronizzato da AWS IoT SiteWise, in grado di sincronizzare i modelli di asset con 600 proprietà e gerarchie totali.

 **Note**

Questi limiti sono applicabili solo alle entità sincronizzate. Richiedi un aumento della quota se hai bisogno di aumentare questi limiti per le risorse non sincronizzate.

# AWS IoT TwinMakerIntegrazione della dashboard Grafana

AWS IoT TwinMakersupporta l'integrazione di Grafana tramite un plugin applicativo. Puoi utilizzare Grafana versione 8.2.0 e successive per interagire con la tua applicazione digital twin. Il AWS IoT TwinMaker plug-in fornisce pannelli personalizzati, modelli di dashboard e un'origine dati per la connessione ai dati del gemello digitale.

Per ulteriori informazioni su come effettuare l'onboarding con Grafana e configurare le autorizzazioni per la dashboard, consulta i seguenti argomenti:

## Argomenti

- [Configurazione CORS per il visualizzatore di scene Grafana](#)
- [Configurazione dell'ambiente Grafana](#)
- [Creazione di un ruolo IAM nella dashboard](#)
- [Creazione di una politica per i lettori AWS IoT TwinMaker video](#)

### Note

È necessario modificare la configurazione CORS (cross-origin resource sharing) del bucket Amazon S3 per consentire all'interfaccia utente Grafana di caricare risorse dal bucket. Per le istruzioni, consulta [Configurazione CORS per il visualizzatore di scene Grafana](#).

Per ulteriori informazioni sul plugin AWS IoT TwinMaker Grafana, consulta la documentazione dell'[AWS IoT TwinMakerapp](#).

Per ulteriori informazioni sui componenti chiave del plugin Grafana, consulta quanto segue:

- [AWS IoT TwinMakerfonte di dati](#)
- [modelli di dashboard](#)
- [Pannello Scene Viewer](#)
- [Pannello Video Player](#)



# Configurazione CORS per il visualizzatore di scene Grafana

Il plugin AWS IoT TwinMaker Grafana richiede una configurazione CORS (cross-origin resource sharing), che consente all'interfaccia utente Grafana di caricare risorse dal bucket Amazon S3. Senza la configurazione CORS, riceverai un messaggio di errore come «Load 3D Scene failed with Network Failure» sullo Scene Viewer poiché il dominio Grafana non può accedere alle risorse nel bucket Amazon S3.

Per configurare il tuo bucket Amazon S3 con CORS, segui i seguenti passaggi:

1. Accedi alla console IAM e apri la console [Amazon S3](#).
2. Nell'elenco Bucket, scegli il nome del bucket che usi come bucket di risorse del tuo AWS IoT TwinMaker spazio di lavoro.
3. Seleziona Autorizzazioni.
4. Nella sezione Condivisione di risorse tra origini diverse, seleziona Modifica per aprire l'editor CORS.
5. Nella casella di testo dell'editor di configurazione CORS, digita o copia e incolla la seguente configurazione JSON CORS sostituendo il dominio dell'area di lavoro Grafana con il tuo dominio.

***GRAFANA-WORKSPACE-DOMAIN***

## Note

È necessario mantenere il \* carattere asterisco all'inizio dell'elemento JSON.

"AllowedOrigins":

```
{
  [
    "AllowedHeaders": [
      "*"
    ],
    "AllowedMethods": [
      "GET",
      "PUT",
      "POST",
      "DELETE",
      "HEAD"
    ]
  ]
}
```

```
    ],
    "AllowedOrigins": [
      "*GRAFANA-WORKSPACE-DOMAIN"
    ],
    "ExposeHeaders": [
      "ETag"
    ]
  }
]
```

6. Seleziona **Salva modifiche** per completare la configurazione CORS.

Per ulteriori informazioni su CORS con bucket Amazon S3, [consulta Using cross-origin resource sharing \(CORS\)](#).

## Configurazione dell'ambiente Grafana

Puoi usare Amazon Managed Grafana per un servizio completamente gestito o configurare un ambiente Grafana che gestisci tu stesso. Con Amazon Managed Grafana, puoi distribuire, gestire e scalare rapidamente Grafana open source in base alle tue esigenze. In alternativa, puoi configurare la tua infrastruttura per gestire i server Grafana.

Per ulteriori informazioni su entrambe le opzioni dell'ambiente Grafana, consulta i seguenti argomenti:

- [Grafana gestito da Amazon](#)
- [Grafana autogestita](#)

### Grafana gestito da Amazon

Amazon Managed Grafana fornisce un AWS IoT TwinMaker plug-in che ti consente di integrarti rapidamente con AWS IoT TwinMaker Grafana. Poiché Amazon Managed Grafana gestisce i server Grafana per te, puoi visualizzare i tuoi dati senza dover creare, impacchettare o distribuire hardware o altre infrastrutture Grafana. Per ulteriori informazioni su Amazon Managed Grafana, consulta [Cos'è Amazon Managed Grafana?](#) .

**Note**

Amazon Managed Grafana attualmente supporta la versione 1.3.1 del plugin AWS IoT TwinMaker Grafana.

## Prerequisiti di Amazon Managed Grafana

Per utilizzarlo AWS IoT TwinMaker in una dashboard di Amazon Managed Grafana, devi prima completare i seguenti prerequisiti:

- Crea uno spazio di lavoro AWS IoT TwinMaker. Per ulteriori informazioni sulla creazione di aree di lavoro, consulta [Guida introduttiva](#). AWS IoT TwinMaker

**Note**

Quando crei per la prima volta un'area di lavoro Amazon Managed Grafana nella console di AWS gestione, AWS IoT TwinMaker non è elencata. Tuttavia, il plug-in è già installato su tutte le aree di lavoro. Puoi trovare il AWS IoT TwinMaker plugin nell'elenco dei plugin open source Grafana. Puoi trovare l'AWS IoT TwinMaker origine dati selezionando Aggiungi un'origine dati nella pagina Datasources.

Quando crei un'area di lavoro Amazon Managed Grafana, viene creato automaticamente un ruolo IAM per gestire le autorizzazioni per l'istanza Grafana. Questo è chiamato Workspace IAM Role. È l'opzione del provider di autenticazione che utilizzerai per configurare tutte le AWS IoT TwinMaker fonti di dati per Grafana. Amazon Managed Grafana non supporta l'aggiunta automatica di autorizzazioni per AWS IoT TwinMaker, quindi devi configurarle manualmente. Per ulteriori informazioni sulla configurazione delle autorizzazioni manuali, consulta [Creazione di un ruolo IAM nella dashboard](#)

## Grafana autogestita

Puoi scegliere di ospitare la tua infrastruttura per eseguire Grafana. Per informazioni sull'esecuzione di Grafana localmente sul computer, consulta [Installa Grafana](#). Il AWS IoT TwinMaker plugin è disponibile sul catalogo pubblico Grafana. [Per informazioni sull'installazione di questo plugin nell'ambiente Grafana, vedi AWS IoT TwinMaker App.](#)

Quando esegui Grafana localmente, non puoi condividere facilmente dashboard o fornire l'accesso a più utenti. [Per una guida rapida basata su script sulla condivisione di dashboard utilizzando Grafana locale, consulta l'archivio degli esempi. AWS IoT TwinMaker](#) Questa risorsa illustra come ospitare un ambiente Grafana su Cloud9 e Amazon EC2 su un endpoint pubblico.

Devi determinare quale provider di autenticazione utilizzerai per configurare le fonti di dati. TwinMaker Le credenziali per l'ambiente vengono configurate in base alla catena di credenziali predefinita (vedere [Utilizzo della catena di provider di credenziali predefinita](#)). Le credenziali predefinite possono essere le credenziali permanenti di qualsiasi utente o ruolo. Ad esempio, se esegui Grafana su Amazon EC2, la catena di credenziali predefinita ha accesso al ruolo di [esecuzione di Amazon EC2, che sarebbe quindi](#) il tuo provider di autenticazione. L'IAM Amazon Resource Name (ARN) del provider di autenticazione è richiesto nei passaggi da [Creazione di un ruolo IAM nella dashboard](#)

## Creazione di un ruolo IAM nella dashboard

Con AWS IoT TwinMaker, puoi controllare l'accesso ai dati sulle dashboard Grafana. Gli utenti della dashboard di Grafana devono disporre di ambiti di autorizzazione diversi per visualizzare i dati e, in alcuni casi, scrivere dati. Ad esempio, un operatore di allarmi potrebbe non avere l'autorizzazione per visualizzare i video, mentre un amministratore ha l'autorizzazione per tutte le risorse. Grafana definisce le autorizzazioni tramite fonti di dati, dove vengono fornite credenziali e un ruolo IAM. L'AWS IoT TwinMaker origine dati AWS recupera le credenziali con le autorizzazioni per quel ruolo. Se non viene fornito un ruolo IAM, Grafana utilizza l'ambito delle credenziali, che non può essere ridotto di. AWS IoT TwinMaker

Per utilizzare le AWS IoT TwinMaker dashboard in Grafana, devi creare un ruolo IAM e allegare delle policy. Puoi utilizzare i seguenti modelli per aiutarti a creare queste politiche.

## Creazione di una policy IAM

Crea una policy IAM chiamata *YourWorkspaceId*DashboardPolicy nella console IAM. Questa policy consente ai tuoi spazi di lavoro di accedere al AWS IoT TwinMaker bucket e alle risorse di Amazon S3. Puoi anche decidere di utilizzare [AWS IoT GreengrassEdge Connector per Amazon Kinesis Video](#) Streams, che richiede le autorizzazioni per Kinesis AWS IoT SiteWise Video Streams e le risorse configurate per il componente. Per adattarlo al tuo caso d'uso, scegli uno dei seguenti modelli di policy.

## 1. Nessuna politica sulle autorizzazioni video

Se non desideri utilizzare il [pannello Grafana Video Player](#), crea la policy utilizzando il seguente modello.

```
{
  "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": "*"
    }
  ]
}
```

Viene creato un bucket Amazon S3 per ogni area di lavoro. Contiene i modelli e le scene 3D da visualizzare su una dashboard. Il [SceneViewer](#) pannello carica gli elementi da questo bucket.

## 2. Politica sulle autorizzazioni video ridotte

Per limitare l'accesso al pannello Video Player di Grafana, raggruppa le risorse di AWS IoT Greengrass Edge Connector per Amazon Kinesis Video Streams per tag. Per ulteriori informazioni sulla definizione delle autorizzazioni per le tue risorse video, consulta [Creazione di una politica per i lettori AWS IoT TwinMaker video](#)

## 3. Tutte le autorizzazioni video

Se non vuoi raggruppare i tuoi video, puoi renderli tutti accessibili da Grafana Video Player. Chiunque abbia accesso a un'area di lavoro Grafana può riprodurre video per qualsiasi streaming del tuo account e avere accesso in sola lettura a qualsiasi risorsa. AWS IoT SiteWise Ciò include tutte le risorse che verranno create in futuro.

Crea la politica con il seguente modello:

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

```

Questo modello di policy fornisce le seguenti autorizzazioni:

- Accesso in sola lettura a un bucket S3 per caricare una scena.
- Accesso in sola lettura a tutte AWS IoT TwinMaker le entità e i componenti in un workspace.
- Accesso in sola lettura per lo streaming di tutti i video di Kinesis Video Streams presenti nel tuo account.
- Accesso in sola lettura alla cronologia dei valori delle proprietà di tutti gli AWS IoT SiteWise asset del tuo account.

- Inserimento di dati in qualsiasi proprietà di un AWS IoT SiteWise asset contrassegnata con la chiave `EdgeConnectorForKVS` e il valore `.workspaceId`

## Taggando la AWS IoT SiteWise risorsa della videocamera, richiedi il caricamento del video da Edge.

Utilizzando il Video Player di Grafana, gli utenti possono richiedere manualmente che il video venga caricato dalla cache edge su Kinesis Video Streams. Puoi attivare questa funzionalità per qualsiasi AWS IoT SiteWise risorsa associata al tuo AWS IoT Greengrass Edge Connector per Amazon Kinesis Video Streams e contrassegnata con la chiave `EdgeConnectorForKVS`

Il valore del tag può essere un elenco di `WorkspaceID` delimitato da uno dei seguenti caratteri: `. . :`  
`+ = @ _ / -` Ad esempio, se desideri utilizzare una AWS IoT SiteWise risorsa associata a un AWS IoT Greengrass Edge Connector per Amazon Kinesis Video Streams tra aree di lavoro, puoi utilizzare un tag che segue questo schema: `. WorkspaceA/WorkspaceB/WorkspaceC`  
 Il plugin Grafana impone che AWS IoT TwinMaker `WorkspaceID` venga utilizzato per raggruppare l'ingestione dei dati degli asset. AWS IoT SiteWise

## Aggiungi altre autorizzazioni alla policy della dashboard

Il plugin AWS IoT TwinMaker Grafana utilizza il tuo provider di autenticazione per richiamare `AssumeRole` il ruolo di dashboard che crei. Internamente, il plugin limita l'ambito massimo di autorizzazioni a cui hai accesso utilizzando una politica di sessione nella chiamata `AssumeRole`. [Per ulteriori informazioni sui criteri di sessione, consulta Politiche di sessione.](#)

Questa è la politica massima permissiva che puoi avere nel tuo ruolo di dashboard per uno AWS IoT TwinMaker spazio di lavoro:

```
{
  "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 aggiungi istruzioni che richiedono Allow ulteriori autorizzazioni, queste non funzioneranno sul plugin. AWS IoT TwinMaker Questo è stato progettato per garantire che il plugin utilizzi le autorizzazioni minime necessarie.

Tuttavia, è possibile ridurre ulteriormente le autorizzazioni. Per informazioni, consulta [Creazione di una politica per i lettori AWS IoT TwinMaker video](#).

## Creazione del ruolo IAM di Grafana Dashboard

Nella console IAM, crea un ruolo IAM chiamato *YourWorkspaceId*DashboardRole. *YourWorkspaceId*DashboardPolicy Associa il al ruolo.

Per modificare la politica di attendibilità del ruolo del dashboard, devi autorizzare il provider di autenticazione Grafana a richiamare AssumeRole il ruolo del dashboard. Aggiorna la politica di fiducia con il seguente modello:

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Principal": {  
        "AWS": "ARN of Grafana authentication provider"  
      },  
      "Action": "sts:AssumeRole"  
    }  
  ]  
}
```

Per ulteriori informazioni sulla creazione di un ambiente Grafana e sulla ricerca del provider di autenticazione, consulta. [Configurazione dell'ambiente Grafana](#)

## Creazione di una politica per i lettori AWS IoT TwinMaker video

Di seguito è riportato un modello di policy con tutte le autorizzazioni video necessarie per il AWS IoT TwinMaker plugin in 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*"
      }
    }
  }
]
}

```

Per ulteriori informazioni sulla politica completa, consulta il modello di policy per le autorizzazioni per tutti i video nell'argomento. [Creazione di una policy IAM](#)

## Limita l'accesso alle tue risorse

Il pannello Video Player di Grafana richiama direttamente Kinesis Video Streams e SiteWise IoT per fornire un'esperienza di riproduzione video completa. Per evitare l'accesso non autorizzato a risorse che non sono associate al tuo AWS IoT TwinMaker spazio di lavoro, aggiungi delle condizioni alla policy IAM per il tuo ruolo nella dashboard del workspace.

## Riduci i permessi GET

Puoi limitare l'accesso ai tuoi Amazon Kinesis Video AWS IoT SiteWise Streams e alle tue risorse taggando le risorse. Potresti aver già taggato la tua risorsa AWS IoT SiteWise fotografica in base al AWS IoT TwinMaker WorkspaceID per abilitare la funzionalità di richiesta di caricamento dei video, consulta l'argomento [Upload](#) video from the edge. Puoi utilizzare la stessa coppia chiave-valore di tag per limitare l'accesso GET alle AWS IoT SiteWise risorse e anche per taggare i tuoi Kinesis Video Streams allo stesso modo.

Puoi quindi aggiungere questa condizione alle istruzioni kinesisvideo e iotsitewise in:

*YourWorkspaceId*DashboardPolicy

```

"Condition": {

```

```

"StringLike": {
  "aws:ResourceTag/EdgeConnectorForKVS": "*workspaceId*"
}
}

```

## Caso d'uso reale: raggruppamento di telecamere

In questo scenario, si dispone di un'ampia gamma di telecamere che monitorano il processo di cottura dei biscotti in una fabbrica. I lotti di pastella per biscotti vengono preparati nella sala pastelli, la pastella viene congelata nella sala congelatori e i biscotti vengono cotti nella sala cottura. In ognuna di queste stanze sono presenti telecamere con diversi team di operatori che monitorano separatamente ogni processo. Desiderate che ogni gruppo di operatori sia autorizzato per la rispettiva sala. Quando si crea un gemello digitale per la fabbrica di biscotti, viene utilizzata un'unica area di lavoro, ma le autorizzazioni della fotocamera devono essere limitate per stanza.

È possibile ottenere questa separazione delle autorizzazioni contrassegnando gruppi di telecamere in base al relativo GroupingID. In questo scenario, i GroupingID sono, e. BatterRoom FreezerRoom BakingRoom La videocamera di ogni stanza è collegata a Kinesis Video Streams e deve avere un tag con: EdgeConnectorForKVS Key =, Value =. BatterRoom Il valore può essere un elenco di raggruppamenti delimitati da uno dei seguenti caratteri: . : + = @ \_ / -

Per modificare il *YourWorkspaceId* DashboardPolicy, utilizza le seguenti dichiarazioni politiche:

```

...,
{
  "Effect": "Allow",
  "Action": [
    "kinesisvideo:GetDataEndpoint",
    "kinesisvideo:GetHLSStreamingSessionURL"
  ],
  "Resource": "*",
  "Condition": {
    "StringLike": {
      "aws:ResourceTag/EdgeConnectorForKVS": "*groupingId*"
    }
  }
},
{
  "Effect": "Allow",
  "Action": [
    "iotsitewise:GetAssetPropertyValue",

```

```

    "iotsitewise:GetInterpolatedAssetPropertyValues"
  ],
  "Resource": "*",
  "Condition": {
    "StringLike": {
      "aws:ResourceTag/EdgeConnectorForKVS": "*groupingId*"
    }
  }
},
...

```

Queste istruzioni limitano la riproduzione di video in streaming e l'accesso alla cronologia delle AWS IoT SiteWise proprietà a risorse specifiche di un raggruppamento. *groupingId* È definito dal tuo caso d'uso. Nel nostro scenario, sarebbe RoomID.

## Ambita l'autorizzazione AWS IoT SiteWise BatchPutAssetPropertyValue

Fornendo questa autorizzazione si attiva la [funzione di richiesta di caricamento video in Video Player](#). Quando carichi un video, puoi specificare un intervallo di tempo e inviare la richiesta scegliendo Invia nel pannello della dashboard di Grafana.

Per concedere le BatchPutAssetPropertyValue autorizzazioni a iotsitewise:, usa la politica predefinita:

```

...,
{
  "Effect": "Allow",
  "Action": [
    "iotsitewise:BatchPutAssetPropertyValue"
  ],
  "Resource": "*",
  "Condition": {
    "StringLike": {
      "aws:ResourceTag/EdgeConnectorForKVS": "*workspaceId*"
    }
  }
},
...

```

Utilizzando questo criterio, gli utenti possono BatchPutAssetPropertyValue richiedere qualsiasi proprietà sulla risorsa fotocamera. AWS IoT SiteWise È possibile limitare l'autorizzazione per un PropertyID specifico specificandolo nella condizione dell'istruzione.

```
{
  ...
  "Condition": {
    "StringEquals": {
      "iotsitewise:propertyId": "propertyId"
    }
  }
  ...
}
```

Il pannello Video Player di Grafana inserisce i dati nella proprietà di misurazione, denominata VideoUploadRequest, per avviare il caricamento del video dalla cache edge a Kinesis Video Streams. Trova il PropertyID di questa proprietà nella console. AWS IoT SiteWise Per modificare il *YourWorkspaceId* DashboardPolicy, utilizzare la seguente dichiarazione politica:

```
...,
{
  "Effect": "Allow",
  "Action": [
    "iotsitewise:BatchPutAssetPropertyValue"
  ],
  "Resource": "*",
  "Condition": {
    "StringLike": {
      "aws:ResourceTag/EdgeConnectorForKVS": "*workspaceId*"
    },
    "StringEquals": {
      "iotsitewise:propertyId": "VideoUploadRequestPropertyId"
    }
  }
},
...
```

Questa istruzione limita l'importazione di dati a una proprietà specifica della risorsa fotocamera contrassegnata. AWS IoT SiteWise Per ulteriori informazioni, consulta [How AWS IoT SiteWise works with IAM](#).

# Connect AWS IoT SiteWise Alarms ai dashboard AWS IoT TwinMaker Grafana

## Note

Questa funzionalità è disponibile in anteprima pubblica ed è soggetta a modifiche.

AWS IoT TwinMaker è in grado di importare allarmi AWS IoT SiteWise ed eventi nei AWS IoT TwinMaker componenti. Ciò consente di interrogare lo stato degli allarmi e configurare le soglie di allarme senza implementare un connettore dati personalizzato per AWS IoT SiteWise la migrazione dei dati. Puoi utilizzare il plugin AWS IoT TwinMaker Grafana per visualizzare lo stato degli allarmi e configurare la soglia di allarme in Grafana, senza effettuare chiamate API AWS IoT TwinMaker o interagire direttamente con gli allarmi. AWS IoT SiteWise

## AWS IoT SiteWise prerequisiti per la configurazione degli allarmi

Prima di creare allarmi e integrarli nella dashboard Grafana, assicurati di aver esaminato i seguenti prerequisiti:

- Acquisisci familiarità con il modello e il sistema AWS IoT SiteWise di asset di questo sito. Per ulteriori informazioni, consulta [Creazione di modelli di risorse](#) e [Creazione di risorse](#) nella Guida per l'utente.
- Acquisisci familiarità con i modelli di allarme IoT Events e come collegarli a un AWS IoT SiteWise modello. Per ulteriori informazioni, consulta [Definizione AWS IoT degli allarmi Events](#) nella Guida per l'utente.
- Integra AWS IoT TwinMaker con Grafana in modo da poter accedere alle tue AWS IoT TwinMaker risorse in Grafana. Per ulteriori informazioni, consulta [AWS IoT TwinMakerIntegrazione della dashboard Grafana](#).

## Definisci il ruolo IAM del componente di AWS IoT SiteWise allarme

AWS IoT TwinMaker utilizza il ruolo IAM del workspace per interrogare e configurare la soglia di allarme in Grafana. Le seguenti autorizzazioni sono necessarie nel ruolo di AWS IoT TwinMaker area di lavoro, per interagire con gli AWS IoT SiteWise allarmi in Grafana:



```
{
  "Effect": "Allow",
  "Action": [
    "iotevents:DescribeAlarmModel",
  ],
  "Resource": ["{IoTEventsAlarmModelArn}"]
},{
  "Effect": "Allow",
  "Action": [
    "iotsitewise:BatchPutAssetPropertyValue"
  ],
  "Resource": ["{IoTSitewiseAssetArn}"]
}
```

Nella [AWS IoT TwinMaker console](#), crea un'entità che rappresenti la tua risorsa. AWS IoT SiteWise Assicurati di aggiungere un componente per quell'entità utilizzandolo com.amazon.iotsitewise.alarm come tipo di componente e scegli i modelli di asset e allarme corrispondenti.

## 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

La schermata precedente è un esempio di creazione di questa entità con il tipo com.amazon.iotsitewise.alarm

Quando crei questo componente, importa AWS IoT TwinMaker automaticamente le relative proprietà di allarme da AWS IoT SiteWise e AWS IoT Events. È possibile ripetere questo schema di tipo di componente di allarme per creare componenti di allarme per tutte le risorse necessarie nell'area di lavoro.

## Interroga e aggiorna tramite l'API AWS IoT TwinMaker

Dopo aver creato i componenti di allarme, puoi interrogare lo stato, la soglia e aggiornare le soglie di allarme tramite l' AWS IoT TwinMaker API.

Di seguito è riportato un esempio di richiesta per interrogare lo stato degli allarmi:

```
aws iottwinmaker get-property-value-history --cli-input-json \  
{  
  "workspaceId": "{workspaceId}",  
  "entityId": "{entityId}",  
  "componentName": "{componentName}",  
  "selectedProperties": ["alarm_status"],  
  "startTime": "{startTimeIsoString}",  
  "endTime": "{endTimeIsoString}"  
}
```

Di seguito è riportato un esempio di richiesta per interrogare la soglia di allarme.

```
aws iottwinmaker get-property-value-history --cli-input-json \  
{  
  "workspaceId": "{workspaceId}",  
  "entityId": "{entityId}",  
  "componentName": "{componentName}",  
  "selectedProperties": ["alarm_threshold"],  
  "startTime": "{startTimeIsoString}",  
  "endTime": "{endTimeIsoString}"  
}
```

Di seguito è riportato un esempio di richiesta per aggiornare la soglia di allarme:

```
aws iottwinmaker batch-put-property-values --cli-input-json \  
{  
  "workspaceId": "{workspaceId}",  
  "entries": [  
    {
```

```

    "entityPropertyReference": {
      "entityId": "{entityId}",
      "componentName": "{componentName}",
      "propertyName": "alarm_threshold"
    },
    "propertyValues": [
      {
        "value": {
          "doubleValue": "{newThreshold}"
        },
        "time": "{effectiveTimeIsoString}"
      }
    ]
  }
]
}'

```

## Configura la dashboard Grafana per gli allarmi

È necessario creare un secondo ruolo IAM del dashboard abilitato alla scrittura, che è un ruolo normale ma con l'autorizzazione `iotttwinmaker:BatchPutPropertyValues` all'aggiunta dell'azione allo TwinMaker spazio di lavoro arn, come nell'esempio seguente.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iotttwinmaker:Get*",
        "iotttwinmaker:List*",
        "iotttwinmaker:BatchPutPropertyValues"
      ],
      "Resource": [
        "{workspaceArn}",
        "{workspaceArn}/*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": "iotttwinmaker:ListWorkspaces",
      "Resource": "*"
    }
  ]
}

```

```
    }  
  ]  
}
```

In alternativa, puoi aggiungere questa dichiarazione alla fine del tuo ruolo IAM:

```
{  
  "Effect": "Allow",  
  "Action": [  
    "iottwinmaker:BatchPutPropertyValues"  
  ],  
  "Resource": [  
    "{workspaceArn}",  
    "{workspaceArn}/*"  
  ]  
}
```

L'origine dati deve avere il suo arn di scrittura impostato con il ruolo di scrittura della dashboard che hai creato.

Dopo aver modificato il tuo ruolo IAM, accedi alla dashboard Grafana per assumere il ruolo aggiornato arn. Seleziona la casella di controllo Definisci i permessi di scrittura per Alarm Configuration Panel e copia l'arn per il ruolo Write.

The screenshot shows the 'Settings' page for a data source in AWS IoT TwinMaker. At the top, there are tabs for 'Settings' and 'Dashboards'. Below the tabs, the data source name is 'AWS IoT TwinMaker-4' and there is a 'Default' toggle switch. The 'Connection Details' section includes fields for 'Authentication Provider' (AWS SDK Default), 'Assume Role ARN' (arn:aws:iam:\*), 'External ID' (External ID), 'Endpoint' (https://{service}.{region}.amazonaws.com), and 'Default Region' (us-east-1). A warning box with a red triangle icon is titled 'Assume Role ARN' and contains the 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.' Below this is the 'TwinMaker settings' section, which includes a 'Workspace' field (enter workspace ID), a checked checkbox for 'Define write permissions for Alarm Configuration Panel', and an 'Assume Role ARN Write' field (arn:aws:iam:\*) which is circled in red. At the bottom, there are buttons for 'Back', 'Explore', 'Delete', and 'Save & test'.

## Usa la dashboard Grafana per la visualizzazione degli allarmi

Utilizza la seguente procedura per aggiungere un pannello di configurazione degli allarmi alla dashboard e configurarlo:

1. Seleziona l'area di lavoro nelle opzioni del pannello.

2. Imposta la tua fonte di dati nella configurazione della query.
3. Usa il seguente tipo di query: `Get Property Value History by Entity`
4. Seleziona un'entità o una variabile di entità a cui desideri aggiungere un allarme.
5. Dopo aver selezionato l'entità, selezionate un componente o una variabile di componente a cui applicare una proprietà.
6. Per la proprietà, scegliete: `alarm_status` e `alarm_threshold`.

Quando è connesso, dovresti vedere l'ID dell'allarme e la sua soglia attuale.

### Note

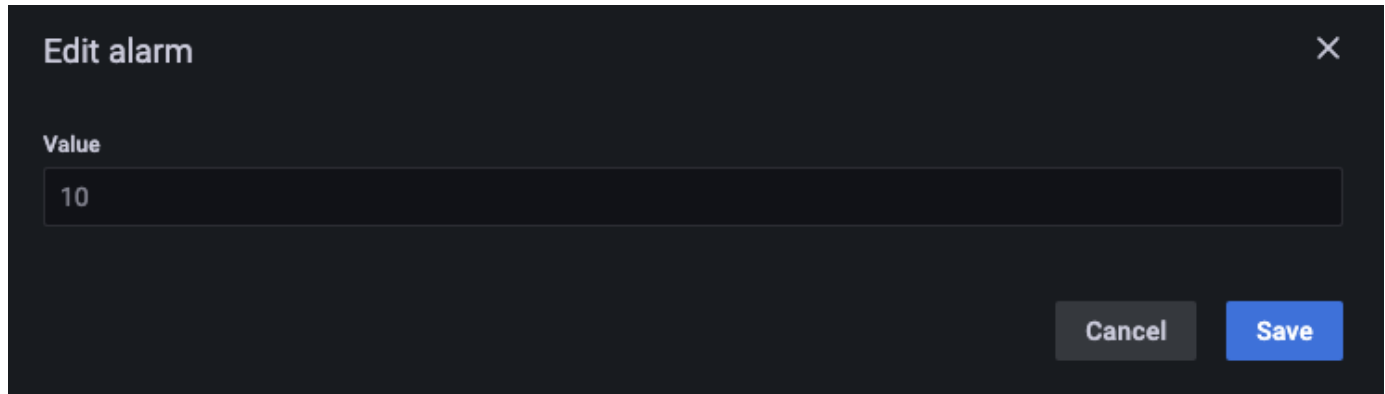
Per l'anteprima pubblica, non viene mostrata alcuna notifica. È necessario controllare lo stato e la soglia dell'allarme per assicurarsi che le proprietà siano state applicate correttamente.

7. È necessario utilizzare il Query Order of Ascending predefinito in modo che venga visualizzato il valore più recente.
8. La sezione dei filtri della Query può essere lasciata vuota. Di seguito è illustrata una configurazione completa:

The screenshot displays the Grafana Labs Alarm Configuration interface. The main panel shows the 'Query' configuration for 'GrafanaLabsAlarmTest'. The query type is 'Get Property Value History by Entity', the entity is 'ReactorCore', and the component name is 'OverheatAlarm'. Selected properties are 'alarm\_status (STRING)' and 'alarm\_threshold (DOUBLE)'. The filter section is empty. The right sidebar shows the 'Panel options' and 'AWS IoT TwinMaker Alarm Configuration' settings.

9. Utilizzando il pulsante Modifica allarme è possibile visualizzare una finestra di dialogo per modificare la soglia di allarme corrente.

## 10. Seleziona Salva per impostare il nuovo valore di soglia.



The screenshot shows a dark-themed dialog box titled "Edit alarm". In the top right corner, there is a close button represented by an "X" icon. Below the title, the word "Value" is displayed above a text input field that contains the number "10". At the bottom right of the dialog, there are two buttons: a grey "Cancel" button and a blue "Save" button.

### Note

Questo pannello deve essere usato solo con un intervallo di tempo reale che includa il presente. Usarlo con intervalli di tempo che terminano e iniziano nel passato può mostrare valori imprevisti quando si modificano le soglie di allarme come sempre la soglia corrente.

# AWS IoT TwinMaker Integrazione con Matterport

Matterport offre una varietà di opzioni di acquisizione per scansionare ambienti reali e creare modelli 3D coinvolgenti, noti anche come gemelli digitali Matterport. Questi modelli sono chiamati spazi Matterport. AWS IoT TwinMaker supporta l'integrazione con Matterport, che consente di importare i gemelli digitali Matterport nelle scene. AWS IoT TwinMaker Associando i gemelli digitali Matterport a AWS IoT TwinMaker, puoi visualizzare e monitorare il tuo sistema digital twin in un ambiente virtuale.



[Per ulteriori informazioni sull'uso di Matterport, leggi la documentazione di Matterport sulla pagina e Matterport.AWS IoT TwinMaker](#)

## Argomenti di integrazione

- [Panoramica dell'integrazione](#)
- [Prerequisiti di integrazione con Matterport](#)
- [Genera e registra le tue credenziali Matterport](#)
- [Memorizza le tue credenziali Matterport in AWS Secrets Manager](#)
- [Importa gli spazi Matterport nelle scene AWS IoT TwinMaker](#)
- [Usa gli spazi Matterport nella dashboard di Grafana AWS IoT TwinMaker](#)
- [Usa gli spazi Matterport nella tua applicazione web AWS IoT TwinMaker](#)



# Panoramica dell'integrazione

Questa integrazione consente di effettuare le seguenti operazioni:

- Usa i tag e gli spazi Matterport nel kit dell' AWS IoT TwinMaker app.
- Visualizza i dati matterport importati nella dashboard Grafana AWS IoT TwinMaker . Per ulteriori informazioni sull'utilizzo di Grafana AWS IoT TwinMaker e Grafana, leggi la documentazione sull'integrazione della dashboard [Grafana](#).
- Importa i tuoi spazi Matterport nelle tue scene. AWS IoT TwinMaker
- Seleziona e importa i tag Matterport che desideri associare ai dati nella tua scena. AWS IoT TwinMaker
- Evidenzia automaticamente lo spazio Matterport, contrassegna le modifiche nella AWS IoT TwinMaker scena e approva quali sincronizzare.

Il processo di integrazione è composto da 3 fasi fondamentali.

1. [Genera e registra le tue credenziali Matterport](#)
2. [Memorizza le tue credenziali Matterport in AWS Secrets Manager](#)
3. [Importa gli spazi Matterport nelle scene AWS IoT TwinMaker](#)

Inizi l'integrazione nella [AWS IoT TwinMaker console](#). Nella pagina Impostazioni della console, in Risorse di terze parti, apri l'integrazione con Matterport per navigare tra le diverse risorse richieste per l'integrazione.

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' is expanded. It 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 'AWSIoTtwinMaker\_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.

## Prerequisiti di integrazione con Matterport

Prima di integrare Matterport con, assicurati di AWS IoT TwinMaker soddisfare i seguenti prerequisiti:

- Hai acquistato un account Matterport di livello aziendale e i prodotti [Matterport necessari](#) per l'integrazione. AWS IoT TwinMaker
- Hai AWS IoT TwinMaker uno spazio di lavoro. Per ulteriori informazioni, consulta [Guida introduttiva AWS IoT TwinMaker](#).

- Hai aggiornato il tuo ruolo nell' AWS IoT TwinMaker area di lavoro. Per ulteriori informazioni sulla creazione di un ruolo nell'area di lavoro, consulta [Creare e gestire un ruolo di servizio](#) per. AWS IoT TwinMaker

Aggiungi quanto segue al tuo ruolo nell'area di lavoro:

```
{
  "Effect": "Allow",
  "Action": "secretsmanager:GetSecretValue",
  "Resource": [
    "AWS Secrets Manager secret ARN"
  ]
}
```

- È necessario contattare Matterport per configurare le licenze necessarie per abilitare l'integrazione. Matterport abiliterà anche un Private Model Embed (PME) per l'integrazione.

Se hai già un account manager Matterport, contattalo direttamente.

Utilizza la seguente procedura per contattare Matterport e richiedere un'integrazione se non disponi di un punto di contatto Matterport:

1. [Apri Matterport and page. AWS IoT TwinMaker](#)
2. Premi il pulsante Contattaci per aprire il modulo di contatto.
3. Compila le informazioni richieste nel modulo.
4. Quando sei pronto, scegli SAY HELLO per inviare la tua richiesta a Matterport.

Dopo aver richiesto l'integrazione, puoi generare le credenziali Matterport SDK e Private Model Embed (PME) richieste per continuare il processo di integrazione.

#### Note

Ciò potrebbe comportare il pagamento di una commissione per l'acquisto di nuovi prodotti o servizi.

# Genera e registra le tue credenziali Matterport

Per integrare Matterport con AWS IoT TwinMaker, è necessario fornire le credenziali Matterport. AWS Secrets Manager Utilizzare la seguente procedura per generare le credenziali Matterport SDK.

1. [Accedi al tuo account Matterport.](#)
2. Vai alla pagina delle impostazioni del tuo account.
3. Una volta nella pagina delle impostazioni, seleziona l'opzione Strumenti per sviluppatori.
4. Nella pagina degli strumenti per sviluppatori, vai alla sezione Gestione delle chiavi SDK.
5. Una volta nella sezione Gestione delle chiavi SDK, seleziona l'opzione per aggiungere una nuova chiave SDK.
6. Una volta ottenuta la chiave Matterport SDK, aggiungi i domini alla chiave for e al AWS IoT TwinMaker tuo server Grafana. Se utilizzi l' AWS IoT TwinMaker app kit, assicurati di aggiungere anche il tuo dominio personalizzato.
7. Successivamente, trova la sezione Gestione dell'integrazione delle applicazioni, dovresti vedere la tua applicazione PME nell'elenco. Registra le seguenti informazioni:
  - L'ID del cliente
  - Il segreto del cliente

## Note

Poiché il Client Secret ti viene presentato solo una volta, ti consigliamo vivamente di registrarlo. Devi presentare il tuo Client Secret nella AWS Secrets Manager console per continuare con l'integrazione con Matterport.

Queste credenziali vengono create automaticamente quando hai acquistato i componenti necessari e il PME per il tuo account è stato abilitato da Matterport. Se queste credenziali non vengono visualizzate, contatta Matterport. Per richiedere un contatto, consulta il modulo [Matterport](#) e contatti. AWS IoT TwinMaker

[Per ulteriori informazioni sulle credenziali di Matterport SDK, consultate la documentazione SDK ufficiale di Matterport SDK Docs Overview.](#)

# Memorizza le tue credenziali Matterport in AWS Secrets Manager

Utilizza la seguente procedura per memorizzare le tue credenziali Matterport in AWS Secrets Manager

## Note

È necessario il Client Secret creato dalla procedura riportata nell'[Genera e registra le tue credenziali Matterport](#) argomento per continuare con l'integrazione con Matterport.

1. Accedi alla console. AWS Secrets Manager
2. Vai alla pagina Segreti e seleziona Memorizza un nuovo segreto.
3. Per il tipo di segreto, seleziona Altro tipo di segreto.
4. Nella sezione Coppie chiave/valore, aggiungi le seguenti coppie chiave-valore, utilizzando le tue credenziali Matterport come valori:
  - Crea una coppia chiave-valore, con Key: e Value: **application\_key** <your Matterport credentials>
  - Crea una coppia chiave-valore, con Key: e Value: **client\_id**. <your Matterport credentials>
  - Crea una coppia chiave-valore, con Key: e Value: **client\_secret**. <your Matterport credentials>

Una volta completata, dovresti avere una configurazione simile all'esempio seguente:

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. Per la chiave di crittografia, è possibile lasciare selezionata la chiave di crittografia predefinita. `aws/secretsmanager`
6. Scegli Avanti per passare alla pagina Configura segreta.
7. Compila il campo Nome segreto e Descrizione.
8. Aggiungi un tag a questo segreto nella sezione Tag.

Quando crei il tag, assegna la chiave `AWSIoTtwinMaker_Matterport` come mostrato nella schermata seguente:

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="AWSIoTwinMaker_Matterport"/>	<input type="text" value="Enter value"/>	<input type="button" value="Remove"/>
<input type="button" value="Add"/>		

 **Note**

È necessario aggiungere un tag. I tag sono obbligatori quando si aggiungono segreti di terze parti AWS Secrets Manager, nonostante i tag siano elencati come opzionali.

Il campo Valore è facoltativo. Dopo aver fornito una chiave, puoi selezionare Aggiungi per passare al passaggio successivo.

- Scegli Avanti per passare alla pagina Configura rotazione. L'impostazione di una rotazione segreta è facoltativa. Se desideri completare l'aggiunta del tuo segreto e non hai bisogno di una rotazione, scegli di nuovo Avanti. Per ulteriori informazioni sulla rotazione segreta, consulta [Ruotare AWS Secrets Manager i segreti](#).
- Conferma la configurazione segreta nella pagina Revisione. Quando sei pronto per aggiungere il tuo segreto, scegli Store.

Per ulteriori informazioni sull'utilizzo AWS Secrets Manager, consulta la seguente AWS Secrets Manager documentazione:

- [Crea e gestisci segreti con AWS Secrets Manager](#)
- [Che cos'è AWS Secrets Manager?](#)
- [Ruota i segreti AWS Secrets Manager](#)

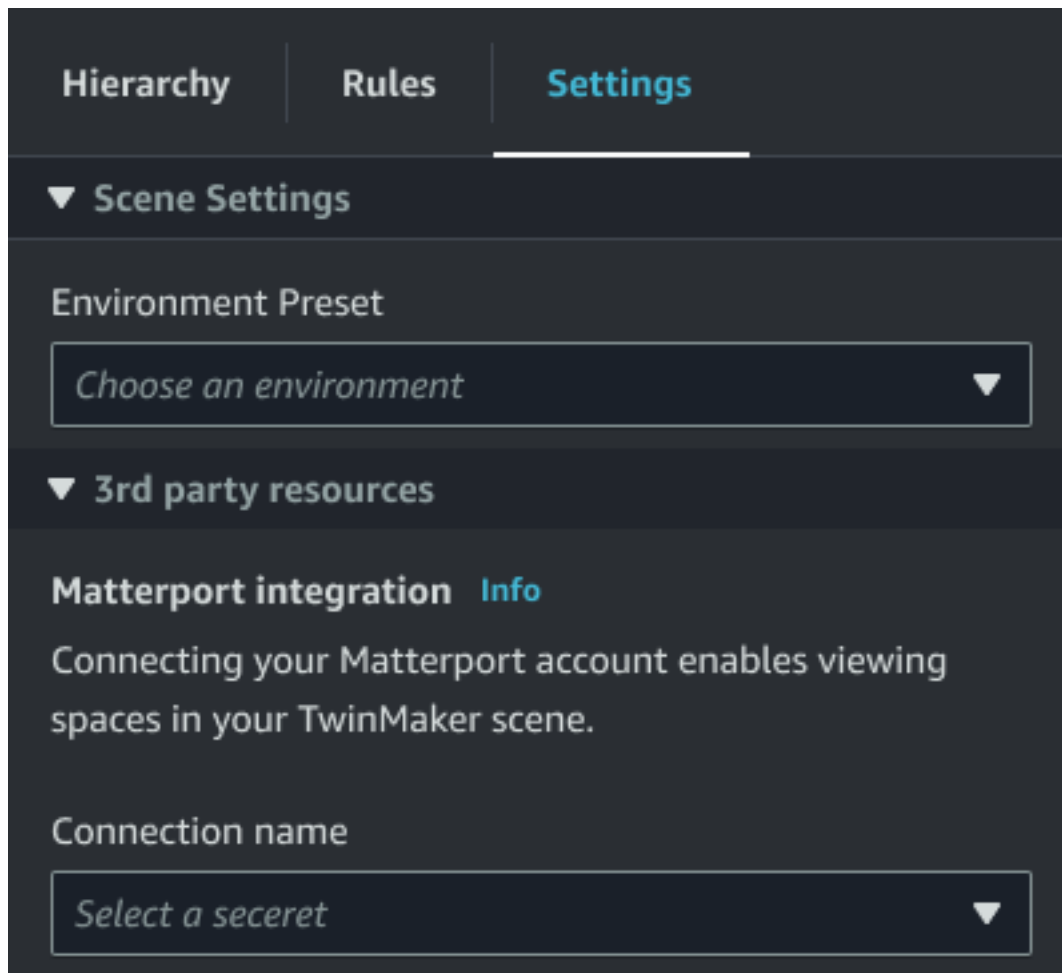
Ora sei pronto per importare le tue risorse Matterport nelle scene. AWS IoT TwinMaker Consultate la procedura riportata nella sezione seguente, [Importa gli spazi Matterport nelle scene AWS IoT TwinMaker](#)

## Importa gli spazi Matterport nelle scene AWS IoT TwinMaker

Aggiungi scansioni Matterport alla tua scena selezionando l'account Matterport collegato dalla pagina delle impostazioni della scena. Utilizzate la seguente procedura per importare le scansioni e i tag Matterport:

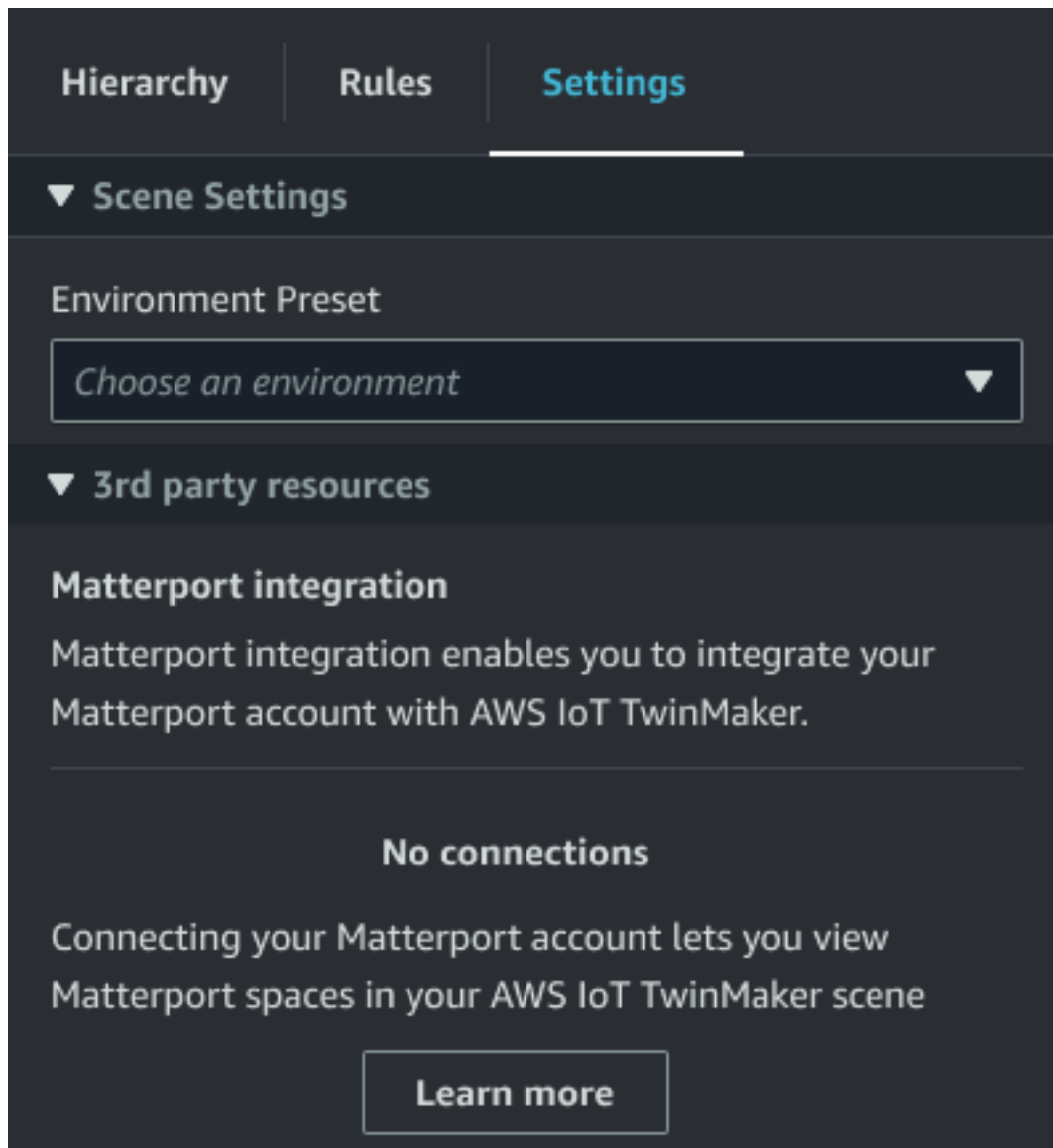
1. Accedi alla console di [AWS IoT TwinMaker](#).
2. Crea o apri una AWS IoT TwinMaker scena esistente in cui desideri utilizzare uno spazio Matterport.
3. Una volta aperta la scena, vai alla scheda Impostazioni.
4. In Impostazioni, in Risorse di terze parti, trova il nome della connessione e inserisci il segreto che hai creato durante la procedura da [Memorizza le tue credenziali Matterport in AWS Secrets Manager](#).



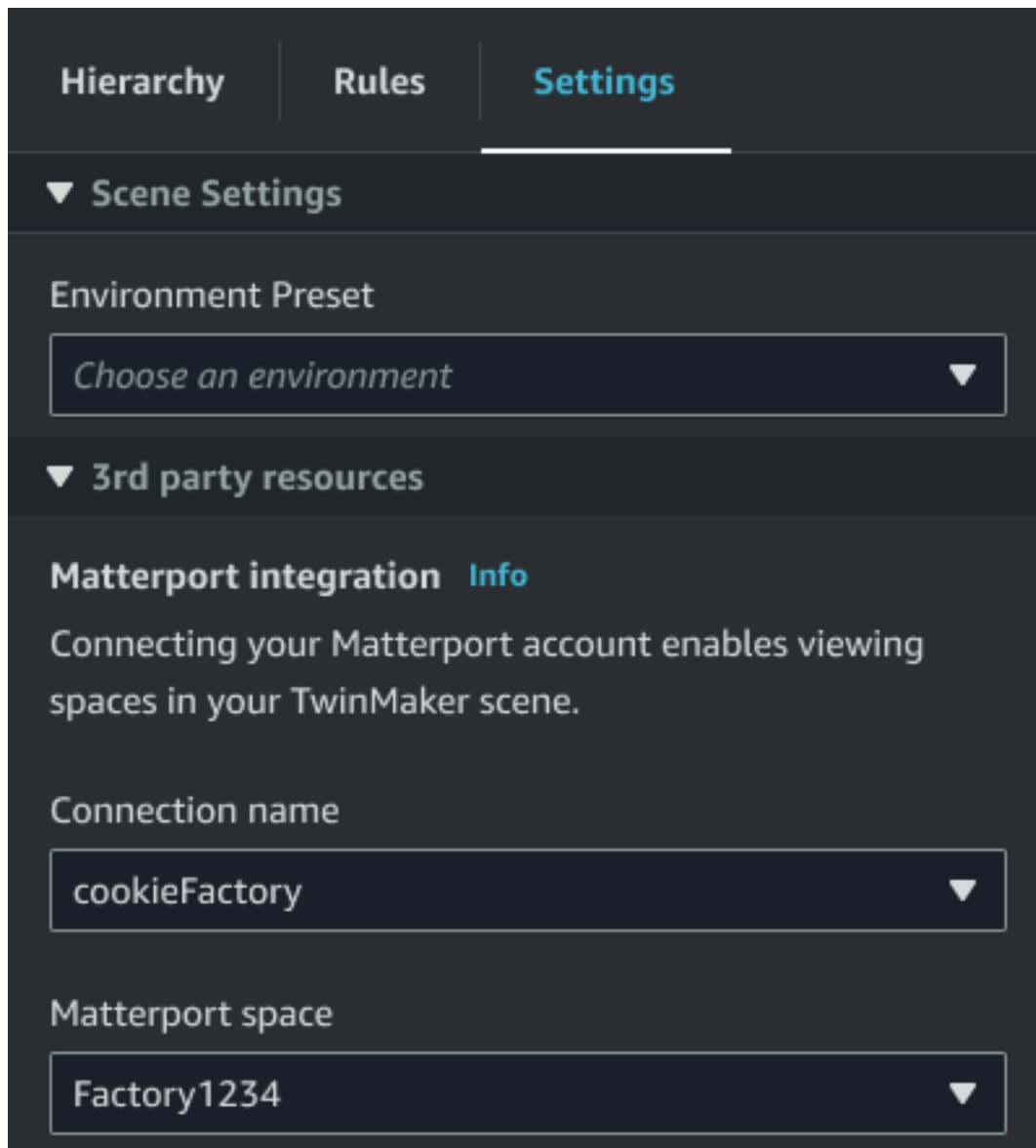


**Note**

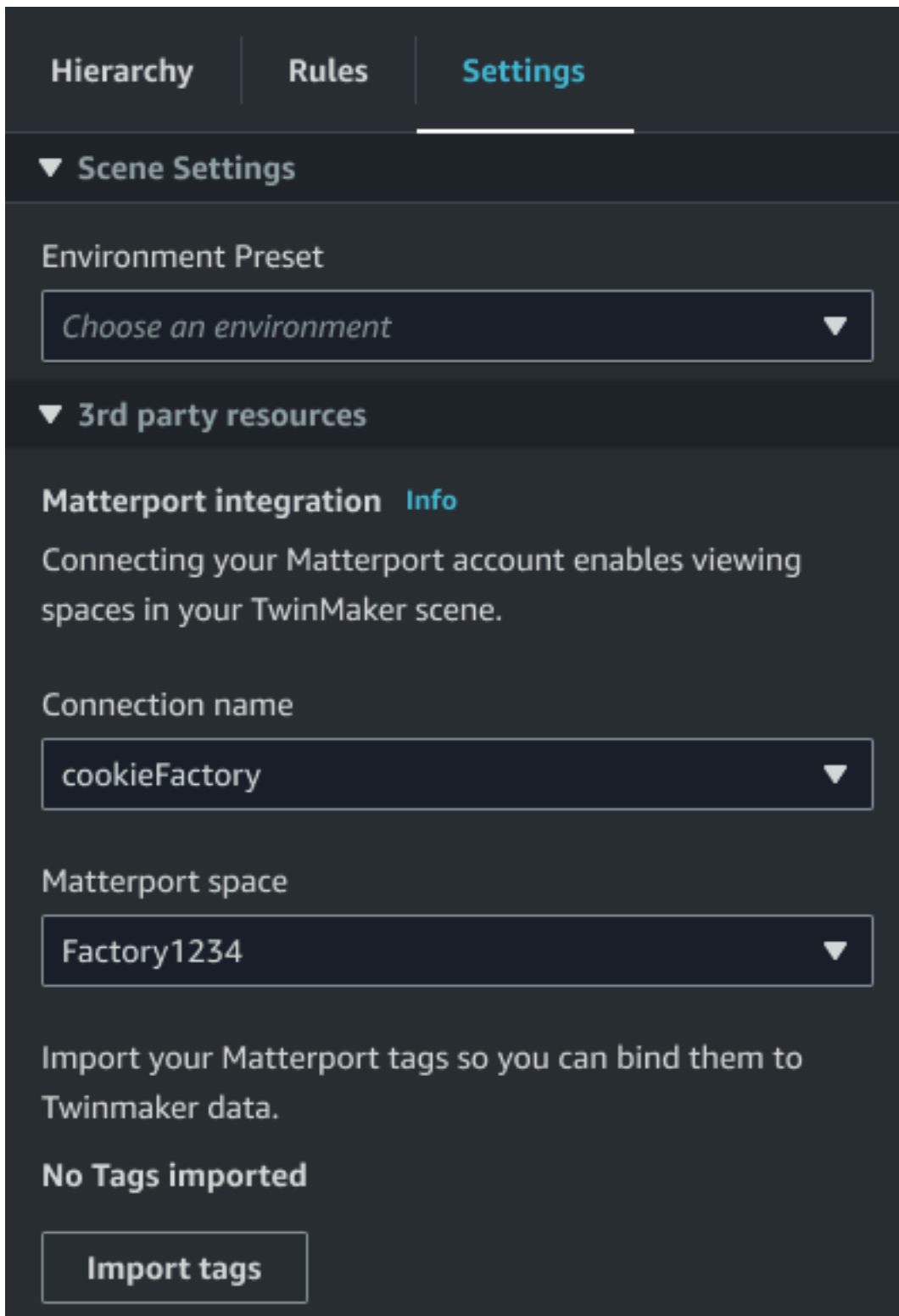
Se viene visualizzato un messaggio che indica Nessuna connessione, accedi alla pagina delle impostazioni della [AWS IoT TwinMaker console](#) per iniziare il processo di integrazione con Matterport.



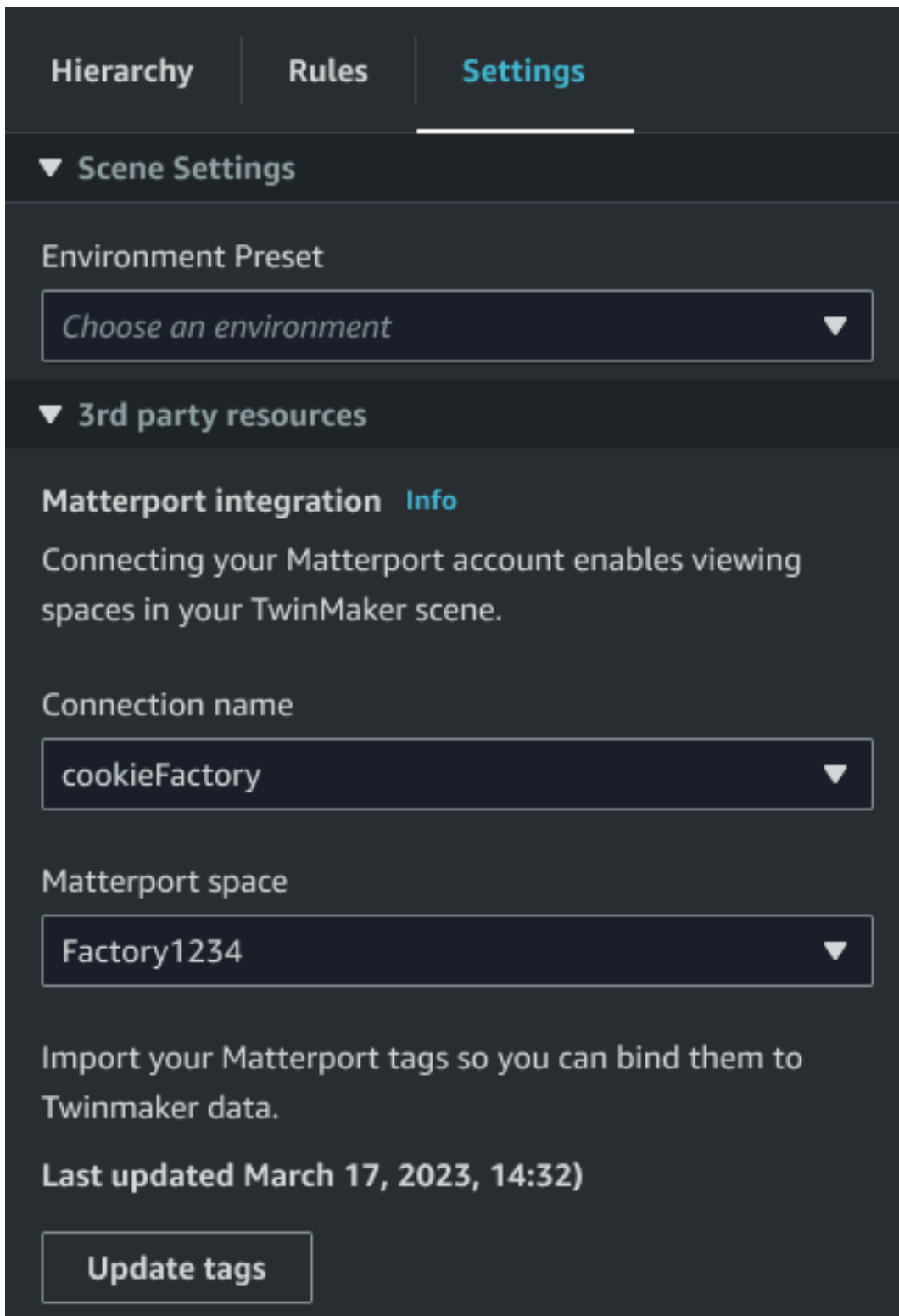
5. Quindi, scegli lo spazio Matterport che desideri utilizzare nella tua scena selezionandolo nel menu a discesa dello spazio Matterport.



6. Dopo aver selezionato uno spazio, puoi importare i tag Matterport e convertirli in tag di AWS IoT TwinMaker scena premendo il pulsante Importa tag.



Dopo aver importato i tag Matterport, il pulsante viene sostituito dal pulsante **Aggiorna tag**. Puoi aggiornare continuamente i tag Matterport in AWS IoT TwinMaker modo che riflettano sempre le modifiche più recenti nel tuo account Matterport.



7. Ti sei integrato con successo AWS IoT TwinMaker con Matterport e ora la tua AWS IoT TwinMaker scena ha sia lo spazio Matterport che i tag importati. Puoi lavorare all'interno di questa scena come faresti con qualsiasi altra scena. AWS IoT TwinMaker

Per ulteriori informazioni sull'utilizzo delle AWS IoT TwinMaker scene, consultate [Creazione e modifica di AWS IoT TwinMaker scene](#).

## Usa gli spazi Matterport nella dashboard di Grafana AWS IoT TwinMaker

Dopo aver importato lo spazio Matterport in una AWS IoT TwinMaker scena, puoi visualizzare quella scena con lo spazio Matterport nella dashboard Grafana. Se hai già configurato Grafana con AWS IoT TwinMaker, puoi semplicemente aprire la dashboard Grafana per visualizzare la tua scena con lo spazio Matterport importato.

Se non hai ancora AWS IoT TwinMaker configurato Grafana, completa prima il processo di integrazione di Grafana. Hai due scelte per l'integrazione AWS IoT TwinMaker con Grafana. Puoi usare un'istanza Grafana autogestita o Amazon Managed Grafana.

Consulta la seguente documentazione per saperne di più sulle opzioni e sul processo di integrazione di Grafana:

- [AWS IoT TwinMaker Integrazione della dashboard Grafana](#).
- [Grafana gestita da Amazon](#).
- [Grafana autogestita](#).

## Usa gli spazi Matterport nella tua applicazione web AWS IoT TwinMaker

Dopo aver importato lo spazio Matterport in una AWS IoT TwinMaker scena, puoi visualizzare quella scena con lo spazio Matterport nell'applicazione web dell'app kit. AWS IoT

Consulta la seguente documentazione per saperne di più sull'utilizzo del kit applicativo: AWS IoT

- Per ulteriori informazioni sull'utilizzo AWS IoT TwinMaker con l' AWS IoT app kit, consulta [Crea un'applicazione web personalizzata utilizzando AWS IoT TwinMaker i componenti dell'interfaccia utente](#).
- Per saperne di più sull'utilizzo AWS IoT dell'application kit, visita la pagina [Github AWS IoT dell'Application kit](#).
- Per istruzioni su come avviare una nuova applicazione Web utilizzando l' AWS IoT application kit, visita la pagina della documentazione ufficiale [dell'IoT App Kit](#).

# AWS IoT TwinMaker integrazione video

Le videocamere rappresentano una buona opportunità per la simulazione di gemelli digitali. È possibile utilizzarla AWS IoT TwinMaker per simulare la posizione e le condizioni fisiche della fotocamera. Crea entità AWS IoT TwinMaker per le tue telecamere in loco e usa i componenti video per trasmettere video e metadati in diretta dal tuo sito alla tua AWS IoT TwinMaker scena o a una dashboard Grafana.

AWS IoT TwinMaker può acquisire video da dispositivi periferici in due modi. Puoi trasmettere video da dispositivi periferici con il connettore edge per lo streaming video Kinesis oppure puoi salvare video sul dispositivo periferico e avviare il caricamento dei video con messaggi MQTT. Usa questo componente per trasmettere dati video dai tuoi dispositivi per utilizzarli con i servizi. AWS IoT Per generare le risorse necessarie e implementare il connettore perimetrale per Kinesis Video Streams, consulta la [Guida introduttiva al connettore perimetrale per Kinesis Video Streams on. GitHub](#) Per ulteriori informazioni sul AWS IoT Greengrass componente, consulta la AWS IoT Greengrass documentazione sul [connettore edge per Kinesis Video Streams](#).

Dopo aver creato i AWS IoT SiteWise modelli richiesti e configurato il componente Kinesis Video Streams Greengrass, puoi trasmettere o registrare video sull'edge verso la tua applicazione digital twin nella console. AWS IoT TwinMaker Puoi anche visualizzare i live streaming e i metadati dai tuoi dispositivi in una dashboard Grafana. Per ulteriori informazioni sull'integrazione di Grafana AWS IoT TwinMaker e, vedere. [AWS IoT TwinMaker Integrazione della dashboard Grafana](#)

## Usa il connettore edge per Kinesis Video Stream per lo streaming di video AWS IoT TwinMaker

Con il connettore edge per Kinesis video Stream, puoi trasmettere video e dati a un'entità nella tua AWS IoT TwinMaker scena. A tale scopo, utilizzi un componente video. Per creare il componente video da utilizzare nelle scene, completate la procedura seguente.

### Prerequisiti

Prima di creare il componente video nella AWS IoT TwinMaker scena, assicuratevi di aver completato i seguenti prerequisiti.

- Ha creato i AWS IoT SiteWise modelli e le risorse richiesti per il connettore edge per lo streaming video Kinesis. Per ulteriori informazioni sulla creazione delle AWS IoT SiteWise risorse per il connettore, consulta [Guida introduttiva al connettore edge per Kinesis video Stream](#).
- Hai implementato il connettore Kinesis Video Stream Edge sul AWS IoT Greengrass tuo dispositivo. [Per ulteriori informazioni sulla distribuzione del componente Kinesis Video Stream Edge Connector, consulta il README sulla distribuzione.](#)

## Crea componenti video per le scene AWS IoT TwinMaker

Completa i seguenti passaggi per creare il connettore edge per il componente di streaming video Kinesis per la tua scena.

1. Nella AWS IoT TwinMaker console, apri la scena a cui desideri aggiungere il componente video.
2. Dopo l'apertura della scena, scegli un'entità esistente o crea l'entità a cui desideri aggiungere il componente, quindi scegli Aggiungi componente.
3. Nel riquadro Aggiungi componente, inserisci un nome per il componente e, per Tipo, scegli `com.amazon.iotsitewise.connector.edgevideo`.
4. Scegli un modello di AWS IoT SiteWise risorsa selezionando il nome del modello di fotocamera che hai creato. Questo nome deve avere il seguente formato: `EdgeConnectorForKVS_CameraModel-0abc`, dove la stringa di lettere e numeri alla fine corrisponde al nome della risorsa.
5. Per Asset, scegliete le risorse AWS IoT SiteWise della videocamera da cui desiderate effettuare lo streaming video. Viene visualizzata una piccola finestra che mostra un'anteprima del flusso video corrente.

### Note

Per testare lo streaming video, scegli test. Questo test invia un evento MQTT per avviare lo streaming video in diretta. Attendi qualche istante per vedere il video apparire nel lettore.

6. Per aggiungere il componente video alla tua entità, scegli Aggiungi componente.



# Aggiungi video e metadati dallo stream video di Kinesis a una dashboard Grafana

Dopo aver creato un componente video per la tua entità nella AWS IoT TwinMaker scena, puoi configurare il pannello video in Grafana per vedere i live streaming. Assicurati di averlo integrato correttamente AWS IoT TwinMaker con Grafana. Per ulteriori informazioni, consulta [AWS IoT TwinMakerIntegrazione della dashboard Grafana](#).

## Important

Per visualizzare i video nella dashboard Grafana, devi assicurarti che le origini dati Grafana dispongano delle autorizzazioni IAM appropriate. Per creare il ruolo e la policy richiedi, consulta. [Creazione di un ruolo IAM nella dashboard](#)

Completa i seguenti passaggi per visualizzare Kinesis Video Streams e i metadati nella dashboard di Grafana.

1. Apri il pannello di controllo AWS IoT TwinMaker.
2. Scegli Aggiungi pannello, quindi scegli Aggiungi un pannello vuoto.
3. Dall'elenco dei pannelli, scegli il pannello del lettore AWS IoT TwinMaker video.
4. Nel pannello del lettore AWS IoT TwinMaker video, inserisci il nome dello stream KinesisVideoStreamName, con il nome dello stream video Kinesis da cui desideri eseguire lo streaming video.

## Note

Per trasmettere i metadati al pannello video Grafana, devi prima aver creato un'entità con un componente di streaming video.

5. Facoltativo: per trasmettere i metadati dalle AWS IoT SiteWise risorse al lettore video, per Entità, scegli l'AWS IoT TwinMakerentità che hai creato nella scena. AWS IoT TwinMaker Per il nome del componente, scegliete il componente video che avete creato per l'entità nella AWS IoT TwinMaker scena.

# Usare la AWS IoT TwinMaker libreria Flink

AWS IoT TwinMaker fornisce una libreria Flink che puoi usare per leggere e scrivere dati su archivi dati esterni utilizzati nei tuoi gemelli digitali.

È possibile utilizzare la libreria AWS IoT TwinMaker Flink installandola come connettore personalizzato in Managed Service for Apache Flink ed eseguendo query SQL Flink in un notebook Zeppelin in Managed Service for Apache Flink. Il notebook può essere promosso a un'applicazione di elaborazione di flussi in esecuzione continua. La libreria sfrutta i AWS IoT TwinMaker componenti per recuperare i dati dall'area di lavoro.

La libreria AWS IoT TwinMaker Flink richiede quanto segue.

## Prerequisiti

1. Uno spazio di lavoro completamente popolato con scene e componenti. Utilizza i tipi di componenti integrati per i dati provenienti dai AWS servizi (AWS IoT SiteWise da Kinesis Video Streams). Crea tipi di componenti personalizzati per i dati provenienti da fonti di terze parti. Per ulteriori informazioni, consulta [???](#).
2. Comprensione dei notebook Studio con Managed Service for Apache Flink for Apache Flink. [Questi notebook sono basati su Apache Zeppelin e utilizzano il framework Apache Flink.](#) Per ulteriori informazioni, consulta [Utilizzo di un notebook Studio con Managed Service for Apache Flink for Apache Flink.](#)

Per istruzioni sull'uso della libreria, consultate la guida per l'utente della libreria [AWS IoT TwinMakerFlink](#).

Per istruzioni sulla configurazione AWS IoT TwinMaker con l'avvio rapido negli [AWS IoT TwinMaker esempi](#), consultate il [file README per l'applicazione sample insights](#).

# Registrazione e monitoraggio in AWS IoT TwinMaker

Il monitoraggio è una parte importante per mantenere l'affidabilità, la disponibilità e le prestazioni di AWS IoT TwinMaker e delle altre soluzioni AWS. AWS IoT TwinMaker supporta i seguenti strumenti di monitoraggio per controllare il servizio, segnalare eventuali problemi ed eseguire operazioni automatiche quando appropriato:

- Amazon CloudWatch monitora in tempo reale le AWS risorse e le applicazioni che esegui su AWS. È possibile raccogliere i parametri e tenerne traccia, creare pannelli di controllo personalizzati e impostare allarmi che inviino una notifica o intraprendano azioni quando un parametro specificato raggiunge una determinata soglia. Ad esempio, puoi impostare perché CloudWatch tenga traccia dell'uso della CPU o di altri parametri per le tue istanze Amazon EC2 e avviare automaticamente nuove istanze quando necessario. Per ulteriori informazioni, consulta la [Guida per CloudWatch l'utente di Amazon](#).
- Amazon CloudWatch Logs monitora, archivia e fornisce l'accesso ai tuoi file di registro da AWS IoT TwinMaker gateway e altre fonti. CloudTrail CloudWatch I log consentono di monitorare le informazioni nei file di log e notificare quando vengono raggiunte determinate soglie. Puoi inoltre archiviare i dati del log in storage estremamente durevole. Per ulteriori informazioni, consulta la [Guida per l'utente CloudWatch di Amazon Logs](#).
- AWS CloudTrail acquisisce le chiamate API e gli eventi correlati effettuati da o per conto del tuo account AWS e fornisce i file di log a un bucket Amazon S3 specificato. Puoi identificare quali utenti e account hanno richiamato AWS, l'indirizzo IP di origine da cui sono state effettuate le chiamate e quando sono avvenute. Per ulteriori informazioni, consultare la [Guida per l'utente AWS CloudTrail](#).

## Argomenti

- [Monitoraggio AWS IoT TwinMaker con CloudWatch metriche Amazon](#)
- [Registrazione delle chiamate API AWS IoT TwinMaker con AWS CloudTrail](#)

## Monitoraggio AWS IoT TwinMaker con CloudWatch metriche Amazon

È possibile monitorare AWS IoT TwinMaker utilizzando CloudWatch, che raccoglie dati grezzi e li elabora in metriche leggibili e quasi in tempo reale. Queste statistiche vengono conservate per

un periodo di 15 mesi, per permettere l'accesso alle informazioni storiche e offrire una prospettiva migliore sulle prestazioni del servizio o dell'applicazione Web. È anche possibile impostare allarmi che controllano determinate soglie e inviare notifiche o intraprendere azioni quando queste soglie vengono raggiunte. Per ulteriori informazioni, consulta la [Guida per CloudWatch l'utente di Amazon](#).

AWS IoT TwinMaker pubblica le metriche e le dimensioni elencate nelle sezioni seguenti nel `AWS/IoT TwinMaker` namespace.

### Tip

AWS IoT TwinMaker pubblica i parametri su un intervallo di un minuto. Quando visualizzi queste metriche nei grafici della CloudWatch console, ti consigliamo di scegliere un periodo di 1 minuto per visualizzare la massima risoluzione disponibile dei dati metrici.

## Indice

- [Parametri](#)

## Parametri

AWS IoT TwinMaker pubblica i seguenti parametri.

### Parametri

Parametro	Descrizione
<code>ComponentTypeCreationFailure</code>	<p>Questa metrica indica se la creazione del tipo di componente ha avuto esito positivo.</p> <p>La metrica viene pubblicata quando un tipo di componente è nello <code>CREATING</code> stato. Ciò accade quando viene creato un tipo di componente con le proprietà richieste nell'inizializzatore dello schema e queste proprietà vengono istanziate con valori predefiniti.</p> <p>Il valore della metrica può indicare <code>0</code> il successo o <code>1</code> il fallimento.</p>

Parametro	Descrizione
	<p>Dimensioni: ComponentTypeId, WorkspaceId.</p> <p>Unità: Numero</p>
ComponentTypeUpdateFailure	<p>Questa metrica indica se l'aggiornamento del tipo di componente ha avuto successo.</p> <p>La metrica viene pubblicata quando un tipo di componente è nelloUPDATING stato. Ciò accade quando un tipo di componente viene aggiornato con le proprietà richieste nell'inizializzatore dello schema e queste proprietà vengono istanziate con valori predefiniti.</p> <p>Il valore della metrica può indicare 0 il successo o 1 il fallimento.</p> <p>Dimensioni: ComponentTypeId, WorkspaceId.</p> <p>Unità: Numero</p>
EntityCreationFailure	<p>Questa metrica indica se la creazione dell'entità ha avuto successo. La metrica viene pubblicata quando un'entità è nelloCREATING stato. Ciò accade quando un'entità viene creata con un componente.</p> <p>Il valore della metrica può indicare 0 il successo o 1 il fallimento.</p> <p>Dimensioni: EntityName, EntityId, WorkspaceId.</p> <p>Unità: Numero</p>

Parametro	Descrizione
EntityUpdateFailure	<p>Questa metrica indica se l'aggiornamento dell'entità ha avuto successo. La metrica viene pubblicata quando un'entità è nelloUPDATING stato. Ciò accade quando un'entità viene aggiornata.</p> <p>Il valore della metrica può indicare 0 il successo o 1 il fallimento.</p> <p>Dimensioni: EntityName, EntityId, Workspace Id.</p> <p>Unità: Numero</p>
EntityDeletionFailure	<p>Questa metrica indica se l'eliminazione dell'entità ha avuto successo. La metrica viene pubblicata quando un'entità è nelloDELETING stato. Ciò accade quando un'entità viene eliminata.</p> <p>Il valore della metrica può indicare 0 il successo o 1 il fallimento.</p> <p>Dimensioni: EntityName, EntityId, Workspace Id.</p> <p>Unità: Numero</p>

 Tip

Tutte le metriche vengono pubblicateAWS/IoTTwinMaker nello spazio dei nomi.

# Registrazione delle chiamate API AWS IoT TwinMaker con AWS CloudTrail

AWS IoT TwinMaker è integrato con AWS CloudTrail, un servizio che offre un registro delle operazioni eseguite da un utente, un ruolo o un AWS servizio in AWS IoT TwinMaker. CloudTrail acquisisce le chiamate API AWS IoT TwinMaker come eventi. Le chiamate acquisite includono le chiamate dalla console di AWS IoT TwinMaker e le chiamate di codice alle operazioni delle API AWS IoT TwinMaker. Se si crea un trail, è possibile abilitare la distribuzione continua di CloudTrail eventi in un bucket Amazon S3, inclusi gli eventi per AWS IoT TwinMaker. Se non configuri un trail, puoi comunque visualizzare gli eventi più recenti nella CloudTrail console di in Event history (Cronologia eventi). Le informazioni raccolte da CloudTrail, consentono di determinare la richiesta effettuata a AWS IoT TwinMaker, l'indirizzo IP di origine da cui è stata eseguita la richiesta, l'autore della richiesta, il momento in cui è stata eseguita e altri dettagli.

Per ulteriori informazioni CloudTrail, consulta la [Guida per AWS CloudTrail l'utente](#).

## AWS IoT TwinMaker informazioni in CloudTrail

Quando crei il tuo AWS account, CloudTrail viene abilitato automaticamente. CloudTrail i record supportano l'attività degli eventi che si verifica in AWS IoT TwinMaker, insieme ad altri eventi AWS di servizio nella cronologia degli eventi. È possibile visualizzare, cercare e scaricare gli eventi recenti nell'account AWS. Per ulteriori informazioni, consulta [Visualizzazione di eventi mediante la cronologia CloudTrail eventi](#) di.

Per una registrazione continua degli eventi nell'account AWS che includa gli eventi per AWS IoT TwinMaker, creare un trail. Un trail consente di CloudTrail distribuire i file di log in un bucket Amazon S3. Per impostazione di default, quando crei un trail nella console, il trail sarà valido in tutte le AWS regioni. CloudTrail registra gli eventi di tutte le Regioni nella AWS partizione e distribuisce i file di log nel bucket Amazon S3 specificato. Inoltre, è possibile configurare altri AWS servizi per analizzare con maggiore dettaglio e usare i dati raccolti nei CloudTrail log. Per ulteriori informazioni, consulta gli argomenti seguenti:

- [Panoramica della creazione di un percorso](#)
- [CloudTrail servizi e integrazioni supportati](#)
- [Configurazione delle notifiche Amazon SNS per CloudTrail](#)
- [Ricezione di file di CloudTrail log da più Regioni](#) e [Ricezione di file di CloudTrail log da più account](#)

La maggior parte delle operazioni AWS IoT TwinMaker viene registrata da CloudTrail e documentata nell'[AWS IoT TwinMaker API Reference](#).

Le seguenti operazioni del piano dati non vengono registrate da CloudTrail:

- [GetPropertyValue](#)
- [GetPropertyValueHistory](#)
- [BatchPutPropertyValues](#)

Ogni evento o voce del log contiene informazioni sull'utente che ha generato la richiesta. Le informazioni di identità consentono di determinare quanto segue:

- Se la richiesta è stata effettuata con credenziali utente o root.
- Se la richiesta è stata effettuata con le credenziali di sicurezza temporanee per un ruolo o un utente federato.
- Se la richiesta è stata effettuata da un altro servizio AWS.

Per ulteriori informazioni, consulta [Elemento CloudTrail userIdentity](#).



# Sicurezza in AWS IoT TwinMaker

La sicurezza del cloud AWS è la massima priorità. In qualità di AWS cliente, puoi beneficiare di data center e architetture di rete progettati per soddisfare i requisiti delle organizzazioni più sensibili alla sicurezza.

La sicurezza è una responsabilità condivisa tra te e te. AWS Il [modello di responsabilità condivisa](#) descrive questo aspetto come sicurezza del cloud e sicurezza nel cloud:

- Sicurezza del cloud: AWS è responsabile della protezione dell'infrastruttura che gestisce AWS i servizi in Cloud AWS. AWS fornisce inoltre servizi che è possibile utilizzare in modo sicuro. I revisori esterni testano e verificano regolarmente l'efficacia della nostra sicurezza nell'ambito dei [AWS Programmi di AWS conformità dei Programmi di conformità](#) dei di . Per ulteriori informazioni sui programmi di conformità applicabili AWS IoT TwinMaker, consulta [AWS Servizi nell'ambito del programma di conformitàAWS](#) .
- Sicurezza nel cloud: la tua responsabilità è determinata dal AWS servizio che utilizzi. Sei anche responsabile di altri fattori, tra cui la riservatezza dei dati, i requisiti della tua azienda e le leggi e normative vigenti.

Questa documentazione ti aiuta a capire come applicare il modello di responsabilità condivisa durante l'utilizzo AWS IoT TwinMaker. I seguenti argomenti mostrano come eseguire la configurazione AWS IoT TwinMaker per soddisfare gli obiettivi di sicurezza e conformità. Imparerai anche a utilizzare altri AWS servizi che ti aiutano a monitorare e proteggere AWS IoT TwinMaker le tue risorse.

## Argomenti

- [Protezione dei dati in AWS IoT TwinMaker](#)
- [Identity and Access Management per AWS IoT TwinMaker](#)
- [AWS IoT TwinMaker e endpoint VPC di interfaccia \(\)AWS PrivateLink](#)
- [Convalida della conformità per AWS IoT TwinMaker](#)
- [Resilienza in AWS IoT TwinMaker](#)
- [Sicurezza dell'infrastruttura in AWS IoT TwinMaker](#)

# Protezione dei dati in AWS IoT TwinMaker

Il modello di [responsabilità AWS condivisa modello](#) di di si applica alla protezione dei dati in AWS IoT TwinMaker. Come descritto in questo modello, AWS è responsabile della protezione dell'infrastruttura globale che gestisce tutti i Cloud AWS. L'utente è responsabile del controllo dei contenuti ospitati su questa infrastruttura. Questi contenuti includono la configurazione della protezione e le attività di gestione per i servizi Servizi AWS utilizzati. Per ulteriori informazioni sulla privacy dei dati, vedi le [Domande frequenti sulla privacy dei dati](#). Per informazioni sulla protezione dei dati in Europa, consulta il post del blog relativo al [Modello di responsabilità condivisa AWS e GDPR](#) nel Blog sulla sicurezza AWS .

Ai fini della protezione dei dati, consigliamo di proteggere Account AWS le credenziali e configurare i singoli utenti con AWS IAM Identity Center or AWS Identity and Access Management (IAM). In tal modo, a ogni utente verranno assegnate solo le autorizzazioni necessarie per svolgere i suoi compiti. Ti suggeriamo, inoltre, di proteggere i dati nei seguenti modi:

- Utilizza l'autenticazione a più fattori (MFA) con ogni account.
- Usa SSL/TLS per comunicare con le risorse. AWS È richiesto TLS 1.2 ed è consigliato TLS 1.3.
- Configura l'API e la registrazione delle attività degli utenti con. AWS CloudTrail
- Utilizza soluzioni di AWS crittografia, insieme a tutti i controlli di sicurezza predefiniti all'interno Servizi AWS.
- Utilizza i servizi di sicurezza gestiti avanzati, come Amazon Macie, che aiutano a individuare e proteggere i dati sensibili archiviati in Amazon S3.
- Se hai bisogno di moduli crittografici convalidati FIPS 140-2 per l'accesso AWS tramite un'interfaccia a riga di comando o un'API, utilizza un endpoint FIPS. Per ulteriori informazioni sugli endpoint FIPS disponibili, consulta il [Federal Information Processing Standard \(FIPS\) 140-2](#).

Ti consigliamo vivamente di non inserire mai informazioni riservate o sensibili, ad esempio gli indirizzi e-mail dei clienti, nei tag o nei campi di testo in formato libero, ad esempio nel campo Nome. Ciò include quando lavori o Servizi AWS utilizzi la console, l'API AWS IoT TwinMaker o gli SDK. AWS CLI AWS I dati inseriti nei tag o nei campi di testo in formato libero utilizzati per i nomi possono essere utilizzati per i la fatturazione o i log di diagnostica. Quando fornisci un URL a un server esterno, ti suggeriamo vivamente di non includere informazioni sulle credenziali nell'URL per convalidare la tua richiesta al server.

## Crittografia dei dati a riposo

AWS IoT TwinMaker archivia le informazioni sull'area di lavoro in un bucket Amazon S3 che il servizio crea per te, se lo desideri. Il bucket creato dal servizio per te ha la crittografia lato server predefinita abilitata. Se scegli di utilizzare il tuo bucket Amazon S3 quando crei un nuovo spazio di lavoro, ti consigliamo di abilitare la crittografia lato server predefinita. Per ulteriori informazioni sulla crittografia predefinita in Amazon S3, consulta [Impostazione del comportamento di crittografia lato server predefinito per i bucket Amazon S3](#).

## Crittografia in transito

Tutti i dati inviati AWS IoT TwinMaker vengono inviati tramite una connessione TLS utilizzando il protocollo HTTPS, quindi sono sicuri per impostazione predefinita durante il transito.

### Note

Ti consigliamo di utilizzare HTTPS sugli indirizzi dei bucket Amazon S3 come controllo per applicare la crittografia in transito quando AWS IoT TwinMaker interagisce con un bucket Amazon S3. Per ulteriori informazioni sui bucket Amazon S3, consulta [Creazione, configurazione e utilizzo dei bucket Amazon S3](#).

## Identity and Access Management per AWS IoT TwinMaker

AWS Identity and Access Management (IAM) è un software Servizio AWS che aiuta un amministratore a controllare in modo sicuro l'accesso alle AWS risorse. Gli amministratori IAM controllano chi può essere autenticato (effettuato l'accesso) e autorizzato (disporre delle autorizzazioni) a utilizzare le risorse. AWS IoT TwinMaker IAM è uno Servizio AWS strumento che puoi utilizzare senza costi aggiuntivi.

### Argomenti

- [Destinatari](#)
- [Autenticazione con identità](#)
- [Gestione dell'accesso con policy](#)
- [Come AWS IoT TwinMaker funziona con IAM](#)
- [Esempi di policy basate sull'identità per AWS IoT TwinMaker](#)

- [Risoluzione dei problemi AWS IoT TwinMaker di identità e accesso](#)
- [Utilizzo di ruoli collegati ai servizi per AWS IoT TwinMaker](#)
- [AWS politiche gestite per AWS IoT TwinMaker](#)

## Destinatari

Il modo in cui usi AWS Identity and Access Management (IAM) varia a seconda del lavoro che AWS IoT TwinMaker svolgi.

Utente del servizio: se utilizzi il AWS IoT TwinMaker servizio per svolgere il tuo lavoro, l'amministratore ti fornisce le credenziali e le autorizzazioni necessarie. Man mano che utilizzi più AWS IoT TwinMaker funzionalità per svolgere il tuo lavoro, potresti aver bisogno di autorizzazioni aggiuntive. La comprensione della gestione dell'accesso ti consente di richiedere le autorizzazioni corrette all'amministratore. Se non riesci ad accedere a una funzionalità di AWS IoT TwinMaker, consulta [Risoluzione dei problemi AWS IoT TwinMaker di identità e accesso](#).

Amministratore del servizio: se sei responsabile delle AWS IoT TwinMaker risorse della tua azienda, probabilmente hai pieno accesso a AWS IoT TwinMaker. È tuo compito determinare a quali AWS IoT TwinMaker funzionalità e risorse devono accedere gli utenti del servizio. Devi inviare le richieste all'amministratore IAM per cambiare le autorizzazioni degli utenti del servizio. Esamina le informazioni contenute in questa pagina per comprendere i concetti di base relativi a IAM. Per saperne di più su come la tua azienda può utilizzare IAM con AWS IoT TwinMaker, consulta [Come AWS IoT TwinMaker funziona con IAM](#).

Amministratore IAM: un amministratore IAM potrebbe essere interessato a ottenere dei dettagli su come scrivere policy per gestire l'accesso a AWS IoT TwinMaker. Per visualizzare esempi di policy AWS IoT TwinMaker basate sull'identità che puoi utilizzare in IAM, consulta [Esempi di policy basate sull'identità per AWS IoT TwinMaker](#)

## Autenticazione con identità

L'autenticazione è il modo in cui accedi AWS utilizzando le tue credenziali di identità. Devi essere autenticato (aver effettuato l' Utente root dell'account AWS accesso AWS) come utente IAM o assumendo un ruolo IAM.

Puoi accedere AWS come identità federata utilizzando le credenziali fornite tramite una fonte di identità. AWS IAM Identity Center Gli utenti (IAM Identity Center), l'autenticazione Single Sign-On della tua azienda e le tue credenziali di Google o Facebook sono esempi di identità federate. Se

accedi come identità federata, l'amministratore ha configurato in precedenza la federazione delle identità utilizzando i ruoli IAM. Quando accedi AWS utilizzando la federazione, assumi indirettamente un ruolo.

A seconda del tipo di utente, puoi accedere al AWS Management Console o al portale di AWS accesso. Per ulteriori informazioni sull'accesso a AWS, vedi [Come accedere al tuo Account AWS nella Guida per l'Accedi ad AWS utente](#).

Se accedi a AWS livello di codice, AWS fornisce un kit di sviluppo software (SDK) e un'interfaccia a riga di comando (CLI) per firmare crittograficamente le tue richieste utilizzando le tue credenziali. Se non utilizzi AWS strumenti, devi firmare tu stesso le richieste. Per ulteriori informazioni sull'utilizzo del metodo consigliato per firmare autonomamente le richieste, consulta [Signing AWS API request](#) nella IAM User Guide.

A prescindere dal metodo di autenticazione utilizzato, potrebbe essere necessario specificare ulteriori informazioni sulla sicurezza. Ad esempio, ti AWS consiglia di utilizzare l'autenticazione a più fattori (MFA) per aumentare la sicurezza del tuo account. Per ulteriori informazioni, consulta [Autenticazione a più fattori](#) nella Guida per l'utente di AWS IAM Identity Center e [Utilizzo dell'autenticazione a più fattori \(MFA\) in AWS](#) nella Guida per l'utente di IAM.

## Account AWS utente root

Quando si crea un account Account AWS, si inizia con un'identità di accesso che ha accesso completo a tutte Servizi AWS le risorse dell'account. Questa identità è denominata utente Account AWS root ed è accessibile effettuando l'accesso con l'indirizzo e-mail e la password utilizzati per creare l'account. Si consiglia vivamente di non utilizzare l'utente root per le attività quotidiane. Conserva le credenziali dell'utente root e utilizzarle per eseguire le operazioni che solo l'utente root può eseguire. Per un elenco completo delle attività che richiedono l'accesso come utente root, consulta la sezione [Attività che richiedono le credenziali dell'utente root](#) nella Guida per l'utente di IAM.

## Identità federata

Come procedura consigliata, richiedi agli utenti umani, compresi gli utenti che richiedono l'accesso come amministratore, di utilizzare la federazione con un provider di identità per accedere Servizi AWS utilizzando credenziali temporanee.

Un'identità federata è un utente dell'elenco utenti aziendale, di un provider di identità Web AWS Directory Service, della directory Identity Center o di qualsiasi utente che accede utilizzando le Servizi

AWS credenziali fornite tramite un'origine di identità. Quando le identità federate accedono Account AWS, assumono ruoli e i ruoli forniscono credenziali temporanee.

Per la gestione centralizzata degli accessi, consigliamo di utilizzare AWS IAM Identity Center. Puoi creare utenti e gruppi in IAM Identity Center oppure puoi connetterti e sincronizzarti con un set di utenti e gruppi nella tua fonte di identità per utilizzarli su tutte le tue applicazioni. Account AWS Per ulteriori informazioni sul Centro identità IAM, consulta [Cos'è Centro identità IAM?](#) nella Guida per l'utente di AWS IAM Identity Center .

## Utenti e gruppi IAM

Un [utente IAM](#) è un'identità interna Account AWS che dispone di autorizzazioni specifiche per una singola persona o applicazione. Ove possibile, consigliamo di fare affidamento a credenziali temporanee invece di creare utenti IAM con credenziali a lungo termine come le password e le chiavi di accesso. Tuttavia, per casi d'uso specifici che richiedono credenziali a lungo termine con utenti IAM, si consiglia di ruotare le chiavi di accesso. Per ulteriori informazioni, consulta la pagina [Rotazione periodica delle chiavi di accesso per casi d'uso che richiedono credenziali a lungo termine](#) nella Guida per l'utente di IAM.

Un [gruppo IAM](#) è un'identità che specifica un insieme di utenti IAM. Non è possibile eseguire l'accesso come gruppo. È possibile utilizzare gruppi per specificare le autorizzazioni per più utenti alla volta. I gruppi semplificano la gestione delle autorizzazioni per set di utenti di grandi dimensioni. Ad esempio, è possibile avere un gruppo denominato Amministratori IAM e concedere a tale gruppo le autorizzazioni per amministrare le risorse IAM.

Gli utenti sono diversi dai ruoli. Un utente è associato in modo univoco a una persona o un'applicazione, mentre un ruolo è destinato a essere assunto da chiunque ne abbia bisogno. Gli utenti dispongono di credenziali a lungo termine permanenti, mentre i ruoli forniscono credenziali temporanee. Per ulteriori informazioni, consulta [Quando creare un utente IAM \(invece di un ruolo\)](#) nella Guida per l'utente di IAM.

## Ruoli IAM

Un [ruolo IAM](#) è un'identità interna all'utente Account AWS che dispone di autorizzazioni specifiche. È simile a un utente IAM, ma non è associato a una persona specifica. Puoi assumere temporaneamente un ruolo IAM in AWS Management Console [cambiando ruolo](#). Puoi assumere un ruolo chiamando un'operazione AWS CLI o AWS API o utilizzando un URL personalizzato. Per ulteriori informazioni sui metodi per l'utilizzo dei ruoli, consulta [Utilizzo di ruoli IAM](#) nella Guida per l'utente di IAM.

I ruoli IAM con credenziali temporanee sono utili nelle seguenti situazioni:

- **Accesso utente federato:** per assegnare le autorizzazioni a una identità federata, è possibile creare un ruolo e definire le autorizzazioni per il ruolo. Quando un'identità federata viene autenticata, l'identità viene associata al ruolo e ottiene le autorizzazioni da esso definite. Per ulteriori informazioni sulla federazione dei ruoli, consulta [Creazione di un ruolo per un provider di identità di terza parte](#) nella Guida per l'utente di IAM. Se utilizzi IAM Identity Center, configura un set di autorizzazioni. IAM Identity Center mette in correlazione il set di autorizzazioni con un ruolo in IAM per controllare a cosa possono accedere le identità dopo l'autenticazione. Per informazioni sui set di autorizzazioni, consulta [Set di autorizzazioni](#) nella Guida per l'utente di AWS IAM Identity Center .
- **Autorizzazioni utente IAM temporanee:** un utente IAM o un ruolo può assumere un ruolo IAM per ottenere temporaneamente autorizzazioni diverse per un'attività specifica.
- **Accesso multi-account:** è possibile utilizzare un ruolo IAM per permettere a un utente (un principale affidabile) con un account diverso di accedere alle risorse nell'account. I ruoli sono lo strumento principale per concedere l'accesso multi-account. Tuttavia, con alcuni Servizi AWS, è possibile allegare una policy direttamente a una risorsa (anziché utilizzare un ruolo come proxy). Per informazioni sulle differenze tra ruoli e policy basate su risorse per l'accesso multi-account, consulta [Differenza tra i ruoli IAM e le policy basate su risorse](#) nella Guida per l'utente di IAM.
- **Accesso a più servizi:** alcuni Servizi AWS utilizzano le funzionalità di altri Servizi AWS. Ad esempio, quando effettui una chiamata in un servizio, è comune che tale servizio esegua applicazioni in Amazon EC2 o archivi oggetti in Amazon S3. Un servizio può eseguire questa operazione utilizzando le autorizzazioni dell'entità chiamante, utilizzando un ruolo di servizio o utilizzando un ruolo collegato al servizio.
- **Sessioni di accesso diretto (FAS):** quando utilizzi un utente o un ruolo IAM per eseguire azioni AWS, sei considerato un preside. Quando si utilizzano alcuni servizi, è possibile eseguire un'operazione che attiva un'altra azione in un servizio diverso. FAS utilizza le autorizzazioni del principale che chiama un Servizio AWS, combinate con la richiesta Servizio AWS per effettuare richieste ai servizi downstream. Le richieste FAS vengono effettuate solo quando un servizio riceve una richiesta che richiede interazioni con altri Servizi AWS o risorse per essere completata. In questo caso è necessario disporre delle autorizzazioni per eseguire entrambe le azioni. Per i dettagli delle policy relative alle richieste FAS, consulta la pagina [Forward access sessions](#).
- **Ruolo di servizio:** un ruolo di servizio è un [ruolo IAM](#) che un servizio assume per eseguire azioni per tuo conto. Un amministratore IAM può creare, modificare ed eliminare un ruolo di servizio

dall'interno di IAM. Per ulteriori informazioni, consulta la sezione [Creazione di un ruolo per delegare le autorizzazioni a un Servizio AWS](#) nella Guida per l'utente di IAM.

- Ruolo collegato al servizio: un ruolo collegato al servizio è un tipo di ruolo di servizio collegato a un Servizio AWS. Il servizio può assumere il ruolo per eseguire un'azione per tuo conto. I ruoli collegati al servizio vengono visualizzati nel tuo account Account AWS e sono di proprietà del servizio. Un amministratore IAM può visualizzare le autorizzazioni per i ruoli collegati ai servizi, ma non modificarle.
- Applicazioni in esecuzione su Amazon EC2: puoi utilizzare un ruolo IAM per gestire le credenziali temporanee per le applicazioni in esecuzione su un'istanza EC2 e che AWS CLI effettuano richieste API. AWS Cloud è preferibile all'archiviazione delle chiavi di accesso nell'istanza EC2. Per assegnare un ruolo AWS a un'istanza EC2 e renderlo disponibile per tutte le sue applicazioni, crei un profilo di istanza collegato all'istanza. Un profilo dell'istanza contiene il ruolo e consente ai programmi in esecuzione sull'istanza EC2 di ottenere le credenziali temporanee. Per ulteriori informazioni, consulta [Utilizzo di un ruolo IAM per concedere autorizzazioni ad applicazioni in esecuzione su istanze di Amazon EC2](#) nella Guida per l'utente di IAM.

Per informazioni sull'utilizzo dei ruoli IAM, consulta [Quando creare un ruolo IAM \(invece di un utente\)](#) nella Guida per l'utente di IAM.

## Gestione dell'accesso con policy

Puoi controllare l'accesso AWS creando policy e collegandole a AWS identità o risorse. Una policy è un oggetto AWS che, se associato a un'identità o a una risorsa, ne definisce le autorizzazioni. AWS valuta queste politiche quando un principale (utente, utente root o sessione di ruolo) effettua una richiesta. Le autorizzazioni nelle policy determinano l'approvazione o il rifiuto della richiesta. La maggior parte delle politiche viene archiviata AWS come documenti JSON. Per ulteriori informazioni sulla struttura e sui contenuti dei documenti delle policy JSON, consulta [Panoramica delle policy JSON](#) nella Guida per l'utente di IAM.

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. In altre parole, quale principale può eseguire azioni su quali risorse e in quali condizioni.

Per impostazione predefinita, utenti e ruoli non dispongono di autorizzazioni. Per concedere agli utenti l'autorizzazione a eseguire azioni sulle risorse di cui hanno bisogno, un amministratore IAM può creare policy IAM. Successivamente l'amministratore può aggiungere le policy IAM ai ruoli e gli utenti possono assumere i ruoli.



Le policy IAM definiscono le autorizzazioni relative a un'azione, a prescindere dal metodo utilizzato per eseguirla. Ad esempio, supponiamo di disporre di una policy che consente l'azione `iam:GetRole`. Un utente con tale policy può ottenere informazioni sul ruolo dall' AWS Management Console AWS CLI, dall' AWS CLI o dall' AWS API.

## Policy basate su identità

Le policy basate su identità sono documenti di policy di autorizzazione JSON che è possibile allegare a un'identità (utente, gruppo di utenti o ruoli IAM). Tali policy definiscono le azioni che utenti e ruoli possono eseguire, su quali risorse e in quali condizioni. Per informazioni su come creare una policy basata su identità, consulta [Creazione di policy IAM](#) nella Guida per l'utente di IAM.

Le policy basate su identità possono essere ulteriormente classificate come policy inline o policy gestite. Le policy inline sono integrate direttamente in un singolo utente, gruppo o ruolo. Le politiche gestite sono politiche autonome che puoi allegare a più utenti, gruppi e ruoli nel tuo Account AWS. Le politiche gestite includono politiche AWS gestite e politiche gestite dai clienti. Per informazioni su come scegliere tra una policy gestita o una policy inline, consulta [Scelta fra policy gestite e policy inline](#) nella Guida per l'utente di IAM.

## Policy basate su risorse

Le policy basate su risorse sono documenti di policy JSON che è possibile collegare a una risorsa. Gli esempi più comuni di policy basate su risorse sono le policy di attendibilità dei ruoli IAM e le policy dei bucket Amazon S3. Nei servizi che supportano policy basate sulle risorse, gli amministratori dei servizi possono utilizzarle per controllare l'accesso a una risorsa specifica. Quando è collegata a una risorsa, una policy definisce le azioni che un principale può eseguire su tale risorsa e a quali condizioni. È necessario [specificare un principale](#) in una policy basata sulle risorse. I principali possono includere account, utenti, ruoli, utenti federati o Servizi AWS.

Le policy basate sulle risorse sono policy inline che si trovano in tale servizio. Non puoi utilizzare le policy AWS gestite di IAM in una policy basata sulle risorse.

## Liste di controllo degli accessi (ACL)

Le liste di controllo degli accessi (ACL) controllano quali principali (membri, utenti o ruoli dell'account) hanno le autorizzazioni per accedere a una risorsa. Le ACL sono simili alle policy basate su risorse, sebbene non utilizzino il formato del documento di policy JSON.

Amazon S3 e Amazon VPC sono esempi di servizi che supportano gli ACL. AWS WAF Per maggiori informazioni sulle ACL, consulta [Panoramica delle liste di controllo degli accessi \(ACL\)](#) nella Guida per gli sviluppatori di Amazon Simple Storage Service.

## Altri tipi di policy

AWS supporta tipi di policy aggiuntivi e meno comuni. Questi tipi di policy possono impostare il numero massimo di autorizzazioni concesse dai tipi di policy più comuni.

- **Limiti delle autorizzazioni:** un limite delle autorizzazioni è una funzione avanzata nella quale si imposta il numero massimo di autorizzazioni che una policy basata su identità può concedere a un'entità IAM (utente o ruolo IAM). È possibile impostare un limite delle autorizzazioni per un'entità. Le autorizzazioni risultanti sono l'intersezione delle policy basate su identità dell'entità e i relativi limiti delle autorizzazioni. Le policy basate su risorse che specificano l'utente o il ruolo nel campo `Principal` sono condizionate dal limite delle autorizzazioni. Un rifiuto esplicito in una qualsiasi di queste policy sostituisce l'autorizzazione. Per ulteriori informazioni sui limiti delle autorizzazioni, consulta [Limiti delle autorizzazioni per le entità IAM](#) nella Guida per l'utente di IAM.
- **Politiche di controllo dei servizi (SCP):** le SCP sono politiche JSON che specificano le autorizzazioni massime per un'organizzazione o un'unità organizzativa (OU) in AWS Organizations. AWS Organizations è un servizio per il raggruppamento e la gestione centralizzata di più Account AWS di proprietà dell'azienda. Se abiliti tutte le funzionalità in un'organizzazione, puoi applicare le policy di controllo dei servizi (SCP) a uno o tutti i tuoi account. L'SCP limita le autorizzazioni per le entità negli account dei membri, inclusa ciascuna. Utente root dell'account AWS Per ulteriori informazioni su organizzazioni e policy SCP, consulta la pagina sulle [Policy di controllo dei servizi](#) nella Guida per l'utente di AWS Organizations .
- **Policy di sessione:** le policy di sessione sono policy avanzate che vengono trasmesse come parametro quando si crea in modo programmatico una sessione temporanea per un ruolo o un utente federato. Le autorizzazioni della sessione risultante sono l'intersezione delle policy basate su identità del ruolo o dell'utente e le policy di sessione. Le autorizzazioni possono anche provenire da una policy basata su risorse. Un rifiuto esplicito in una qualsiasi di queste policy sostituisce l'autorizzazione. Per ulteriori informazioni, consulta [Policy di sessione](#) nella Guida per l'utente di IAM.

## Più tipi di policy

Quando più tipi di policy si applicano a una richiesta, le autorizzazioni risultanti sono più complicate da comprendere. Per scoprire come si AWS determina se consentire una richiesta quando sono coinvolti più tipi di policy, consulta [Logica di valutazione delle policy](#) nella IAM User Guide.

## Come AWS IoT TwinMaker funziona con IAM

Prima di utilizzare IAM per gestire l'accesso a AWS IoT TwinMaker, scopri con quali funzionalità IAM è disponibile l'uso AWS IoT TwinMaker.

Funzionalità IAM che puoi utilizzare con AWS IoT TwinMaker

Funzionalità IAM	AWS IoT TwinMaker supporto
<a href="#">Policy basate su identità</a>	Sì
<a href="#">Policy basate su risorse</a>	No
<a href="#">Azioni di policy</a>	Sì
<a href="#">Risorse relative alle policy</a>	Sì
<a href="#">Chiavi di condizione delle policy</a>	Sì
<a href="#">Liste di controllo degli accessi (ACL)</a>	No
<a href="#">ABAC (tag nelle policy)</a>	Parziale
<a href="#">Credenziali temporanee</a>	Sì
<a href="#">Autorizzazioni del principale</a>	Sì
<a href="#">Ruoli di servizio</a>	Sì
<a href="#">Ruoli collegati al servizio</a>	No

Per avere una visione generale di come AWS IoT TwinMaker e altri AWS servizi funzionano con la maggior parte delle funzionalità IAM, consulta [AWS i servizi che funzionano con IAM](#) nella Guida per l'AWS IAM Identity Center utente.

## Politiche basate sull'identità per AWS IoT TwinMaker

Supporta le policy basate su identità Si

Le policy basate su identità sono documenti di policy di autorizzazione JSON che è possibile allegare a un'identità (utente, gruppo di utenti o ruolo IAM). Tali policy definiscono le azioni che utenti e ruoli possono eseguire, su quali risorse e in quali condizioni. Per informazioni su come creare una policy basata su identità, consulta [Creazione di policy IAM](#) nella Guida per l'utente di IAM.

Con le policy basate su identità di IAM, è possibile specificare quali operazioni e risorse sono consentite o respinte, nonché le condizioni in base alle quali le operazioni sono consentite o respinte. Non è possibile specificare l'entità principale in una policy basata sull'identità perché si applica all'utente o al ruolo a cui è associato. Per informazioni su tutti gli elementi utilizzabili in una policy JSON, consulta [Guida di riferimento agli elementi delle policy JSON IAM](#) nella Guida per l'utente di IAM.

Esempi di politiche basate sull'identità per AWS IoT TwinMaker

Per visualizzare esempi di politiche basate sull' AWS IoT TwinMaker identità, vedere. [Esempi di policy basate sull'identità per AWS IoT TwinMaker](#)

## Politiche basate sulle risorse all'interno AWS IoT TwinMaker

Supporta le policy basate su risorse No

Le policy basate su risorse sono documenti di policy JSON che è possibile collegare a una risorsa. Gli esempi più comuni di policy basate su risorse sono le policy di attendibilità dei ruoli IAM e le policy dei bucket Amazon S3. Nei servizi che supportano policy basate sulle risorse, gli amministratori dei servizi possono utilizzarle per controllare l'accesso a una risorsa specifica. Quando è collegata a una risorsa, una policy definisce le azioni che un principale può eseguire su tale risorsa e a quali condizioni. È necessario [specificare un principale](#) in una policy basata sulle risorse. I principali possono includere account, utenti, ruoli, utenti federati o. Servizi AWS

Per consentire l'accesso multi-account, puoi specificare un intero account o entità IAM in un altro account come principale in una policy basata sulle risorse. L'aggiunta di un principale multi-account

a una policy basata sulle risorse rappresenta solo una parte della relazione di trust. Quando il principale e la risorsa sono diversi Account AWS, un amministratore IAM dell'account affidabile deve inoltre concedere all'entità principale (utente o ruolo) l'autorizzazione ad accedere alla risorsa. L'autorizzazione viene concessa collegando all'entità una policy basata sull'identità. Tuttavia, se una policy basata su risorse concede l'accesso a un principale nello stesso account, non sono richieste ulteriori policy basate su identità. Per ulteriori informazioni, consulta [Differenza tra i ruoli IAM e le policy basate su risorse](#) nella Guida per l'utente di IAM.

## Azioni politiche per AWS IoT TwinMaker

Supporta le azioni di policy	Sì
------------------------------	----

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. Cioè, quale principale può eseguire azioni su quali risorse, e in quali condizioni.

L'elemento `Action` di una policy JSON descrive le azioni che è possibile utilizzare per consentire o negare l'accesso a un criterio. Le azioni politiche in genere hanno lo stesso nome dell'operazione AWS API associata. Ci sono alcune eccezioni, ad esempio le azioni di sola autorizzazione che non hanno un'operazione API corrispondente. Esistono anche alcune operazioni che richiedono più operazioni in una policy. Queste operazioni aggiuntive sono denominate operazioni dipendenti.

Includi le operazioni in una policy per concedere le autorizzazioni a eseguire l'operazione associata.

Per visualizzare un elenco di AWS IoT TwinMaker azioni, vedere [Azioni definite da AWS IoT TwinMaker](#) nel Service Authorization Reference.

Le azioni politiche in AWS IoT TwinMaker uso utilizzano il seguente prefisso prima dell'azione:

```
iottwinmaker
```

Per specificare più operazioni in una sola istruzione, occorre separarle con la virgola.

```
"Action": [  
  "iottwinmaker:action1",  
  "iottwinmaker:action2"  
]
```

Per visualizzare esempi di politiche AWS IoT TwinMaker basate sull'identità, vedere. [Esempi di policy basate sull'identità per AWS IoT TwinMaker](#)

## Risorse politiche per AWS IoT TwinMaker

Supporta le risorse di policy	Si
-------------------------------	----

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. Cioè, quale principale può eseguire operazioni su quali risorse, e in quali condizioni.

L'elemento JSON `Resource` della policy specifica l'oggetto o gli oggetti ai quali si applica l'azione. Le istruzioni devono includere un elemento `Resource` o un elemento `NotResource`. Come best practice, specifica una risorsa utilizzando il suo [nome della risorsa Amazon \(ARN\)](#). Puoi eseguire questa operazione per azioni che supportano un tipo di risorsa specifico, note come autorizzazioni a livello di risorsa.

Per le azioni che non supportano le autorizzazioni a livello di risorsa, ad esempio le operazioni di elenco, utilizza un carattere jolly (\*) per indicare che l'istruzione si applica a tutte le risorse.

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

Per visualizzare un elenco dei tipi di AWS IoT TwinMaker risorse e dei relativi ARN, consulta [Resources defined by AWS IoT TwinMaker](#) nel Service Authorization Reference. Per informazioni sulle operazioni con cui è possibile specificare l'ARN di ogni risorsa, consulta la sezione [Operazioni definite da AWS IoT TwinMaker](#).

Per visualizzare esempi di politiche AWS IoT TwinMaker basate sull'identità, vedere. [Esempi di policy basate sull'identità per AWS IoT TwinMaker](#)

## Chiavi relative alle condizioni delle politiche per AWS IoT TwinMaker

Supporta le chiavi di condizione delle policy specifiche del servizio	Si
---	----

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. Cioè, quale principale può eseguire azioni su quali risorse, e in quali condizioni.

L'elemento `Condition` (o blocco `Condition`) consente di specificare le condizioni in cui un'istruzione è in vigore. L'elemento `Condition` è facoltativo. Puoi compilare espressioni condizionali che utilizzano [operatori di condizione](#), ad esempio uguale a o minore di, per soddisfare la condizione nella policy con i valori nella richiesta.

Se specifichi più elementi `Condition` in un'istruzione o più chiavi in un singolo elemento `Condition`, questi vengono valutati da AWS utilizzando un'operazione AND logica. Se si specificano più valori per una singola chiave di condizione, AWS valuta la condizione utilizzando un'operazione logica. OR Tutte le condizioni devono essere soddisfatte prima che le autorizzazioni dell'istruzione vengano concesse.

Puoi anche utilizzare variabili segnaposto quando specifichi le condizioni. Ad esempio, puoi autorizzare un utente IAM ad accedere a una risorsa solo se è stata taggata con il relativo nome utente IAM. Per ulteriori informazioni, consulta [Elementi delle policy IAM: variabili e tag](#) nella Guida per l'utente di IAM.

AWS supporta chiavi di condizione globali e chiavi di condizione specifiche del servizio. Per visualizzare tutte le chiavi di condizione AWS globali, consulta le chiavi di [contesto delle condizioni AWS globali nella Guida](#) per l'utente IAM.

Per visualizzare un elenco di chiavi di AWS IoT TwinMaker condizione, consulta [Condition keys for AWS IoT TwinMaker](#) nel Service Authorization Reference. Per sapere con quali azioni e risorse puoi utilizzare una chiave di condizione, vedi [Azioni definite da AWS IoT TwinMaker](#).

Per visualizzare esempi di politiche AWS IoT TwinMaker basate sull'identità, vedere. [Esempi di policy basate sull'identità per AWS IoT TwinMaker](#)

## Liste di controllo degli accessi (ACL) in AWS IoT TwinMaker

Supporta le ACL

No

Le liste di controllo degli accessi (ACL) controllano quali principali (membri, utenti o ruoli dell'account) hanno le autorizzazioni ad accedere a una risorsa. Le ACL sono simili alle policy basate su risorse, sebbene non utilizzino il formato del documento di policy JSON.

## Controllo degli accessi basato sugli attributi (ABAC) con AWS IoT TwinMaker

Supporta ABAC (tag nelle policy)

Parziale

Il controllo dell'accesso basato su attributi (ABAC) è una strategia di autorizzazione che definisce le autorizzazioni in base agli attributi. In AWS, questi attributi sono chiamati tag. Puoi allegare tag a entità IAM (utenti o ruoli) e a molte AWS risorse. L'assegnazione di tag alle entità e alle risorse è il primo passaggio di ABAC. In seguito, vengono progettate policy ABAC per consentire operazioni quando il tag dell'entità principale corrisponde al tag sulla risorsa a cui si sta provando ad accedere.

La strategia ABAC è utile in ambienti soggetti a una rapida crescita e aiuta in situazioni in cui la gestione delle policy diventa impegnativa.

Per controllare l'accesso basato su tag, fornisci informazioni sui tag nell'[elemento condizione](#) di una policy utilizzando le chiavi di condizione `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` o `aws:TagKeys`.

Se un servizio supporta tutte e tre le chiavi di condizione per ogni tipo di risorsa, il valore per il servizio è Yes (Sì). Se un servizio supporta tutte e tre le chiavi di condizione solo per alcuni tipi di risorsa, allora il valore sarà Parziale.

Per ulteriori informazioni su ABAC, consulta [Che cos'è ABAC?](#) nella Guida per l'utente di IAM. Per visualizzare un tutorial con i passaggi per l'impostazione di ABAC, consulta [Utilizzo del controllo degli accessi basato su attributi \(ABAC\)](#) nella Guida per l'utente di IAM.

## Utilizzo di credenziali temporanee con AWS IoT TwinMaker

Supporta le credenziali temporanee	Sì
------------------------------------	----

Alcune Servizi AWS non funzionano quando si accede utilizzando credenziali temporanee. Per ulteriori informazioni, incluse quelle che Servizi AWS funzionano con credenziali temporanee, consulta la sezione relativa alla [Servizi AWS compatibilità con IAM nella IAM User Guide](#).

Stai utilizzando credenziali temporanee se accedi AWS Management Console utilizzando qualsiasi metodo tranne nome utente e password. Ad esempio, quando accedete AWS utilizzando il link Single Sign-On (SSO) della vostra azienda, tale processo crea automaticamente credenziali temporanee. Le credenziali temporanee vengono create in automatico anche quando accedi alla console come utente e poi cambi ruolo. Per ulteriori informazioni sullo scambio dei ruoli, consulta [Cambio di un ruolo \(console\)](#) nella Guida per l'utente di IAM.

È possibile creare manualmente credenziali temporanee utilizzando l'API o AWS CLI. AWS consiglia di generare quindi possibile utilizzare tali credenziali temporanee per accedere. AWS consiglia di generare



dinamicamente credenziali temporanee anziché utilizzare chiavi di accesso a lungo termine. Per ulteriori informazioni, consulta [Credenziali di sicurezza provvisorie in IAM](#).

## Autorizzazioni principali multiservizio per AWS IoT TwinMaker

Supporta sessioni di accesso diretto (FAS)	Sì
--	----

Quando utilizzi un utente o un ruolo IAM per eseguire azioni AWS, sei considerato un principale. Quando si utilizzano alcuni servizi, è possibile eseguire un'operazione che attiva un'altra azione in un servizio diverso. FAS utilizza le autorizzazioni del principale che chiama un Servizio AWS, in combinazione con la richiesta Servizio AWS per effettuare richieste ai servizi downstream. Le richieste FAS vengono effettuate solo quando un servizio riceve una richiesta che richiede interazioni con altri Servizi AWS o risorse per essere completata. In questo caso è necessario disporre delle autorizzazioni per eseguire entrambe le azioni. Per i dettagli delle policy relative alle richieste FAS, consulta la pagina [Inoltre sessioni di accesso](#).

## Ruoli di servizio per AWS IoT TwinMaker

Supporta i ruoli di servizio	Sì
------------------------------	----

Un ruolo di servizio è un [ruolo IAM](#) che un servizio assume per eseguire operazioni per tuo conto. Un amministratore IAM può creare, modificare ed eliminare un ruolo di servizio dall'interno di IAM. Per ulteriori informazioni, consulta la sezione [Creazione di un ruolo per delegare le autorizzazioni a un Servizio AWS](#) nella Guida per l'utente di IAM.

### Warning

La modifica delle autorizzazioni per un ruolo di servizio potrebbe compromettere la funzionalità. AWS IoT TwinMaker Modifica i ruoli di servizio solo quando viene AWS IoT TwinMaker fornita una guida in tal senso.

## Ruoli collegati ai servizi per AWS IoT TwinMaker

Supporta i ruoli collegati ai servizi	No
---------------------------------------	----

Un ruolo collegato al servizio è un tipo di ruolo di servizio collegato a un servizio AWS. Il servizio può assumere il ruolo per eseguire un'azione per tuo conto. I ruoli collegati al servizio vengono visualizzati nel tuo account Account AWS e sono di proprietà del servizio. Un amministratore IAM può visualizzare le autorizzazioni per i ruoli collegati ai servizi, ma non modificarle.

Per ulteriori informazioni su come creare e gestire i ruoli collegati ai servizi, consulta [Servizi AWS supportati da IAM](#). Trova un servizio nella tabella che include un Yes nella colonna Service-linked role (Ruolo collegato ai servizi). Scegli il collegamento Sì per visualizzare la documentazione relativa al ruolo collegato ai servizi per tale servizio.

## Esempi di policy basate sull'identità per AWS IoT TwinMaker

Per impostazione predefinita, gli utenti e i ruoli non dispongono dell'autorizzazione per creare o modificare risorse AWS IoT TwinMaker. Inoltre, non possono eseguire attività utilizzando AWS Management Console, AWS Command Line Interface (AWS CLI) o l'API. AWS Per concedere agli utenti l'autorizzazione a eseguire azioni sulle risorse di cui hanno bisogno, un amministratore IAM può creare policy IAM. L'amministratore può quindi aggiungere le policy IAM ai ruoli e gli utenti possono assumere i ruoli.

Per informazioni su come creare una policy basata su identità IAM utilizzando questi documenti di policy JSON di esempio, consulta [Creazione di policy IAM](#) nella Guida per l'utente di IAM.

Per informazioni dettagliate sulle azioni e sui tipi di risorse definiti da AWS IoT TwinMaker, incluso il formato degli ARN per ciascun tipo di risorsa, consulta [Azioni, risorse e chiavi di condizione AWS IoT TwinMaker](#) nel Service Authorization Reference.

### Argomenti

- [Best practice per le policy](#)
- [Utilizzo della console di AWS IoT TwinMaker](#)
- [Consentire agli utenti di visualizzare le loro autorizzazioni](#)

## Best practice per le policy

Le politiche basate sull'identità determinano se qualcuno può creare, accedere o eliminare AWS IoT TwinMaker risorse nel tuo account. Queste azioni possono comportare costi aggiuntivi per l'Account AWS. Quando crei o modifichi policy basate su identità, segui queste linee guida e raccomandazioni:

- Inizia con le policy AWS gestite e passa alle autorizzazioni con privilegi minimi: per iniziare a concedere autorizzazioni a utenti e carichi di lavoro, utilizza le politiche gestite che concedono le autorizzazioni per molti casi d'uso comuni. AWS Sono disponibili nel tuo Account AWS Ti consigliamo di ridurre ulteriormente le autorizzazioni definendo politiche gestite dai AWS clienti specifiche per i tuoi casi d'uso. Per ulteriori informazioni, consulta [Policy gestite da AWS](#) o [Policy gestite da AWS per le funzioni dei processi](#) nella Guida per l'utente IAM.
- Applica le autorizzazioni con privilegi minimi: quando imposti le autorizzazioni con le policy IAM, concedi solo le autorizzazioni richieste per eseguire un'attività. Puoi farlo definendo le azioni che possono essere intraprese su risorse specifiche in condizioni specifiche, note anche come autorizzazioni con privilegi minimi. Per ulteriori informazioni sull'utilizzo di IAM per applicare le autorizzazioni, consulta [Policy e autorizzazioni in IAM](#) nella Guida per l'utente di IAM.
- Condizioni d'uso nelle policy IAM per limitare ulteriormente l'accesso: per limitare l'accesso ad azioni e risorse puoi aggiungere una condizione alle tue policy. Ad esempio, è possibile scrivere una condizione di policy per specificare che tutte le richieste devono essere inviate utilizzando SSL. Puoi anche utilizzare le condizioni per concedere l'accesso alle azioni del servizio se vengono utilizzate tramite uno specifico Servizio AWS, ad esempio AWS CloudFormation. Per ulteriori informazioni, consulta la sezione [Elementi delle policy JSON di IAM: condizione](#) nella Guida per l'utente di IAM.
- Utilizzo di IAM Access Analyzer per convalidare le policy IAM e garantire autorizzazioni sicure e funzionali: IAM Access Analyzer convalida le policy nuove ed esistenti in modo che aderiscano alla sintassi della policy IAM (JSON) e alle best practice di IAM. IAM Access Analyzer offre oltre 100 controlli delle policy e consigli utili per creare policy sicure e funzionali. Per ulteriori informazioni, consulta [Convalida delle policy per IAM Access Analyzer](#) nella Guida per l'utente di IAM.
- Richiedi l'autenticazione a più fattori (MFA): se hai uno scenario che richiede utenti IAM o un utente root nel Account AWS tuo, attiva l'MFA per una maggiore sicurezza. Per richiedere la MFA quando vengono chiamate le operazioni API, aggiungi le condizioni MFA alle policy. Per ulteriori informazioni, consulta [Configurazione dell'accesso alle API protetto con MFA](#) nella Guida per l'utente di IAM.

Per maggiori informazioni sulle best practice in IAM, consulta [Best practice di sicurezza in IAM](#) nella Guida per l'utente di IAM.

## Utilizzo della console di AWS IoT TwinMaker

Per accedere alla AWS IoT TwinMaker console, devi disporre di un set minimo di autorizzazioni. Queste autorizzazioni devono consentirti di elencare e visualizzare i dettagli sulle AWS IoT

TwinMaker risorse del tuo. Account AWS Se crei una policy basata sull'identità più restrittiva rispetto alle autorizzazioni minime richieste, la console non funzionerà nel modo previsto per le entità (utenti o ruoli) associate a tale policy.

Non è necessario consentire autorizzazioni minime per la console per gli utenti che effettuano chiamate solo verso AWS CLI o l' AWS API. Al contrario, concedi l'accesso solo alle operazioni che corrispondono all'operazione API che stanno cercando di eseguire.

Per garantire che utenti e ruoli possano ancora utilizzare la AWS IoT TwinMaker console, allega anche la policy AWS IoT TwinMaker ConsoleAccess o la policy ReadOnly AWS gestita alle entità. Per ulteriori informazioni, consulta [Aggiungere autorizzazioni a un utente](#) nella Guida per l'AWS IAM Identity Center utente.

## Consentire agli utenti di visualizzare le loro autorizzazioni

Questo esempio mostra in che modo è possibile creare una policy che consente agli utenti IAM di visualizzare le policy inline e gestite che sono collegate alla relativa identità utente. Questa politica include le autorizzazioni per completare questa azione sulla console o utilizzando l'API o a livello di codice. AWS CLI 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": "*"
}
]
```

## Risoluzione dei problemi AWS IoT TwinMaker di identità e accesso

Utilizza le seguenti informazioni per aiutarti a diagnosticare e risolvere i problemi più comuni che potresti riscontrare quando lavori con un AWS IoT TwinMaker IAM.

### Argomenti

- [Non sono autorizzato a eseguire alcuna azione in AWS IoT TwinMaker](#)
- [Non sono autorizzato a eseguire iam: PassRole](#)
- [Voglio consentire a persone esterne a me di accedere Account AWS alle mie AWS IoT TwinMaker risorse](#)

### Non sono autorizzato a eseguire alcuna azione in AWS IoT TwinMaker

Se ricevi un errore che indica che non sei autorizzato a eseguire un'operazione, le tue policy devono essere aggiornate per poter eseguire l'operazione.

L'errore di esempio seguente si verifica quando l'utente IAM mateojackson prova a utilizzare la console per visualizzare i dettagli relativi a una risorsa *my-example-widget* fittizia ma non dispone di autorizzazioni `iottwinmaker:GetWidget` fittizie.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
iottwinmaker:GetWidget on resource: my-example-widget
```

In questo caso, la policy per l'utente mateojackson deve essere aggiornata per consentire l'accesso alla risorsa *my-example-widget* utilizzando l'azione `iottwinmaker:GetWidget`.

Se hai bisogno di aiuto, contatta il tuo AWS amministratore. L'amministratore è la persona che ti ha fornito le credenziali di accesso.

## Non sono autorizzato a eseguire iam: PassRole

Se ricevi un errore che indica che non sei autorizzato a eseguire l'operazione `iam:PassRole`, le tue policy devono essere aggiornate per poter passare un ruolo a AWS IoT TwinMaker.

Alcuni Servizi AWS consentono di passare un ruolo esistente a quel servizio invece di creare un nuovo ruolo di servizio o un ruolo collegato al servizio. Per eseguire questa operazione, è necessario disporre delle autorizzazioni per trasmettere il ruolo al servizio.

L'errore di esempio seguente si verifica quando un utente IAM denominato `marymajor` cerca di utilizzare la console per eseguire un'operazione in AWS IoT TwinMaker. Tuttavia, l'operazione richiede che il servizio disponga delle autorizzazioni concesse da un ruolo di servizio. Mary non dispone delle autorizzazioni per passare il ruolo al servizio.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
iam:PassRole
```

In questo caso, le policy di Mary devono essere aggiornate per poter eseguire l'operazione `iam:PassRole`.

Se hai bisogno di aiuto, contatta il tuo AWS amministratore. L'amministratore è la persona che ti ha fornito le credenziali di accesso.

## Voglio consentire a persone esterne a me di accedere Account AWS alle mie AWS IoT TwinMaker risorse

È possibile creare un ruolo con il quale utenti in altri account o persone esterne all'organizzazione possono accedere alle tue risorse. È possibile specificare chi è attendibile per l'assunzione del ruolo. Per servizi che supportano policy basate su risorse o liste di controllo accessi (ACL), utilizza tali policy per concedere alle persone l'accesso alle tue risorse.

Per ulteriori informazioni, consulta gli argomenti seguenti:

- Per sapere se AWS IoT TwinMaker supporta queste funzionalità, consulta [Come AWS IoT TwinMaker funziona con IAM](#).

- Per scoprire come fornire l'accesso alle tue risorse attraverso Account AWS le risorse di tua proprietà, consulta [Fornire l'accesso a un utente IAM in un altro Account AWS di tua proprietà](#) nella IAM User Guide.
- Per scoprire come fornire l'accesso alle tue risorse a terze parti Account AWS, consulta [Fornire l'accesso a soggetti Account AWS di proprietà di terze parti](#) nella Guida per l'utente IAM.
- Per informazioni su come fornire l'accesso tramite la federazione delle identità, consulta [Fornire l'accesso a utenti autenticati esternamente \(Federazione delle identità\)](#) nella Guida per l'utente di IAM.
- Per informazioni sulle differenze tra l'utilizzo di ruoli e policy basate su risorse per l'accesso multi-account, consultare [Differenza tra i ruoli IAM e le policy basate su risorse](#) nella Guida per l'utente di IAM.

## Utilizzo di ruoli collegati ai servizi per AWS IoT TwinMaker

AWS IoT TwinMaker utilizza ruoli collegati ai [servizi AWS Identity and Access Management](#) (IAM). Un ruolo collegato ai servizi è un tipo unico di ruolo IAM a cui è collegato direttamente. AWS IoT TwinMaker I ruoli collegati ai servizi sono predefiniti AWS IoT TwinMaker e includono tutte le autorizzazioni richieste dal servizio per chiamare altri servizi per tuo conto. AWS

Un ruolo collegato al servizio semplifica la configurazione AWS IoT TwinMaker perché non è necessario aggiungere manualmente le autorizzazioni necessarie. AWS IoT TwinMaker definisce le autorizzazioni dei ruoli collegati ai servizi e, se non diversamente definito, solo può assumerne i ruoli. AWS IoT TwinMaker Le autorizzazioni definite includono la policy di attendibilità e la policy delle autorizzazioni che non può essere allegata a nessun'altra entità IAM.

È possibile eliminare un ruolo collegato ai servizi solo dopo aver eliminato le risorse correlate. In questo modo proteggi AWS IoT TwinMaker le tue risorse perché non puoi rimuovere inavvertitamente l'autorizzazione ad accedere alle risorse.

Per informazioni sugli altri servizi che supportano i ruoli collegati al servizio, consulta [Servizi AWS che funzionano con IAM](#) e cerca i servizi che riportano Sì nella colonna Ruoli collegati al servizio. Scegli Sì in corrispondenza di un link per visualizzare la documentazione relativa al ruolo collegato al servizio per tale servizio.

## Autorizzazioni di ruolo collegate al servizio per AWS IoT TwinMaker

AWS IoT TwinMaker utilizza il ruolo collegato al servizio denominato `AWSServiceRoleForIoT TwinMaker`: consente di AWS IoT TwinMaker chiamare altri AWS servizi e di sincronizzare le relative risorse per conto dell'utente.

Il ruolo `AWSServiceRoleForIoT TwinMaker` collegato al servizio prevede che i seguenti servizi assumano il ruolo:

- `iottwinmaker.amazonaws.com`

La politica di autorizzazione dei ruoli denominata `AWSIoT TwinMakerServiceRolePolicy` consente di AWS IoT TwinMaker completare le seguenti azioni sulle risorse specificate:

- Operazione: `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` su all your `iotsitewise asset` and `asset-model` resources

Per consentire a utenti, gruppi o ruoli di creare, modificare o eliminare un ruolo orientato ai servizi, devi configurare le autorizzazioni. Per ulteriori informazioni, consulta [Autorizzazioni del ruolo collegato ai servizi](#) nella Guida per l'utente di IAM.

## Creazione di un ruolo collegato al servizio per AWS IoT TwinMaker

Non hai bisogno di creare manualmente un ruolo collegato ai servizi. Quando sincronizzi le tue AWS IoT SiteWise risorse e i tuoi modelli di asset (asset sync) nell'API AWS Management Console, l'AWS API AWS CLI, AWS IoT TwinMaker crea automaticamente il ruolo collegato al servizio.

Se elimini questo ruolo collegato ai servizi, puoi ricrearlo seguendo lo stesso processo utilizzato per ricreare il ruolo nell'account. Quando sincronizzi gli AWS IoT SiteWise asset e i modelli di asset (asset sync), AWS IoT TwinMaker crea nuovamente il ruolo collegato al servizio.

Puoi anche utilizzare la console IAM per creare un ruolo collegato ai servizi con lo use case «IoT TwinMaker - Managed Role». Nella AWS CLI o nell'AWS API, crea un ruolo collegato al servizio con



il nome del servizio. `iotttwinmaker.amazonaws.com` Per ulteriori informazioni, consulta [Creazione di un ruolo collegato ai servizi](#) nella Guida per l'utente di IAM. Se elimini il ruolo collegato ai servizi, puoi utilizzare lo stesso processo per crearlo nuovamente.

## Modifica di un ruolo collegato al servizio per AWS IoT TwinMaker

AWS IoT TwinMaker non consente di modificare il ruolo collegato al `AWSServiceRoleForIoTtwinmaker` servizio. Dopo aver creato un ruolo collegato al servizio, non potrai modificarne il nome perché varie entità potrebbero farvi riferimento. È possibile tuttavia modificarne la descrizione utilizzando IAM. Per ulteriori informazioni, consulta [Modifica di un ruolo collegato ai servizi](#) nella Guida per l'utente di IAM.

## Eliminazione di un ruolo collegato al servizio per AWS IoT TwinMaker

Se non è più necessario utilizzare una funzionalità o un servizio che richiede un ruolo collegato al servizio, ti consigliamo di eliminare il ruolo. In questo modo non sarà più presente un'entità non utilizzata che non viene monitorata e gestita attivamente. Tuttavia, è necessario pulire tutte le aree di lavoro collegate a ServiceLinked che utilizzano ancora il ruolo collegato al servizio prima di poter eliminare manualmente il ruolo.

### Note

Se il AWS IoT TwinMaker servizio utilizza il ruolo quando si tenta di eliminare le risorse, l'eliminazione potrebbe non riuscire. In questo caso, attendi alcuni minuti e quindi ripeti l'operazione.

Per eliminare manualmente il ruolo collegato ai servizi mediante IAM

Utilizza la console IAM AWS CLI, o l' AWS API per eliminare il ruolo `AWSServiceRoleForIoTtwinmaker` collegato al servizio. Per ulteriori informazioni, consulta [Eliminazione del ruolo collegato al servizio](#) nella Guida per l'utente di IAM.

## Regioni supportate per i ruoli collegati ai servizi AWS IoT TwinMaker

AWS IoT TwinMaker supporta l'utilizzo di ruoli collegati al servizio in tutte le regioni in cui il servizio è disponibile. Per ulteriori informazioni, consulta [AWS Regioni ed endpoint](#).

## AWS politiche gestite per AWS IoT TwinMaker

Per aggiungere autorizzazioni a utenti, gruppi e ruoli, è più facile utilizzare le politiche AWS gestite che scrivere le politiche da soli. Creare [policy gestite dal cliente IAM](#) per fornire al tuo team solo le autorizzazioni di cui ha bisogno richiede tempo e competenza. Per iniziare rapidamente, puoi utilizzare le nostre politiche AWS gestite. Queste policy coprono i casi d'uso comuni e sono disponibili nel tuo Account AWS. Per ulteriori informazioni sulle policy AWS gestite, consulta le [policy AWS gestite](#) nella IAM User Guide.

AWS i servizi mantengono e aggiornano le politiche AWS gestite. Non è possibile modificare le autorizzazioni nelle politiche AWS gestite. I servizi occasionalmente aggiungono altre autorizzazioni a una policy gestita da AWS per supportare nuove funzionalità. Questo tipo di aggiornamento interessa tutte le identità (utenti, gruppi e ruoli) a cui è collegata la policy. È più probabile che i servizi aggiornino una policy gestita da AWS quando viene avviata una nuova funzionalità o quando diventano disponibili nuove operazioni. I servizi non rimuovono le autorizzazioni da una policy AWS gestita, quindi gli aggiornamenti delle policy non comprometteranno le autorizzazioni esistenti.

Inoltre, AWS supporta politiche gestite per le funzioni lavorative che si estendono su più servizi. Ad esempio, la policy `ReadOnlyAccess` AWS gestita fornisce l'accesso in sola lettura a tutti i AWS servizi e le risorse. Quando un servizio lancia una nuova funzionalità, AWS aggiunge autorizzazioni di sola lettura per nuove operazioni e risorse. Per l'elenco e la descrizione delle policy di funzione dei processi, consulta la sezione [Policy gestite da AWS per funzioni di processi](#) nella Guida per l'utente di IAM.

## AWS politica gestita: `AWSIoTtwinMakerServiceRolePolicy`

Non puoi collegarti `AWSIoTtwinMakerServiceRolePolicy` alle tue entità IAM. Questa policy è associata a un ruolo collegato ai servizi che consente a di eseguire operazioni per tuo conto. Per ulteriori informazioni, consulta [Autorizzazioni di ruolo collegate al servizio per AWS IoT TwinMaker](#).

La politica di autorizzazione dei ruoli denominata `AWSIoTtwinMakerServiceRolePolicy` consente di AWS IoT TwinMaker completare le seguenti azioni sulle risorse specificate:

- Operazione: `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` su 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 aggiornamenti alle politiche AWS gestite

Visualizza i dettagli sugli aggiornamenti delle politiche AWS gestite da quando questo servizio ha iniziato a tenere traccia di queste modifiche. Per gli avvisi automatici sulle modifiche apportate a questa pagina, sottoscrivi il feed RSS nella pagina della cronologia dei documenti .

Modifica	Descrizione	Data
<p><a href="#">AWSIoTtwinMakerServiceRolePolicy</a>— È stata aggiunta una politica</p>	<p>AWS IoT TwinMaker ha aggiunto la politica di autorizzazione dei ruoli denominata <code>AWSIoTtwinMakerServiceRolePolicy</code> che AWS IoT TwinMaker consente di completare le seguenti azioni sulle risorse specificate:</p> <ul style="list-style-type: none"> <li>Operazione: <code>iotsitewise:DescribeAsset</code>, <code>iotsitewise:ListAssets</code>, <code>iotsitewise:DescribeAssetModel</code>, and <code>iotsitewise:ListAssetModels</code>, <code>iottwinmaker:GetEntity</code>, <code>iottwinmaker:CreateEntity</code>, <code>iottwinmaker:UpdateEntity</code>, <code>iottwinmaker&gt;DeleteEntity</code>, <code>iottwinmaker&gt;ListEntities</code>, <code>iottwinmaker:GetComponentType</code>, <code>iottwinmaker:CreateComponentType</code>, <code>iottwinmaker:UpdateComponentType</code>, <code>iottwinmaker&gt;DeleteComponentType</code>, <code>iottwinmaker&gt;ListComponentTypes</code> su</li> </ul>	<p>17 novembre 2023</p>

Modifica	Descrizione	Data
	<p>all your iotsitewise asset and asset-model resources</p> <p>Per ulteriori informazioni, consulta <a href="#">Autorizzazioni di ruolo collegate al servizio per AWS IoT TwinMaker</a>.</p>	
ha iniziato a tenere traccia delle modifiche	ha iniziato a tenere traccia delle modifiche per le sue politiche AWS gestite.	11 maggio 2022

## AWS IoT TwinMaker e endpoint VPC di interfaccia (AWS PrivateLink)

È possibile stabilire una connessione privata tra il cloud privato virtuale (VPC) e creare un AWS IoT TwinMaker endpoint VPC di interfaccia. Gli endpoint di interfaccia sono alimentati da [AWS PrivateLink](#), che puoi utilizzare per accedere in modo privato alle AWS IoT TwinMaker API senza un gateway Internet, un dispositivo NAT (Network Address Translation), una connessione VPN o una connessione Direct Connect AWS. Le istanze nel tuo VPC non necessitano di indirizzi IP pubblici per comunicare con AWS IoT TwinMaker le API. Il traffico tra il tuo VPC e AWS IoT TwinMaker non esce dalla rete Amazon.

Ogni endpoint dell'interfaccia è rappresentato da una o più [interfacce di rete elastiche](#) nelle tue sottoreti.

Per ulteriori informazioni, consulta [Interface VPC endpoints \(AWS PrivateLink\)](#) nella Amazon VPC User Guide.

### Considerazioni sugli endpoint AWS IoT TwinMaker VPC

Prima di configurare un endpoint VPC di interfaccia per AWS IoT TwinMaker, consulta le [proprietà e le limitazioni dell'endpoint dell'interfaccia nella](#) Amazon VPC User Guide.

AWS IoT TwinMaker supporta l'esecuzione di chiamate a tutte le sue azioni API dal tuo VPC.

- Per le operazioni dell'API del piano dati, utilizza il seguente endpoint:

```
data.iottwinmaker.region.amazonaws.com
```

Le operazioni dell'API del piano dati includono quanto segue:

- [GetPropertyValue](#)
  - [GetPropertyValueHistory](#)
  - [BatchPutPropertyValues](#)
- Per le operazioni dell'API del piano di controllo, utilizzate il seguente endpoint:

```
api.iottwinmaker.region.amazonaws.com
```

Le operazioni API del piano di controllo supportate includono quanto segue:

- [CreateComponentType](#)
- [CreateEntity](#)
- [CreateScene](#)
- [CreateWorkspace](#)
- [DeleteComponentType](#)
- [DeleteEntity](#)
- [DeleteScene](#)
- [DeleteWorkspace](#)
- [GetComponentType](#)
- [GetEntity](#)
- [GetScene](#)
- [GetWorkspace](#)
- [ListComponentTypes](#)
- [ListComponentTypes](#)
- [ListEntities](#)
- [ListScenes](#)

- [ListWorkspaces](#)
- [TagResource](#)
- [UntagResource](#)
- [UpdateComponentType](#)
- [UpdateEntity](#)
- [UpdateScene](#)
- [UpdateWorkspace](#)

## Creazione di un endpoint VPC interfaccia per l' AWS IoT TwinMaker

Puoi creare un endpoint VPC per il AWS IoT TwinMaker servizio utilizzando la console Amazon VPC o (). AWS Command Line Interface AWS CLI Per ulteriori informazioni, consulta [Creazione di un endpoint dell'interfaccia](#) nella Guida per l'utente di Amazon VPC.

Crea un endpoint VPC per AWS IoT TwinMaker cui utilizzi il seguente nome di servizio.

- Per le operazioni dell'API del piano dati, utilizza il seguente nome di servizio:

```
com.amazonaws.region.iottwinmaker.data
```

- Per le operazioni dell'API del piano di controllo, utilizzate il seguente nome di servizio:

```
com.amazonaws.region.iottwinmaker.api
```

Se abiliti il DNS privato per l'endpoint, puoi effettuare richieste API a AWS IoT TwinMaker utilizzando il nome DNS predefinito per la regione, ad esempio, `iottwinmaker.us-east-1.amazonaws.com`

Per ulteriori informazioni, consulta [Accesso a un servizio tramite un endpoint dell'interfaccia](#) in Guida per l'utente di Amazon VPC.

AWS IoT TwinMaker PrivateLink è supportato nelle seguenti regioni:

- us-east-1

Il ControlPlane servizio è supportato nelle seguenti zone di disponibilità: use1-az1use1-az2, euse1-az6.



Il DataPlane servizio è supportato nelle seguenti zone di disponibilità: use1-az1, use1-az2, use1-az4.

- us-west-2

I DataPlane servizi ControlPlane e sono supportati nelle seguenti zone di disponibilità: usw2-az1, usw2-az2, eusw2-az3.

- eu-west-1
- eu-central-1
- ap-southeast-1
- ap-southeast-2

Per ulteriori informazioni sulle zone di disponibilità, vedere [ID delle zone di disponibilità per le AWS risorse - AWS Resource Access Manager](#).

## Accesso AWS IoT TwinMaker tramite un endpoint VPC di interfaccia

Quando crei un endpoint di interfaccia, AWS IoT TwinMaker genera nomi host DNS specifici dell'endpoint con cui puoi comunicare. AWS IoT TwinMaker L'opzione DNS privato è abilitata per impostazione predefinita. Per ulteriori informazioni, consulta [Using private hosted zones](#) nella Amazon VPC User Guide.

Se abiliti il DNS privato per l'endpoint, puoi effettuare richieste API AWS IoT TwinMaker tramite uno dei seguenti endpoint VPC.

- Per le operazioni dell'API del piano dati, utilizza il seguente endpoint. Sostituisci *la regione* con la tua AWS regione.

```
data.iottwinmaker.region.amazonaws.com
```

- Per le operazioni dell'API del piano di controllo, utilizza il seguente endpoint. Sostituisci *la regione* con la tua AWS regione.

```
api.iottwinmaker.region.amazonaws.com
```

Se disabiliti il DNS privato per l'endpoint, devi fare quanto segue per accedere AWS IoT TwinMaker tramite l'endpoint:

- Specificare l'URL dell'endpoint VPC nelle richieste API.
  - Per le operazioni dell'API del piano dati, utilizza il seguente URL dell'endpoint. Sostituisci una *regione* con l'ID *vpc-endpoint-id* della regione dell'endpoint VPC.

```
vpc-endpoint-id.data.iottwinmaker.region.vpce.amazonaws.com
```

- Per le operazioni dell'API del piano di controllo, utilizza il seguente URL dell'endpoint. Sostituisci una *regione* con l'ID *vpc-endpoint-id* della regione dell'endpoint VPC.

```
vpc-endpoint-id.api.iottwinmaker.region.vpce.amazonaws.com
```

- Disabilita l'iniezione del prefisso dell'host. Gli AWS CLI e AWS SDK antepongono all'endpoint del servizio vari prefissi host quando si chiama ciascuna operazione API. Ciò fa sì che gli AWS CLI e AWS SDK generino URL non validi per AWS IoT TwinMaker quando si specifica un endpoint VPC.

#### Important

Non puoi disabilitare l'iniezione del prefisso host in o. AWS CLI AWS Tools for PowerShell. Ciò significa che se hai disabilitato il DNS privato, non potrai utilizzare AWS CLI o AWS Tools for PowerShell accedere AWS IoT TwinMaker tramite l'endpoint VPC. Se desideri utilizzare questi strumenti per accedere AWS IoT TwinMaker tramite l'endpoint, abilita il DNS privato.

Per ulteriori informazioni su come disabilitare l'inserimento del prefisso dell'host negli AWS SDK, consulta le seguenti sezioni della documentazione per ciascun SDK:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java](#)
- [AWS SDK for Java 2.x](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for .NET](#)
- [AWS SDK for PHP](#)
- [AWS SDK for Python \(Boto3\)](#)
- [AWS SDK for Ruby](#)

Per ulteriori informazioni, consulta [Accesso a un servizio tramite un endpoint dell'interfaccia](#) in Guida per l'utente di Amazon VPC.

## Creazione di una policy per gli endpoint VPC per AWS IoT TwinMaker

È possibile allegare un criterio all'endpoint VPC che controlla l'accesso all' AWS IoT TwinMaker. La policy specifica le informazioni riportate di seguito:

- Il principale che può eseguire operazioni.
- Le azioni che possono essere eseguite.
- Le risorse sui cui si possono eseguire operazioni.

Per ulteriori informazioni, consulta [Controllo degli accessi ai servizi con endpoint VPC](#) in Guida per l'utente di Amazon VPC.

Esempio: policy degli endpoint VPC per le azioni AWS IoT TwinMaker

Di seguito è riportato un esempio di policy sugli endpoint per AWS IoT TwinMaker. Se collegata a un endpoint, questa policy concede l'accesso alle AWS IoT TwinMaker azioni elencate per l'utente IAM `iottwinmakeradmin` nell' AWS account `123456789012` su tutte le risorse.

```
{
  "Statement": [
    {
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:user/role"
      },
      "Resource": "*",
      "Effect": "Allow",
      "Action": [
        "iottwinmaker:CreateEntity",
        "iottwinmaker:GetScene",
        "iottwinmaker:ListEntities"
      ]
    }
  ]
}
```

# Convalida della conformità per AWS IoT TwinMaker

Per sapere se un Servizio AWS programma rientra nell'ambito di specifici programmi di conformità, consulta Servizi AWS la sezione [Scope by Compliance Program Servizi AWS](#) e scegli il programma di conformità che ti interessa. Per informazioni generali, consulta Programmi di [AWS conformità Programmi](#) di di .

È possibile scaricare report di audit di terze parti utilizzando AWS Artifact. Per ulteriori informazioni, consulta [Scaricamento dei report in AWS Artifact](#) .

La vostra responsabilità di conformità durante l'utilizzo Servizi AWS è determinata dalla sensibilità dei dati, dagli obiettivi di conformità dell'azienda e dalle leggi e dai regolamenti applicabili. AWS fornisce le seguenti risorse per contribuire alla conformità:

- [Guide introduttive su sicurezza e conformità](#): queste guide all'implementazione illustrano considerazioni sull'architettura e forniscono passaggi per implementare ambienti di base incentrati sulla AWS sicurezza e la conformità.
- [Progettazione per la sicurezza e la conformità HIPAA su Amazon Web Services](#): questo white paper descrive in che modo le aziende possono utilizzare AWS per creare applicazioni idonee all'HIPAA.

## Note

Non Servizi AWS tutte sono idonee all'HIPAA. Per ulteriori informazioni, consulta la sezione [Riferimenti sui servizi conformi ai requisiti HIPAA](#).

- [AWS Risorse per la per la conformità](#): questa raccolta di cartelle di lavoro e guide potrebbe essere valida per il tuo settore e la tua località.
- [AWS Guide alla conformità dei clienti](#): comprendi il modello di responsabilità condivisa attraverso la lente della conformità. Le guide riassumono le migliori pratiche per la protezione Servizi AWS e mappano le linee guida per i controlli di sicurezza su più framework (tra cui il National Institute of Standards and Technology (NIST), il Payment Card Industry Security Standards Council (PCI) e l'International Organization for Standardization (ISO)).
- [Valutazione delle risorse con regole](#) nella Guida per gli AWS Config sviluppatori: il AWS Config servizio valuta la conformità delle configurazioni delle risorse alle pratiche interne, alle linee guida e alle normative del settore.

- [AWS Security Hub](#)— Ciò Servizio AWS fornisce una visione completa dello stato di sicurezza interno. AWS La Centrale di sicurezza utilizza i controlli di sicurezza per valutare le risorse AWS e verificare la conformità agli standard e alle best practice del settore della sicurezza. Per un elenco dei servizi e dei controlli supportati, consulta la pagina [Documentazione di riferimento sui controlli della Centrale di sicurezza](#).
- [AWS Audit Manager](#)— Ciò Servizio AWS consente di verificare continuamente AWS l'utilizzo per semplificare la gestione dei rischi e la conformità alle normative e agli standard di settore.

## Resilienza in AWS IoT TwinMaker

L'infrastruttura AWS globale è costruita attorno a Regioni AWS zone di disponibilità. Regioni AWS forniscono più zone di disponibilità fisicamente separate e isolate, collegate con reti a bassa latenza, ad alto throughput e altamente ridondanti. Con le zone di disponibilità, puoi progettare e gestire applicazioni e database che eseguono automaticamente il failover tra zone di disponibilità senza interruzioni. Le zone di disponibilità sono più disponibili, tolleranti ai guasti e scalabili rispetto alle infrastrutture a data center singolo o multiplo tradizionali.

[Per ulteriori informazioni sulle zone di disponibilità, vedere Global Regioni AWS Infrastructure.AWS](#)

Oltre all'infrastruttura AWS globale, AWS IoT TwinMaker offre diverse funzionalità per supportare le esigenze di resilienza e backup dei dati.

## Sicurezza dell'infrastruttura in AWS IoT TwinMaker

In quanto servizio gestito, AWS IoT TwinMaker è protetto dalle procedure di sicurezza della rete AWS globale descritte nel white paper [Amazon Web Services: Overview of Security Processes](#).

Utilizzi chiamate API AWS pubblicate per accedere AWS IoT TwinMaker attraverso la rete. I client devono supportare Transport Layer Security (TLS) 1.2 o versioni successive. È consigliabile TLS 1.3 o versioni successive. I client devono, inoltre, supportare le suite di crittografia con PFS (Perfect Forward Secrecy), ad esempio Ephemeral Diffie-Hellman (DHE) o Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). La maggior parte dei sistemi moderni, come Java 7 e versioni successive, supporta tali modalità.

Inoltre, le richieste devono essere firmate utilizzando un ID chiave di accesso e una chiave di accesso segreta associata a un principale IAM. In alternativa, è possibile utilizzare [AWS Security Token Service](#) (AWS STS) per generare le credenziali di sicurezza temporanee per sottoscrivere le richieste.

# Punti finali e quote

## AWS IoT TwinMaker endpoint e quote

[È possibile trovare informazioni sugli AWS IoT TwinMaker endpoint e sulle quote nel Riferimento generale.AWS](#)

- [Per informazioni sugli endpoint di servizio, consulta AWS IoT TwinMaker Service Endpoints.](#)
- [Per informazioni sulle quote, consulta AWS IoT TwinMaker Service Quotas.](#)
- [Per informazioni sui limiti di limitazione delle API,AWS IoT TwinMaker consulta Limiti di limitazione delle API.](#)

## Informazioni aggiuntive sugli endpoint AWS IoT TwinMaker

Per connetterti a livello di codice AWS IoT TwinMaker, usa un endpoint. Se utilizzi un client HTTP, devi inserire come prefisso le API del piano di controllo e del piano dati come segue. Tuttavia, non è necessario aggiungere un prefisso all' AWS SDK e ai AWS Command Line Interface comandi perché aggiungono automaticamente il prefisso necessario.

- Utilizzate il `api` prefisso per le API del piano di controllo. Ad esempio, `api.iottwinmaker.us-west-1.amazonaws.com`.
- Usa il `data` prefisso per le API del piano dati. Ad esempio, `data.iottwinmaker.us-west-1.amazonaws.com`.

# Cronologia dei documenti per la Guida per l'utente di AWS IoT TwinMaker

La tabella seguente descrive le versioni della documentazione per AWS IoT TwinMaker.

Modifica	Descrizione	Data
<a href="#">Nuovo ruolo collegato al servizio e nuova policy IAM</a>	<p>AWS IoT TwinMaker ha aggiunto un nuovo ruolo collegato al servizio, chiamato. <a href="#">AWSServiceRoleForIoT TwinMaker</a> AWS IoT TwinMaker ha aggiunto questo nuovo ruolo collegato ai servizi per consentire di AWS IoT TwinMaker chiamare altri AWS servizi e sincronizzare le relative risorse per conto dell'utente. La nuova policy <code>AWSIoT TwinMakerServiceRolePolicy IAM</code> è associata a questo ruolo e la policy concede l'autorizzazione a chiamare altri AWS servizi e AWS IoT TwinMaker a sincronizzare le loro risorse per tuo conto.</p>	17 novembre 2023
<a href="#">Versione iniziale</a>	Versione iniziale della Guida per l'AWS IoT TwinMaker utente	30 novembre 2021

Le traduzioni sono generate tramite traduzione automatica. In caso di conflitto tra il contenuto di una traduzione e la versione originale in Inglese, quest'ultima prevarrà.