



Panduan Pengguna

AWS IoT TwinMaker



AWS IoT TwinMaker: Panduan Pengguna

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

Merek dagang dan tampilan dagang Amazon tidak boleh digunakan sehubungan dengan produk atau layanan apa pun yang bukan milik Amazon, dengan cara apa pun yang dapat menyebabkan kebingungan di antara pelanggan, atau dengan cara apa pun yang menghina atau mendiskreditkan Amazon. Semua merek dagang lain yang tidak dimiliki oleh Amazon merupakan kekayaan masing-masing pemiliknya, yang mungkin atau mungkin tidak berafiliasi, terkait dengan, atau disponsori oleh Amazon.

Table of Contents

Apa itu AWS IoT TwinMaker?	1
Cara kerjanya	1
Konsep dan komponen kunci	2
Ruang kerja	3
Model komponen entitas	4
Visualisasi	6
Memulai dengan AWS IoT TwinMaker	8
Membuat dan mengelola peran layanan untuk AWS IoT TwinMaker	9
Tetapkan kepercayaan	9
Izin Amazon S3	9
Tetapkan izin ke bucket Amazon S3 tertentu	10
Izin untuk konektor bawaan	12
Izin untuk konektor ke sumber data eksternal	15
Ubah peran IAM ruang kerja Anda untuk menggunakan konektor data Athena	16
Buat ruang kerja	18
Buat entitas pertama Anda	20
Menyiapkan AWS akun	24
Mendaftar untuk Akun AWS	24
Buat pengguna dengan akses administratif	25
Menggunakan dan membuat tipe komponen	27
Jenis komponen bawaan	27
Fitur inti dari jenis AWS IoT TwinMaker komponen	28
Membuat definisi properti	29
Membuat fungsi	30
Contoh jenis komponen	31
Alarm (abstrak)	31
Telemetry Timestream	33
Alarm (mewarisi dari alarm abstrak)	33
Contoh peralatan	34
Operasi massal	38
Konsep dan terminologi kunci	38
AWS IoT TwinMaker metadataTransferJob fungsionalitas	39
Melakukan operasi impor dan ekspor massal	40
metadataTransferJob prasyarat	41

Izin IAM	41
Jalankan operasi massal	45
Penanganan kesalahan	48
Impor templat metadata	49
AWS IoT TwinMaker metadataTransferJob contoh	53
AWS IoT TwinMaker skema pekerjaan transfer metadata	54
Konektor data	72
Konektor data	72
Konektor penginisialisasi skema	73
DataReaderByEntity	74
DataReaderByComponentType	75
DataReader	77
AttributePropertyValueReaderByEntity	78
DataWriter	79
Contoh-contoh	80
Konektor data tabular Athena	89
AWS IoT TwinMaker Prasyarat konektor data Athena	89
Menggunakan konektor data Athena	90
Menggunakan referensi JSON konektor data tabular Athena	94
Menggunakan konektor data Athena	95
Visualisasikan data tabular Athena di Grafana	96
AWS IoT TwinMaker konektor data deret waktu	97
AWS IoT TwinMaker prasyarat konektor data deret waktu	98
Latar belakang konektor data deret waktu	98
Mengembangkan konektor data deret waktu	100
Meningkatkan konektor data Anda	109
Menguji konektor Anda	110
Keamanan	110
Menciptakan AWS IoT TwinMaker sumber daya	110
Apa selanjutnya	112
AWS IoT TwinMaker konektor data pabrik cookie	112
Membuat AWS IoT TwinMaker adegan	118
Sebelum membuat adegan	118
Optimalkan sumber daya Anda sebelum mengimpornya ke AWS IoT TwinMaker	118
Praktik terbaik untuk kinerja di AWS IoT TwinMaker	119
Pelajari selengkapnya	119

Mengunggah sumber daya di AWS IoT TwinMaker	120
Unggah file ke Perpustakaan Sumber Daya menggunakan konsol	120
Buat adegan Anda	120
Gunakan navigasi 3D AWS IoT TwinMaker dalam adegan Anda	122
Tambahkan kamera tetap	124
Pengeditan yang ditingkatkan	124
Penempatan objek adegan yang ditargetkan	125
Pemilihan submodel	125
Edit entitas dalam hierarki adegan	126
Tambahkan anotasi ke entitas	126
Tambahkan overlay ke Tag	131
Edit adegan Anda	139
Tambahkan model	139
Tambahkan widget	140
Menambahkan tanda	144
Optimalkan model 3D Anda	144
Menggunakan Ubin 3D di adegan Anda	144
Adegan dinamis	147
Adegan statis versus dinamis	147
Jenis dan entitas komponen adegan	148
Konsep adegan dinamis	149
AWS IoT TwinMaker integrasi kit aplikasi	150
Beralih mode AWS IoT TwinMaker harga	151
Grafik pengetahuan	153
AWS IoT TwinMaker konsep inti grafik pengetahuan	153
Menggunakan grafik pengetahuan	154
Hasilkan grafik adegan	156
AWS IoT TwinMaker prasyarat grafik adegan	157
Mengikat node 3D di adegan Anda	158
Buat aplikasi web	160
Grafik pengetahuan Grafana panel	162
AWS IoT TwinMaker prasyarat editor kueri	162
Grafik pengetahuan Izin Grafana	163
Grafik pengetahuan sumber daya tambahan	167
Sinkronisasi aset dengan AWS IoT SiteWise	181
Menggunakan sinkronisasi aset dengan AWS IoT SiteWise	181

Menggunakan ruang kerja khusus	181
Menggunakan IoT SiteWiseDefaultWorkspace	187
Perbedaan antara ruang kerja khusus dan default	187
Sumber daya disinkronkan dari AWS IoT SiteWise	188
Ruang kerja khusus dan default	188
Hanya ruang kerja default	189
Sumber daya tidak disinkronkan	190
Gunakan entitas dan tipe komponen yang disinkronkan di AWS IoT TwinMaker	191
Menganalisis status sinkronisasi dan kesalahan	191
Sinkronkan status pekerjaan	192
Hapus pekerjaan sinkronisasi	193
Batas sinkronisasi aset	195
Menyiapkan dasbor Grafana	197
Konfigurasi CORS	198
Menyiapkan lingkungan Grafana Anda	199
Amazon Managed Grafana	199
Grafana yang dikelola sendiri	200
Membuat peran dasbor	201
Buat kebijakan IAM	201
Unggah video dari tepi	205
Tambahkan lebih banyak izin	205
Membuat peran IAM Dasbor Grafana	207
Membuat kebijakan Pemutar AWS IoT TwinMaker Video	207
Cakupan ke bawah akses ke sumber daya Anda	209
Cakupan ke bawah DAPATKAN izin	209
Cakupan AWS IoT SiteWise BatchPutAssetPropertyValue izin ke bawah	211
Hubungkan Alarm ke dasbor Grafana	213
AWS IoT SiteWise prasyarat konfigurasi alarm	213
Tentukan AWS IoT SiteWise peran IAM komponen alarm	213
Kueri dan perbarui melalui AWS IoT TwinMaker API	215
Konfigurasi dasbor Grafana Anda untuk alarm	216
Gunakan dasbor Grafana untuk visualisasi alarm	218
Integrasi Matterport	221
Ikhtisar integrasi	222
Prasyarat integrasi Matterport	223
Kredensi Matterport SDK	225

Simpan kredensial Matterport di AWS Secrets Manager	226
Matterport memindai dalam adegan AWS IoT TwinMaker	229
Matterport di dasbor AWS IoT TwinMaker Grafana Anda	235
Integrasi Matterport dengan kit aplikasi AWS IoT	235
Streaming video ke AWS IoT TwinMaker	236
Gunakan konektor tepi untuk aliran video Kinesis untuk mengalirkan video AWS IoT TwinMaker	236
Prasyarat	236
Buat komponen video untuk AWS IoT TwinMaker adegan	237
Tambahkan video dan metadata dari aliran video Kinesis ke dasbor Grafana	238
Menggunakan AWS IoT TwinMaker pustaka Flink	239
Pencatatan dan pemantauan	240
Pemantauan dengan CloudWatch metrik Amazon	240
Metrik	241
Pembuatan log panggilan API dengan AWS CloudTrail	243
AWS IoT TwinMaker informasi dalam CloudTrail	244
Keamanan	246
Perlindungan data	247
Enkripsi diam	248
Enkripsi bergerak	248
Identity and Access Management	248
Audiens	249
Mengautentikasi dengan identitas	249
Mengelola akses menggunakan kebijakan	253
Bagaimana AWS IoT TwinMaker bekerja dengan IAM	256
Contoh kebijakan berbasis identitas	263
Pemecahan Masalah	266
Menggunakan peran terkait layanan	268
AWS kebijakan terkelola	270
VPC titik akhir (AWS PrivateLink)	275
Pertimbangan untuk titik akhir AWS IoT TwinMaker VPC	275
Membuat VPC titik akhir antarmuka untuk AWS IoT TwinMaker	276
Mengakses AWS IoT TwinMaker melalui titik akhir antarmuka VPC	278
Membuat kebijakan VPC endpoint untuk AWS IoT TwinMaker	279
Validasi Kepatuhan	280
Ketangguhan	282

Keamanan Infrastruktur	282
Titik akhir dan kuota	283
AWS IoT TwinMaker titik akhir dan kuota	283
Informasi titik akhir tambahan	283
Riwayat dokumen	284
.....	cclxxxv

Apa itu AWS IoT TwinMaker?

AWS IoT TwinMaker adalah AWS IoT layanan yang dapat Anda gunakan untuk membangun kembar digital operasional sistem fisik dan digital. AWS IoT TwinMaker menciptakan visualisasi digital menggunakan pengukuran dan analisis dari berbagai sensor dunia nyata, kamera, dan aplikasi perusahaan untuk membantu Anda melacak pabrik fisik, bangunan, atau pabrik industri Anda. Anda dapat menggunakan data dunia nyata ini untuk memantau operasi, mendiagnosis dan memperbaiki kesalahan, dan mengoptimalkan operasi.

Kembar digital adalah representasi digital langsung dari suatu sistem dan semua komponen fisik dan digitalnya. Ini diperbarui secara dinamis dengan data untuk meniru struktur, keadaan, dan perilaku sistem yang sebenarnya. Anda dapat menggunakannya untuk mendorong hasil bisnis.

Pengguna akhir berinteraksi dengan data dari kembar digital Anda dengan menggunakan aplikasi antarmuka pengguna.

Cara kerjanya

Untuk memenuhi persyaratan minimum untuk membuat kembar digital, Anda harus melakukan hal berikut.

- Model perangkat, peralatan, ruang, dan proses di lokasi fisik.
- Hubungkan model ini ke sumber data yang menyimpan informasi kontekstual penting, seperti umpan kamera data sensor.
- Buat visualisasi yang membantu pengguna memahami data dan wawasan untuk membuat keputusan bisnis dengan lebih efisien.
- Buat kembar digital tersedia bagi pengguna akhir untuk mendorong hasil bisnis.

AWS IoT TwinMaker mengatasi tantangan ini dengan menyediakan kemampuan berikut.

- Grafik pengetahuan sistem komponen entitas: AWS IoT TwinMaker menyediakan alat untuk perangkat pemodelan, peralatan, ruang, dan proses dalam grafik pengetahuan.

Grafik pengetahuan ini berisi metadata tentang sistem dan dapat terhubung ke data di lokasi yang berbeda. AWS IoT TwinMaker menyediakan konektor bawaan untuk data yang disimpan di AWS IoT SiteWise dan Kinesis Video Streams. Anda juga dapat membuat konektor khusus ke data yang disimpan di lokasi lain.

Grafik pengetahuan dan konektor bersama-sama menyediakan satu antarmuka untuk menanyakan data di lokasi yang berbeda.

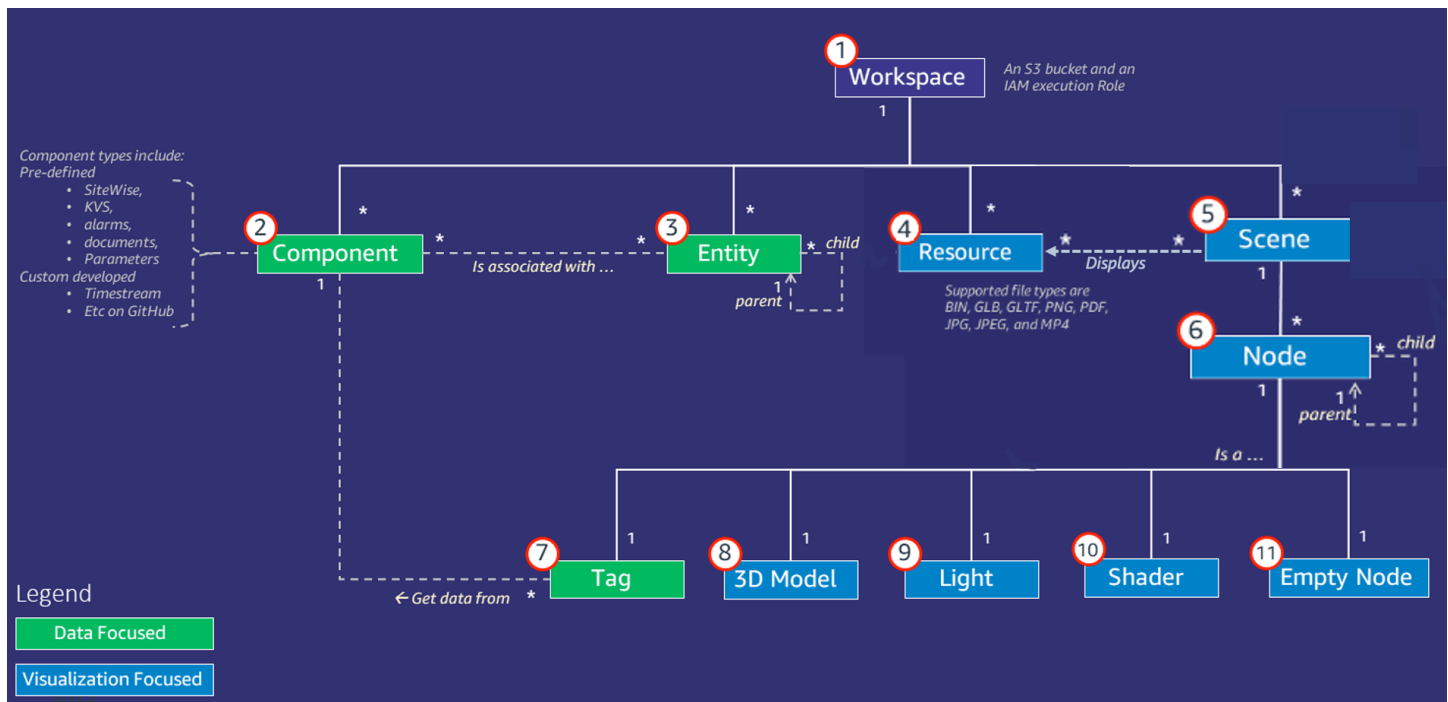
- Scene composer: AWS IoT TwinMaker Konsol menyediakan alat komposisi adegan untuk membuat adegan dalam 3D. Anda mengunggah model 3D/CAD yang dibuat sebelumnya, dioptimalkan untuk tampilan web dan dikonversi ke format.gltf atau.glb. Anda kemudian menggunakan komposer adegan untuk menempatkan beberapa model dalam satu adegan, menciptakan representasi visual dari operasi mereka.

Anda juga dapat melapisi data dalam adegan. Misalnya, Anda dapat membuat tag di lokasi adegan yang terhubung ke data suhu dari sensor. Ini mengaitkan data dengan lokasi.

- Aplikasi: AWS IoT TwinMaker menyediakan plug-in untuk Grafana dan Grafana Terkelola Amazon yang dapat Anda gunakan untuk membangun aplikasi dasbor bagi pengguna akhir.
- Alat pihak ketiga: Mendix bermitra dengan AWS IoT TwinMaker untuk memberikan solusi lengkap untuk IoT industri. Lihat lokakarya [Lean Daily Management Application dengan Mendix dan AWS IoT TwinMaker](#) untuk memulai dengan menggunakan Mendix Low Code Application Development Platform (LCAP) dengan AWS layanan seperti, Kinesis AWS IoT TwinMaker Video Streams dan. AWS IoT SiteWise

Konsep dan komponen kunci

Diagram berikut menggambarkan bagaimana konsep-konsep kunci AWS IoT TwinMaker cocok bersama.



Note

Tanda bintang (*) dalam diagram menunjukkan one-to-many hubungan. Untuk kuota untuk masing-masing hubungan ini, lihat [AWS IoT TwinMaker titik akhir dan kuota](#).

Bagian berikut menjelaskan konsep yang diilustrasikan dalam diagram.

Ruang kerja

Ruang kerja adalah wadah tingkat atas untuk aplikasi kembar digital Anda. Anda membuat kumpulan logis entitas, komponen, aset adegan, dan sumber daya lainnya untuk kembar digital Anda di dalam ruang kerja ini. Ini juga berfungsi sebagai batas keamanan untuk mengelola akses ke aplikasi kembar digital dan sumber daya yang dikandungnya. Setiap ruang kerja ditautkan ke bucket Amazon S3 tempat data ruang kerja Anda disimpan. Anda menggunakan peran IAM untuk membatasi akses ke ruang kerja Anda.

Ruang kerja dapat berisi beberapa komponen, entitas, adegan, dan sumber daya. Jenis komponen, entitas, adegan, atau sumber daya hanya ada dalam satu ruang kerja.

Model komponen entitas

AWS IoT TwinMaker menyediakan alat yang Anda gunakan untuk memodelkan sistem Anda dengan menggunakan grafik entity-component-based pengetahuan. Anda dapat menggunakan arsitektur komponen entitas untuk membuat representasi sistem fisik Anda. Model komponen entitas ini terdiri dari entitas, komponen, dan hubungan. Untuk informasi selengkapnya tentang sistem komponen entitas, lihat Sistem [komponen entitas](#).

Entitas

Entitas adalah representasi digital dari elemen dalam kembar digital yang menangkap kemampuan elemen itu. Elemen ini bisa berupa peralatan fisik, konsep, atau proses. Entitas memiliki komponen yang terkait dengannya. Komponen ini menyediakan data dan konteks untuk entitas terkait.

Dengan AWS IoT TwinMaker, Anda dapat mengatur entitas ke dalam hierarki kustom untuk manajemen yang lebih efisien. Tampilan default dari entitas dan sistem komponen adalah hierarkis.

Komponen

Komponen menyediakan konteks dan data untuk entitas dalam sebuah adegan. Anda menambahkan komponen ke entitas. Masa pakai suatu komponen terkait dengan masa pakai suatu entitas.

Komponen dapat menambahkan data statis, seperti daftar dokumen atau koordinat lokasi geografis. Mereka juga dapat memiliki fungsi yang terhubung ke sistem lain, termasuk sistem yang berisi data deret waktu seperti AWS IoT SiteWise dan sejarawan awan deret waktu lainnya.

Komponen didefinisikan oleh dokumen JSON yang menggambarkan hubungan antara sumber data dan AWS IoT TwinMaker. Komponen dapat menggambarkan sumber data eksternal atau sumber data yang dibangun di dalamnya AWS IoT TwinMaker. Komponen mengakses sumber data eksternal dengan menggunakan fungsi Lambda yang ditentukan dalam dokumen JSON. Ruang kerja dapat berisi banyak komponen. Komponen menyediakan data ke tag melalui entitas terkait.

AWS IoT TwinMaker menyediakan beberapa komponen bawaan yang dapat Anda tambahkan dari konsol. Anda juga dapat membuat komponen kustom Anda sendiri untuk terhubung ke sumber data seperti telemetri timestream dan koordinat geospasial. Contohnya termasuk TimeStream Telemetry, komponen Geospasial, dan konektor ke sumber data pihak ketiga seperti Snowflake.

AWS IoT TwinMaker menyediakan jenis komponen bawaan berikut untuk kasus penggunaan umum:

- Dokumen, seperti manual pengguna atau gambar yang terletak di URL tertentu.

- Deret waktu, seperti data sensor dari AWS IoT SiteWise.
- Alarm, seperti alarm deret waktu dari sumber data eksternal.
- Video, dari kamera IP yang terhubung ke Kinesis Video Streams.
- Komponen khusus untuk terhubung ke sumber data tambahan. Misalnya, Anda dapat membuat konektor khusus untuk menghubungkan AWS IoT TwinMaker entitas Anda ke data deret waktu yang disimpan secara eksternal.

Sumber data

Sumber data adalah lokasi data sumber kembar digital Anda. AWS IoT TwinMaker mendukung dua jenis sumber data:

- Konektor hierarki, yang memungkinkan Anda untuk terus menyinkronkan model eksternal ke AWS IoT TwinMaker
- Konektor deret waktu, yang memungkinkan Anda terhubung ke database deret waktu seperti AWS IoT SiteWise

Properti

Properti adalah nilai, baik statis maupun deret waktu yang didukung, yang terkandung dalam komponen. Saat Anda menambahkan komponen ke entitas, properti dalam komponen menjelaskan detail tentang status entitas saat ini.

AWS IoT TwinMaker mendukung tiga jenis properti:

- Nilai tunggal, non-time-series properti — Properti ini biasanya pasangan nilai kunci statis dan langsung disimpan AWS IoT TwinMaker dengan metadata entitas terkait.
- Properti deret waktu — AWS IoT TwinMaker menyimpan referensi ke toko deret waktu untuk properti ini. Ini default ke nilai terbaru.
- Properti hubungan — Properti ini menyimpan referensi ke entitas atau komponen lain. Misalnya, `seen_by` adalah komponen hubungan yang mungkin menghubungkan entitas kamera ke entitas lain yang secara langsung divisualisasikan oleh kamera itu.

Anda dapat menanyakan nilai properti di seluruh sumber data heterogen dengan menggunakan antarmuka kueri data terpadu.

Visualisasi

Anda gunakan AWS IoT TwinMaker untuk menambah representasi tiga dimensi dari kembar digital Anda, dan kemudian melihatnya di Grafana. Untuk membuat adegan, gunakan CAD yang ada atau jenis file 3D lainnya. Anda kemudian menggunakan hamparan data untuk menambahkan data yang relevan untuk kembar digital Anda.

Adegan

Adegan adalah representasi tiga dimensi yang menyediakan konteks visual untuk data yang terhubung. AWS IoT TwinMaker Adegan dapat dibuat dengan menggunakan gltf tunggal (GL Transmission Format) atau model glb 3D untuk seluruh lingkungan, atau dengan menggunakan komposisi beberapa model. Adegan juga menyertakan tag untuk menunjukkan tempat menarik dalam adegan.

Adegan adalah wadah tingkat atas untuk visualisasi. Sebuah adegan terdiri dari satu atau lebih node.

Ruang kerja dapat berisi beberapa adegan. Misalnya, ruang kerja dapat berisi satu adegan untuk setiap lantai fasilitas.

Sumber daya

Adegan menampilkan sumber daya, yang ditampilkan sebagai node di AWS IoT TwinMaker konsol. Sebuah adegan dapat berisi banyak sumber daya.

Sumber daya adalah gambar dan glTF berbasis, model tiga dimensi yang digunakan untuk membuat adegan. Sumber daya dapat mewakili satu peralatan, atau situs yang lengkap.

Anda menempatkan sumber daya ke dalam adegan dengan mengunggah file.gltf atau .glb ke pustaka sumber daya ruang kerja Anda dan kemudian menambahkannya ke adegan Anda.

Antarmuka pengguna yang diperbesar

Dengan AWS IoT TwinMaker Anda dapat menambah adegan Anda dengan hamparan data yang menambahkan konteks dan informasi penting, seperti data sensor, ke lokasi di tempat kejadian.

Node: Node adalah contoh tag, lampu, dan model tiga dimensi. Mereka juga bisa kosong untuk menambahkan struktur ke hierarki adegan Anda. Misalnya, Anda dapat mengelompokkan beberapa node bersama-sama di bawah satu simpul kosong.

Tag: Tag adalah jenis node yang mewakili data dari komponen (melalui entitas). Tag dapat dikaitkan dengan hanya satu komponen. Tag adalah anotasi yang ditambahkan ke posisi x, y, z koordinat tertentu dari sebuah adegan. Tag menghubungkan bagian adegan ini ke grafik pengetahuan dengan menggunakan properti entitas. Anda dapat menggunakan tag untuk mengonfigurasi perilaku atau tampilan visual item dalam adegan, seperti alarm.

Lampu: Anda dapat menambahkan lampu ke adegan untuk membawa objek tertentu ke dalam fokus, atau melemparkan bayangan pada objek untuk menunjukkan lokasi fisik mereka.

Model tiga dimensi: Model tiga dimensi adalah representasi visual dari file.gltf atau .glb yang diimpor sebagai sumber daya.

Note

AWS IoT TwinMaker tidak dimaksudkan untuk digunakan dalam, atau terkait dengan, pengoperasian lingkungan berbahaya atau sistem kritis apa pun yang dapat menyebabkan cedera tubuh yang serius atau kematian atau menyebabkan kerusakan lingkungan atau properti.

Data yang dikumpulkan melalui penggunaan Anda AWS IoT TwinMaker harus dievaluasi keakuratannya yang sesuai untuk kasus penggunaan Anda. AWS IoT TwinMaker tidak boleh digunakan sebagai pengganti pemantauan manusia terhadap sistem fisik untuk tujuan menilai apakah sistem tersebut beroperasi dengan aman.

Memulai dengan AWS IoT TwinMaker

Topik di bagian ini menjelaskan cara melakukan hal berikut.

- Buat dan atur ruang kerja baru.
- Buat entitas dan tambahkan komponen ke dalamnya.

Prasyarat:

Untuk membuat ruang kerja dan adegan pertama Anda, Anda memerlukan AWS sumber daya berikut.

- [Akun AWS](#).
- Peran layanan IAM untuk AWS IoT TwinMaker. Peran ini secara otomatis dihasilkan secara default, saat Anda membuat AWS IoT TwinMaker ruang kerja baru di [AWS IoT TwinMaker konsol](#).

Jika Anda tidak memilih untuk membiarkan AWS IoT TwinMaker secara otomatis membuat peran layanan IAM baru, Anda harus menentukan salah satu yang telah Anda buat.

Untuk petunjuk tentang membuat dan mengelola peran layanan ini, lihat [????](#).

Untuk informasi selengkapnya tentang peran layanan IAM, lihat [Membuat peran untuk mendelegasikan izin](#) ke. Layanan AWS

Important

Peran layanan ini harus memiliki kebijakan terlampir yang memberikan izin agar layanan membaca dan menulis ke bucket Amazon S3. AWS IoT TwinMaker menggunakan peran ini untuk mengakses layanan lain atas nama Anda. Anda juga perlu menetapkan hubungan kepercayaan antara peran ini dan AWS IoT TwinMaker agar layanan dapat mengambil peran tersebut. Jika kembar Anda berinteraksi dengan AWS layanan lain, tambahkan izin yang diperlukan untuk layanan tersebut juga.

Topik

- [Membuat dan mengelola peran layanan untuk AWS IoT TwinMaker](#)
- [Buat ruang kerja](#)

- [Buat entitas pertama Anda](#)
- [Menyiapkan AWS akun](#)

Membuat dan mengelola peran layanan untuk AWS IoT TwinMaker

AWS IoT TwinMaker mengharuskan Anda menggunakan peran layanan untuk memungkinkannya mengakses sumber daya di layanan lain atas nama Anda. Peran ini harus memiliki hubungan kepercayaan dengan AWS IoT TwinMaker. Saat membuat ruang kerja, Anda harus menetapkan peran ini ke ruang kerja. Topik ini berisi contoh kebijakan yang menunjukkan cara mengonfigurasi izin untuk skenario umum.

Tetapkan kepercayaan

Kebijakan berikut menetapkan hubungan kepercayaan antara peran Anda dan AWS IoT TwinMaker. Tetapkan hubungan kepercayaan ini ke peran yang Anda gunakan untuk ruang kerja Anda.

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

Izin Amazon S3

Kebijakan berikut memungkinkan peran Anda membaca dan menghapus dari dan menulis ke bucket Amazon S3. Ruang kerja menyimpan sumber daya di Amazon S3, sehingga izin Amazon S3 diperlukan untuk semua ruang kerja.

```
{
```

```
"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

Saat Anda membuat ruang kerja, AWS IoT TwinMaker buat file di bucket Amazon S3 yang menunjukkan bahwa file tersebut digunakan oleh ruang kerja. Kebijakan ini memberikan AWS IoT TwinMaker izin untuk menghapus file tersebut saat Anda menghapus ruang kerja. AWS IoT TwinMaker menempatkan objek lain yang terkait dengan ruang kerja Anda. Anda bertanggung jawab untuk menghapus objek ini saat Anda menghapus ruang kerja.

Tetapkan izin ke bucket Amazon S3 tertentu

Saat membuat ruang kerja di AWS IoT TwinMaker konsol, Anda dapat memilih untuk AWS IoT TwinMaker membuat bucket Amazon S3 untuk Anda. Anda dapat menemukan informasi tentang bucket ini dengan menggunakan AWS CLI perintah berikut.

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

Contoh berikut menunjukkan format output dari perintah ini.

```
{
  "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"
}
```

Untuk memperbarui kebijakan Anda sehingga menetapkan izin untuk bucket Amazon S3 tertentu, gunakan nilai nama bucket.

Kebijakan berikut memungkinkan peran Anda membaca dan menghapus dari dan menulis ke bucket Amazon S3 tertentu.

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

```

Izin untuk konektor bawaan

Jika ruang kerja Anda berinteraksi dengan AWS layanan lain menggunakan konektor bawaan, Anda harus menyertakan izin untuk layanan tersebut dalam kebijakan ini. Jika Anda menggunakan jenis komponen `com.amazon.iotsitewise.connector`, Anda harus menyertakan izin untuk `AWS IoT SiteWise Untuk informasi selengkapnya tentang jenis komponen, lihat ???`.

Note

Jika Anda berinteraksi dengan AWS layanan lain menggunakan jenis komponen kustom, Anda harus memberikan izin peran untuk menjalankan fungsi Lambda yang mengimplementasikan fungsi dalam tipe komponen Anda. Untuk informasi selengkapnya, lihat [???](#).

Contoh berikut menunjukkan cara memasukkan AWS IoT SiteWise dalam kebijakan Anda.

```

{
  "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_*"
    ]
  }
]
}

```

Jika Anda menggunakan tipe komponen `com.amazon.iotsitewise.connector` dan perlu membaca data properti dari, Anda harus menyertakan izin berikut dalam kebijakan Anda. AWS IoT SiteWise

```

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

```

```

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

```

Jika Anda menggunakan tipe komponen `com.amazon.iotsitewise.connector` dan perlu menulis data properti, Anda harus menyertakan izin berikut dalam kebijakan Anda. AWS IoT SiteWise

```

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

```

Jika Anda menggunakan jenis komponen `com.amazon.iotsitewise.connector.edgevideo`, Anda harus menyertakan izin untuk dan Kinesis Video Streams. AWS IoT SiteWise Kebijakan contoh berikut menunjukkan cara menyertakan AWS IoT SiteWise dan izin Kinesis Video Streams dalam kebijakan Anda.

```

...
{
  "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"
  },
  ...

```

Izin untuk konektor ke sumber data eksternal

Jika Anda membuat tipe komponen yang menggunakan fungsi yang terhubung ke sumber data eksternal, Anda harus memberikan izin peran layanan untuk menggunakan fungsi Lambda yang mengimplementasikan fungsi tersebut. Untuk informasi selengkapnya tentang membuat tipe dan fungsi komponen, lihat [???](#).

Contoh berikut memberikan izin ke peran layanan Anda untuk menggunakan fungsi 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_*"
    ]
  }
]
```

Untuk informasi selengkapnya tentang membuat peran dan menetapkan kebijakan serta hubungan kepercayaan kepada mereka dengan menggunakan konsol IAM, API IAM AWS CLI, dan IAM, lihat [Membuat peran untuk mendelegasikan izin ke sebuah](#). Layanan AWS

Ubah peran IAM ruang kerja Anda untuk menggunakan konektor data Athena

Untuk menggunakan [konektor data tabular AWS IoT TwinMaker Athena](#), Anda harus memperbarui peran IAM ruang kerja AWS IoT TwinMaker Anda. Tambahkan izin berikut ke peran IAM ruang kerja Anda:

Note

Perubahan IAM ini hanya berfungsi untuk data tabular Athena yang disimpan dengan dan AWS Glue Amazon S3. Untuk menggunakan Athena dengan sumber data lain, Anda harus mengonfigurasi peran IAM untuk Athena, lihat [Identitas dan manajemen akses](#) di 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
```

Baca [Identitas dan manajemen akses di Athena](#) untuk informasi lebih lanjut tentang konfigurasi IAM Athena.

Buat ruang kerja

Untuk membuat dan mengonfigurasi ruang kerja pertama Anda, gunakan langkah-langkah berikut.

Note

Topik ini menunjukkan cara membuat ruang kerja sederhana dengan satu sumber daya. Untuk ruang kerja berfitur lengkap dengan banyak sumber daya, coba pengaturan sampel di repositori [AWS IoT TwinMaker Github sampel](#).

1. Di halaman beranda [AWS IoT TwinMaker konsol](#), pilih Ruang kerja di panel navigasi kiri.
2. Pada halaman Workspaces, pilih Buat ruang kerja.
3. Pada halaman Buat Ruang Kerja, masukkan nama untuk ruang kerja Anda.
4. (Opsional) Tambahkan deskripsi untuk ruang kerja Anda.

5. Di bawah sumber daya S3, pilih Buat bucket S3. Opsi ini membuat bucket Amazon S3 tempat AWS IoT TwinMaker menyimpan informasi dan sumber daya yang terkait dengan ruang kerja. Setiap ruang kerja membutuhkan bucket sendiri.
6. Di bawah Peran eksekusi, pilih Auto-generate peran baru atau peran IAM kustom yang Anda buat untuk ruang kerja ini.

Jika Anda memilih Buat peran baru secara otomatis, AWS IoT TwinMaker lampirkan kebijakan ke peran yang memberikan izin ke peran layanan baru untuk mengakses AWS layanan lain, termasuk izin untuk membaca dan menulis ke bucket Amazon S3 yang Anda tentukan pada langkah sebelumnya. Untuk informasi tentang menetapkan izin untuk peran ini, lihat. [???](#)

7. Pilih Buat Ruang Kerja. Spanduk berikut muncul di bagian atas halaman Workspaces.



8. Pilih Dapatkan json. Kami menyarankan Anda menambahkan kebijakan IAM yang Anda lihat ke peran IAM yang AWS IoT TwinMaker dibuat untuk pengguna dan akun yang melihat dasbor Grafana. Nama peran ini mengikuti pola ini: *nama ruang kerja*DashboardRole, Untuk petunjuk tentang cara membuat kebijakan dan melampirkannya ke peran, lihat [Memodifikasi kebijakan izin peran](#) (konsol).

Contoh berikut berisi kebijakan untuk ditambahkan ke peran dasbor.

```
{
  "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*"
      ],
```

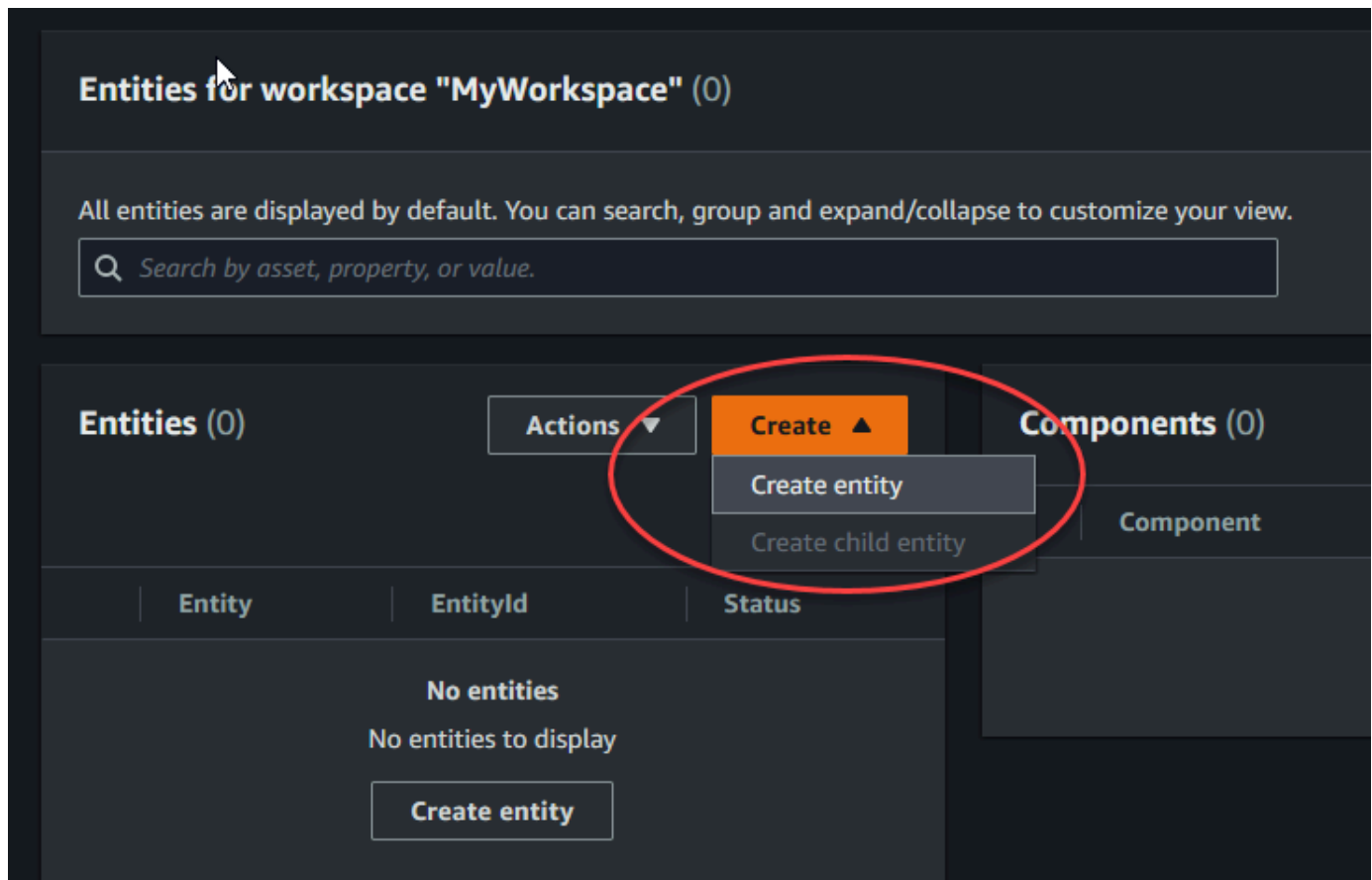
```
    "iottwinmaker:List*"
  ],
  "Resource": [
    "arn:aws:iottwinmaker:us-east-1:account-id:workspace/workspace-name",
    "arn:aws:iottwinmaker:us-east-1:account-id:workspace/workspace-name/*"
  ]
},
{
  "Effect": "Allow",
  "Action": "iottwinmaker:ListWorkspaces",
  "Resource": "*"
}
]
```

Anda sekarang siap untuk mulai membuat model data untuk ruang kerja Anda dengan entitas pertama Anda. Untuk petunjuk tentang cara melakukannya, lihat [Buat entitas pertama Anda](#).

Buat entitas pertama Anda

Untuk membuat entitas pertama Anda, gunakan langkah-langkah berikut.

1. Pada halaman Workspaces, pilih ruang kerja Anda, lalu di panel kiri pilih Entitas.
2. Pada halaman Entitas, pilih Buat, lalu pilih Buat entitas.



3. Di jendela Buat entitas, masukkan nama untuk entitas Anda. Contoh ini menggunakan **CookieMixer** entitas.
4. (Opsional) Masukkan deskripsi untuk entitas Anda.
5. Pilih Buat entitas,

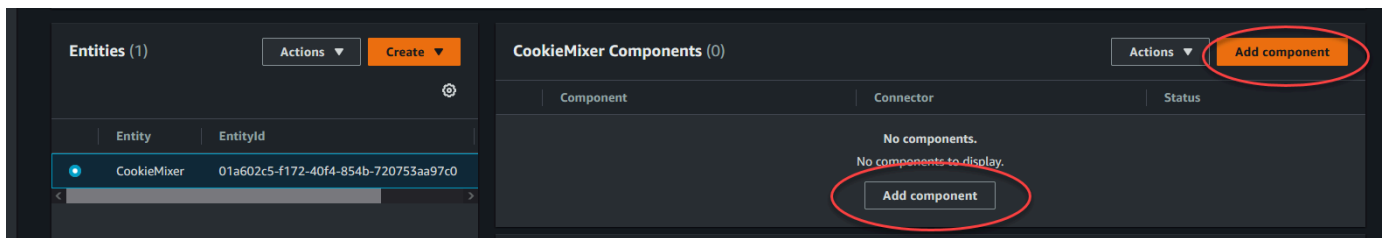
Entitas berisi data tentang setiap item di ruang kerja Anda. Anda memasukkan data ke dalam entitas dengan menambahkan komponen. AWS IoT TwinMaker menyediakan tipe komponen bawaan berikut.

- Parameter: Menambahkan satu set properti kunci-nilai.
- Dokumen: Menambahkan nama dan URL untuk dokumen yang berisi informasi tentang entitas.
- Alarm: Menghubungkan ke sumber data deret waktu alarm.
- SiteWise konektor: Menarik properti deret waktu yang didefinisikan dalam aset AWS IoT SiteWise .
- Konektor Tepi untuk Kinesis AWS IoT Greengrass Video Streams: Menarik data video dari Edge Connector untuk KVS. AWS IoT Greengrass Untuk informasi selengkapnya, lihat [AWS IoT TwinMakerIntegrasi video](#).

Anda dapat melihat jenis komponen ini dan definisinya dengan memilih tipe Komponen di panel kiri. Anda juga dapat membuat tipe komponen baru pada halaman Component types. Untuk informasi selengkapnya tentang membuat tipe komponen, lihat [Menggunakan dan membuat tipe komponen](#).

Dalam contoh ini, kami membuat komponen dokumen sederhana yang menambahkan informasi deskriptif tentang entitas Anda.

1. Pada halaman Entitas, pilih entitas, lalu pilih tambahkan komponen.



2. Di jendela Add component, masukkan nama untuk komponen Anda. Karena contoh ini menggunakan entitas cookie mixer, kita masukkan **MixerDescription** di bidang Nama.

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. Pilih Tambahkan dokumen, lalu masukkan nilai untuk Nama dokumen dan Url Eksternal. Dengan komponen dokumen, Anda dapat menyimpan daftar URL eksternal yang berisi informasi penting tentang entitas.

4. Pilih Tambahkan komponen.

Anda sekarang siap untuk membuat adegan pertama Anda. Untuk petunjuk tentang cara melakukannya, lihat [Membuat dan mengedit AWS IoT TwinMaker adegan](#).

Menyiapkan AWS akun

Jika Anda tidak memiliki Akun AWS, selesaikan langkah-langkah berikut untuk membuatnya.

Untuk mendaftar untuk Akun AWS

1. Buka <https://portal.aws.amazon.com/billing/signup>.
2. Ikuti petunjuk online.

Bagian dari prosedur pendaftaran melibatkan tindakan menerima panggilan telepon dan memasukkan kode verifikasi di keypad telepon.

Saat Anda mendaftar untuk sebuah Akun AWS, sebuah Pengguna root akun AWS dibuat. Pengguna root memiliki akses ke semua Layanan AWS dan sumber daya di akun. Sebagai praktik keamanan terbaik, tetapkan akses administratif ke pengguna, dan gunakan hanya pengguna root untuk melakukan [tugas yang memerlukan akses pengguna root](#).

Mendaftar untuk Akun AWS

Jika Anda tidak memiliki Akun AWS, selesaikan langkah-langkah berikut untuk membuatnya.

Untuk mendaftar untuk Akun AWS

1. Buka <https://portal.aws.amazon.com/billing/signup>.
2. Ikuti petunjuk online.

Bagian dari prosedur pendaftaran melibatkan tindakan menerima panggilan telepon dan memasukkan kode verifikasi di keypad telepon.

Saat Anda mendaftar untuk sebuah Akun AWS, sebuah Pengguna root akun AWS dibuat. Pengguna root memiliki akses ke semua Layanan AWS dan sumber daya di akun. Sebagai praktik keamanan terbaik, tetapkan akses administratif ke pengguna, dan gunakan hanya pengguna root untuk melakukan [tugas yang memerlukan akses pengguna root](#).

AWS mengirimkan Anda email konfirmasi setelah proses pendaftaran selesai. Anda dapat melihat aktivitas akun Anda saat ini dan mengelola akun Anda dengan mengunjungi <https://aws.amazon.com/> dan memilih Akun Saya.

Buat pengguna dengan akses administratif

Setelah Anda mendaftarkan Akun AWS, amankan Pengguna root akun AWS, aktifkan AWS IAM Identity Center, dan buat pengguna administratif sehingga Anda tidak menggunakan pengguna root untuk tugas sehari-hari.

Amankan Anda Pengguna root akun AWS

1. Masuk ke [AWS Management Console](#) sebagai pemilik akun dengan memilih pengguna Root dan memasukkan alamat Akun AWS email Anda. Di laman berikutnya, masukkan kata sandi.

Untuk bantuan masuk dengan menggunakan pengguna root, lihat [Masuk sebagai pengguna root](#) di AWS Sign-In Panduan Pengguna.

2. Mengaktifkan autentikasi multi-faktor (MFA) untuk pengguna root Anda.

Untuk petunjuk, lihat [Mengaktifkan perangkat MFA virtual untuk pengguna Akun AWS root \(konsol\) Anda](#) di Panduan Pengguna IAM.

Buat pengguna dengan akses administratif

1. Aktifkan Pusat Identitas IAM.

Untuk mendapatkan petunjuk, silakan lihat [Mengaktifkan AWS IAM Identity Center](#) di Panduan Pengguna AWS IAM Identity Center .

2. Di Pusat Identitas IAM, berikan akses administratif ke pengguna.

Untuk tutorial tentang menggunakan Direktori Pusat Identitas IAM sebagai sumber identitas Anda, lihat [Mengkonfigurasi akses pengguna dengan default Direktori Pusat Identitas IAM](#) di Panduan AWS IAM Identity Center Pengguna.

Masuk sebagai pengguna dengan akses administratif

- Untuk masuk dengan pengguna Pusat Identitas IAM, gunakan URL masuk yang dikirim ke alamat email saat Anda membuat pengguna Pusat Identitas IAM.

Untuk bantuan masuk menggunakan pengguna Pusat Identitas IAM, lihat [Masuk ke portal AWS akses](#) di Panduan AWS Sign-In Pengguna.

Tetapkan akses ke pengguna tambahan

1. Di Pusat Identitas IAM, buat set izin yang mengikuti praktik terbaik menerapkan izin hak istimewa paling sedikit.

Untuk petunjuknya, lihat [Membuat set izin](#) di Panduan AWS IAM Identity Center Pengguna.

2. Tetapkan pengguna ke grup, lalu tetapkan akses masuk tunggal ke grup.

Untuk petunjuk, lihat [Menambahkan grup](#) di Panduan AWS IAM Identity Center Pengguna.

Menggunakan dan membuat tipe komponen

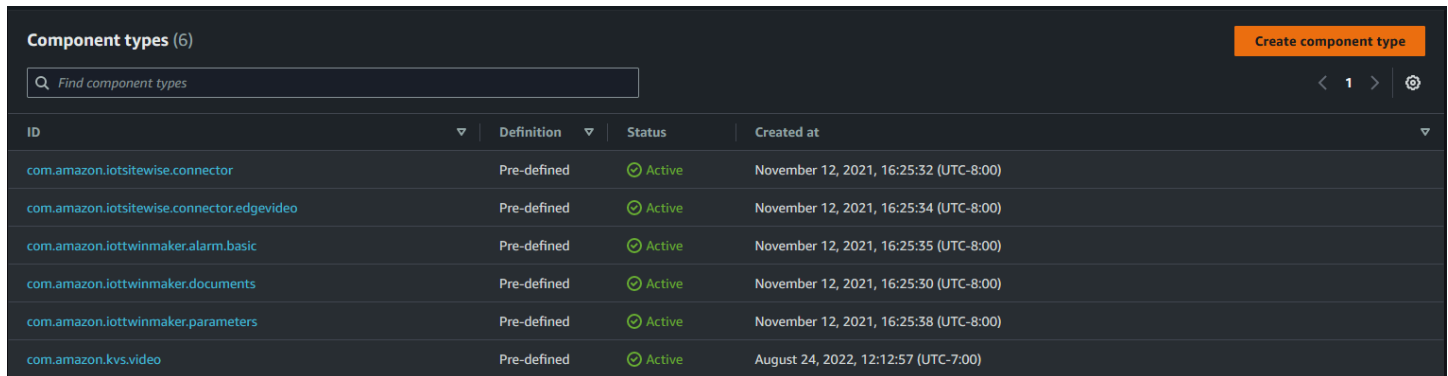
Topik ini memandu Anda melalui nilai dan struktur yang Anda gunakan untuk membuat tipe AWS IoT TwinMaker komponen. Ini menunjukkan cara membuat objek permintaan yang dapat Anda teruskan ke [CreateComponentType](#) API atau dengan menggunakan editor tipe komponen di AWS IoT TwinMaker konsol.

Komponen menyediakan konteks untuk properti dan data untuk entitas terkait mereka.

Jenis komponen bawaan

Di AWS IoT TwinMaker konsol, saat Anda memilih ruang kerja dan kemudian memilih Jenis komponen di panel kiri, Anda akan melihat jenis komponen berikut.

- `com.amazon.iotsitewise.resourcesync`: Jenis komponen yang secara otomatis menyinkronkan aset dan model aset Anda dan mengubahnya menjadi komponen entitas dan tipe komponen. AWS IoT SiteWise AWS IoT TwinMaker Untuk informasi selengkapnya tentang penggunaan sinkronisasi AWS IoT SiteWise aset, lihat [Sinkronisasi aset dengan AWS IoT SiteWise](#).
- `com.amazon.iottwinmaker.alarm.basic`: Komponen alarm dasar yang menarik data alarm dari sumber eksternal ke entitas. Komponen ini tidak berisi fungsi yang menghubungkan ke sumber data tertentu. Ini berarti bahwa komponen alarm abstrak dan dapat diwarisi oleh jenis komponen lain yang menentukan sumber data dan fungsi yang membaca dari sumber itu.
- `com.amazon.iottwinmaker.documents`: Pemetaan sederhana judul ke URL untuk dokumen yang berisi informasi tentang entitas.
- `com.amazon.iotsitewise.connector.edgevideo`: Komponen yang menarik video dari perangkat IoT menggunakan komponen Edge Connector for Kinesis Video Streams ke dalam entitas. AWS IoT Greengrass Komponen [Konektor Tepi untuk Kinesis AWS IoT Greengrass Video](#) Streams AWS IoT TwinMaker bukanlah komponen, melainkan komponen bawaan AWS IoT Greengrass yang digunakan secara lokal di perangkat IoT Anda.
- `com.amazon.iotsitewise.connector`: Komponen yang menarik data ke dalam entitas. AWS IoT SiteWise
- `com.amazon.iottwinmaker.parameters`: Komponen yang menambahkan pasangan nilai kunci statis ke entitas.
- `com.amazon.kvs.video`: Komponen yang menarik video dari Kinesis Video Streams ke entitas. 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)

Fitur inti dari jenis AWS IoT TwinMaker komponen

Daftar berikut menjelaskan fitur inti dari jenis komponen.

- **Definisi properti:** [PropertyDefinitionRequest](#) Objek mendefinisikan properti yang dapat Anda isi di komposer adegan atau dapat diisi dengan data yang ditarik dari sumber data eksternal. Properti statis yang Anda atur disimpan di AWS IoT TwinMaker. Properti deret waktu dan properti lain yang ditarik dari sumber data disimpan secara eksternal.

Anda menentukan definisi properti di dalam string ke `PropertyDefinitionRequest` peta. Setiap string harus unik untuk peta.

- **Fungsi:** [FunctionRequest](#) Objek menentukan fungsi Lambda yang membaca dari dan berpotensi menulis ke sumber data eksternal.

Tipe komponen yang berisi properti dengan nilai yang disimpan secara eksternal tetapi tidak memiliki fungsi yang sesuai untuk mengambil nilai adalah tipe komponen abstrak. Anda dapat memperluas tipe komponen konkret dari tipe komponen abstrak. Anda tidak dapat menambahkan tipe komponen abstrak ke entitas. Mereka tidak muncul di komposer adegan.

Anda menentukan fungsi di dalam string untuk `FunctionRequest` memetakan. String harus menentukan salah satu jenis fungsi standar berikut.

- `dataReader`: Fungsi yang menarik data dari sumber eksternal.
- `dataReaderByEntity`: Fungsi yang menarik data dari sumber eksternal.

Bila Anda menggunakan jenis pembaca data ini, operasi [GetPropertyValueHistory](#) API hanya mendukung kueri khusus entitas untuk properti dalam tipe komponen ini. (Anda hanya dapat meminta riwayat nilai properti untuk `componentName +entityId`.)

- `dataReaderByComponentType`: Fungsi yang menarik data dari sumber eksternal.

Saat Anda menggunakan jenis pembaca data ini, operasi [GetPropertyValueHistory](#) API hanya mendukung kueri lintas entitas untuk properti dalam tipe komponen ini. (Anda hanya dapat meminta riwayat nilai properti untuk `componentTypeId`.)

- `dataWriter`: Fungsi yang menulis data ke sumber eksternal.
- `schemaInitializer`: Fungsi yang secara otomatis menginisialisasi nilai properti setiap kali Anda membuat entitas yang berisi tipe komponen.

Salah satu dari tiga jenis fungsi pembaca data diperlukan dalam tipe komponen non-abstrak.

[Untuk contoh fungsi Lambda yang mengimplementasikan komponen telemetri aliran waktu, termasuk alarm, lihat pembaca data di Sampel.AWS IoT TwinMaker](#)

Note

Karena konektor alarm mewarisi dari jenis komponen alarm abstrak, fungsi Lambda harus mengembalikan `alarm_key` nilainya. Jika Anda tidak mengembalikan nilai ini, Grafana tidak akan mengenalinya sebagai alarm. Ini diperlukan untuk semua komponen yang mengembalikan alarm.

- Warisan: Jenis komponen mempromosikan penggunaan kembali kode melalui pewarisan. Tipe komponen dapat mewarisi hingga 10 jenis komponen induk.

Gunakan `extendsFrom` parameter untuk menentukan tipe komponen dari mana tipe komponen Anda mewarisi properti dan fungsi.

- `isSingleton`: Beberapa komponen berisi properti, seperti koordinat lokasi, yang tidak dapat disertakan lebih dari sekali dalam entitas. Tetapkan nilai `isSingleton` parameter `true` untuk menunjukkan bahwa tipe komponen Anda hanya dapat disertakan sekali dalam entitas.

Membuat definisi properti

Tabel berikut menjelaskan parameter dari `aPropertyDefinitionRequest`.

Parameter	Deskripsi
<code>isExternalId</code>	Boolean yang menentukan apakah properti adalah pengidentifikasi unik (seperti Id AWS

Parameter	Deskripsi
	IoT SiteWise aset) dari nilai properti yang disimpan secara eksternal. Nilai default properti ini adalah <code>false</code> .
<code>isStoredExternally</code>	Sebuah Boolean yang menentukan apakah nilai properti disimpan secara eksternal. Nilai default properti ini adalah <code>false</code> .
<code>isTimeSeries</code>	Boolean yang menentukan apakah properti menyimpan data deret waktu. Nilai default properti ini adalah <code>false</code> .
<code>isRequiredInEntity</code>	Boolean yang menentukan apakah properti harus memiliki nilai dalam entitas yang menggunakan tipe komponen.
<code>dataType</code>	DataType Objek yang menentukan tipe data (seperti string, peta, daftar, dan satuan ukuran) dari properti.
<code>defaultValue</code>	Sebuah DataValue objek yang menentukan nilai default dari properti.
<code>configuration</code>	string-to-string Peta yang menentukan informasi tambahan yang perlu Anda sambungkan ke sumber data eksternal.

Membuat fungsi

Tabel berikut menjelaskan parameter dari `aFunctionRequest`.

Parameter	Deskripsi
<code>implementedBy</code>	DataConnector Objek yang menentukan fungsi Lambda yang menghubungkan ke sumber data eksternal.
<code>requiredProperties</code>	Daftar properti yang dibutuhkan fungsi untuk membaca dan menulis ke sumber data eksternal.
<code>scope</code>	Ruang lingkup fungsi. Gunakan <code>Workspace</code> untuk fungsi dengan cakupan yang mencakup seluruh ruang kerja. Gunakan <code>Entity</code> untuk fungsi dengan cakupan yang terbatas pada entitas yang berisi komponen.

Untuk contoh yang menunjukkan cara membuat dan memperluas tipe komponen, lihat [???](#).

Contoh jenis komponen

Topik ini berisi contoh yang menunjukkan bagaimana menerapkan konsep kunci dari tipe komponen.

Alarm (abstrak)

Contoh berikut adalah jenis komponen alarm abstrak yang muncul di AWS IoT TwinMaker konsol. Ini berisi `functions` daftar yang terdiri dari `dataReader` yang tidak memiliki `implementedBy` nilai.

```
{
  "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
  }
}
```

Catatan:

Nilai untuk `componentTypeId` dan `workspaceID` diperlukan. Nilai `componentTypeId` harus unik untuk ruang kerja Anda. Nilai `alarm_key` adalah pengidentifikasi unik yang dapat digunakan fungsi untuk mengambil data alarm dari sumber eksternal. Nilai kunci diperlukan dan disimpan di AWS IoT TwinMaker. Nilai deret `alarm_status` waktu disimpan di sumber eksternal.

Lebih banyak contoh tersedia di [AWS IoT TwinMaker Sampel](#).

Telemetri Timestream

Contoh berikut adalah jenis komponen sederhana yang mengambil data telemetri tentang jenis komponen tertentu (seperti alarm atau mixer cookie) dari sumber eksternal. Ini menentukan fungsi Lambda yang jenis komponen mewarisi.

```
{
  "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
    }
  }
}
```

Alarm (mewarisi dari alarm abstrak)

Contoh berikut mewarisi dari kedua alarm abstrak dan tipe komponen telemetri timestream. Ini menentukan fungsi Lambda sendiri yang mengambil data alarm.

```
{
  "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

Karena konektor alarm mewarisi dari jenis komponen alarm abstrak, fungsi Lambda harus mengembalikan `alarm_key` nilainya. Jika Anda tidak mengembalikan nilai ini, Grafana tidak akan mengenalinya sebagai alarm. Ini diperlukan untuk semua komponen yang mengembalikan alarm.

Contoh peralatan

Contoh di bagian ini menunjukkan bagaimana memodelkan peralatan potensial. Anda dapat menggunakan contoh-contoh ini untuk mendapatkan beberapa ide tentang cara memodelkan peralatan dalam proses Anda sendiri.

Mixer kue

Contoh berikut mewarisi dari tipe komponen telemetry timestream. Ini menentukan properti deret waktu tambahan untuk tingkat rotasi dan suhu mixer cookie.

```
{
  "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
    }
  }
}
```

Tangki air

Contoh berikut mewarisi dari tipe komponen telemetry timestream. Ini menentukan sifat deret waktu tambahan untuk volume dan laju aliran tangki air.

```
{
  "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
  }
}
```

Lokasi ruang

Contoh berikut berisi properti, nilai-nilai yang disimpan dalam AWS IoT TwinMaker. Karena nilai ditentukan oleh pengguna dan disimpan secara internal, tidak ada fungsi yang diperlukan untuk mengambilnya. Contoh ini juga menggunakan tipe RELATIONSHIP data untuk menentukan hubungan dengan tipe komponen lain.

Komponen ini menyediakan mekanisme ringan untuk menambahkan konteks ke kembar digital. Anda dapat menggunakannya untuk menambahkan metadata yang menunjukkan di mana sesuatu berada. Anda juga dapat menggunakan informasi ini dalam logika yang digunakan untuk menentukan kamera mana yang dapat melihat peralatan atau ruang, atau untuk mengetahui cara mengirim seseorang ke suatu lokasi.


```
{
  "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 operasi massal

Gunakan `metadataTransferJob` untuk mentransfer dan mengelola AWS IoT TwinMaker sumber daya Anda dalam skala besar. `metadataTransferJob` memungkinkan Anda untuk melakukan operasi massal dan mentransfer sumber daya antara AWS IoT TwinMaker dan AWS IoT SiteWise dan Amazon S3.

Anda dapat menggunakan operasi massal dalam skenario berikut:

- Migrasi massal aset dan data antar akun, misalnya migrasi dari akun pengembangan ke akun produksi.
- Manajemen aset skala besar, seperti mengunggah, dan mengedit AWS IoT aset dalam skala besar.
- Impor massal aset Anda ke dalam AWS IoT TwinMaker dan AWS IoT SiteWise.
- Impor massal AWS IoT TwinMaker entitas dari file ontologi yang ada seperti `revit` atau BIM file.

Topik

- [Konsep dan terminologi kunci](#)
- [Melakukan operasi impor dan ekspor massal](#)
- [AWS IoT TwinMaker skema pekerjaan transfer metadata](#)

Konsep dan terminologi kunci

AWS IoT TwinMaker operasi massal menggunakan konsep dan terminologi berikut:

- **Impor:** Tindakan memindahkan sumber daya ke AWS IoT TwinMaker ruang kerja. Misalnya, dari file lokal, file di bucket Amazon S3, atau dari AWS IoT SiteWise ke ruang kerja. AWS IoT TwinMaker
- **Ekspor:** Tindakan memindahkan sumber daya dari AWS IoT TwinMaker ruang kerja ke mesin lokal atau bucket Amazon S3.
- **Sumber:** Lokasi awal dari tempat Anda ingin memindahkan sumber daya.

Misalnya, bucket Amazon S3 adalah sumber impor, dan AWS IoT TwinMaker ruang kerja adalah sumber ekspor.

- **Tujuan:** Lokasi yang diinginkan di mana Anda ingin memindahkan sumber daya Anda.

Misalnya, bucket Amazon S3 adalah tujuan ekspor, dan AWS IoT TwinMaker ruang kerja adalah tujuan impor.

- AWS IoT SiteWise Skema: Skema yang digunakan untuk mengimpor dan mengekspor sumber daya ke dan dari. AWS IoT SiteWise
- AWS IoT TwinMaker Skema: Skema yang digunakan untuk mengimpor dan mengekspor sumber daya ke dan dari. AWS IoT TwinMaker
- AWS IoT TwinMaker sumber daya tingkat atas: Sumber daya yang digunakan di API yang ada. Secara khusus, Entitas atau a ComponentType.
- AWS IoT TwinMaker sumber daya sub-level: Jenis sumber daya bersarang yang digunakan dalam definisi metadata. Secara khusus, sebuah komponen.
- Metadata: Informasi kunci yang diperlukan untuk berhasil mengimpor atau mengekspor AWS IoT SiteWise dan AWS IoT TwinMaker sumber daya.
- metadataTransferJob: Objek yang dibuat saat Anda menjalankan `CreateMetadataTransferJob`.

AWS IoT TwinMaker metadataTransferJob fungsionalitas

Topik ini menjelaskan perilaku AWS IoT TwinMaker berikut ketika Anda menjalankan operasi massal — bagaimana a metadataTransferJob diproses. Ini juga menjelaskan cara mendefinisikan skema dengan metadata yang diperlukan untuk mentransfer sumber daya Anda. AWS IoT TwinMaker operasi massal mendukung fungsionalitas berikut:

- Sumber daya tingkat atas membuat atau mengganti: AWS IoT TwinMaker akan membuat sumber daya baru atau mengganti semua sumber daya yang ada yang diidentifikasi secara unik oleh ID sumber daya.

Misalnya, jika entitas ada dalam sistem, definisi entitas akan diganti dengan yang baru didefinisikan dalam template di bawah Entity kunci.

- Sub-sumber daya membuat atau mengganti:

Dari EntityComponent level, Anda hanya dapat membuat atau mengganti komponen. Entitas harus sudah ada, jika tidak, tindakan akan menghasilkan a ValidationException.

Dari tingkat properti atau hubungan, Anda hanya dapat membuat atau mengganti properti atau hubungan, dan yang mengandung EntityComponent harus sudah ada.

- Penghapusan sub-sumber daya:

AWS IoT TwinMaker juga mendukung penghapusan sub-sumber daya. Sub-sumber daya dapat berupa komponen, properti, atau hubungan.

Jika Anda ingin menghapus komponen, Anda harus melakukannya dari tingkat entitas.

Jika Anda ingin menghapus properti atau hubungan, Anda harus melakukannya dari Entitas atau EntityComponent level.

Untuk menghapus sub-sumber daya, Anda memperbarui sumber daya tingkat yang lebih tinggi dan menghilangkan definisi sub-sumber daya.

- Tidak ada penghapusan sumber daya tingkat atas: tidak AWS IoT TwinMaker akan pernah menghapus sumber daya tingkat atas. Sumber daya tingkat atas mengacu pada entitas atau ComponentType.
- Tidak ada definisi sub-sumber daya untuk sumber daya tingkat atas yang sama dalam satu templat:

Anda tidak dapat memberikan definisi entitas lengkap dan definisi sub-sumber daya (seperti properti) dari entitas yang sama dalam templat yang sama.

Jika EntityId digunakan di Entity, Anda tidak dapat menggunakan ID yang sama di Entitas, properti EntityComponent, atau hubungan.

Jika kombinasi EntityId atau componentName digunakan EntityComponent, Anda tidak dapat menggunakan kombinasi yang sama dalam EntityComponent, properti, atau hubungan.

Jika kombinasi EntityId, componentName, propertyName digunakan dalam properti atau hubungan, Anda tidak dapat menggunakan kombinasi yang sama dalam properti atau hubungan.

- ExternalId adalah opsional untuk AWS IoT TwinMaker: ExternalId Dapat digunakan untuk membantu Anda mengidentifikasi sumber daya Anda.

Melakukan operasi impor dan ekspor massal

Topik ini mencakup cara melakukan operasi impor dan ekspor massal dan cara menangani kesalahan dalam pekerjaan transfer Anda. Ini memberikan contoh pekerjaan transfer menggunakan perintah CLI.

Referensi AWS IoT TwinMaker API berisi informasi tentang tindakan API [CreateMetadataTransferJob](#) dan lainnya.

Topik

- [metadataTransferJob prasyarat](#)
- [Izin IAM](#)
- [Jalankan operasi massal](#)
- [Penanganan kesalahan](#)
- [Impor templat metadata](#)
- [AWS IoT TwinMaker metadataTransferJob contoh](#)

metadataTransferJob prasyarat

Harap lengkapi prasyarat berikut sebelum Anda menjalankan: metadataTransferJob

- Buat AWS IoT TwinMaker ruang kerja. Ruang kerja dapat menjadi tujuan impor atau sumber ekspor untuk file. metadataTransferJob Untuk informasi tentang membuat ruang kerja lihat, [Buat ruang kerja](#).
- Buat bucket Amazon S3 untuk menyimpan sumber daya. Untuk informasi lebih lanjut tentang menggunakan Amazon S3 lihat, [Apa itu Amazon S3?](#)

Izin IAM

Saat Anda melakukan operasi massal, Anda perlu membuat kebijakan IAM dengan izin untuk memungkinkan pertukaran AWS sumber daya antara Amazon S3 AWS IoT TwinMaker, AWS IoT SiteWise, dan mesin lokal Anda. Untuk informasi selengkapnya tentang membuat kebijakan IAM, lihat [Membuat kebijakan IAM](#).

Pernyataan kebijakan untuk AWS IoT TwinMaker, AWS IoT SiteWise dan Amazon S3 tercantum di sini:

- AWS IoT TwinMaker kebijakan:

```
{
  "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 kebijakan:

```

{
  "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": "*"
  }
]
}

```

- Kebijakan Amazon S3:

```

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

```

```

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

```

Atau, Anda dapat mencakup kebijakan Amazon S3 untuk hanya mengakses satu bucket Amazon S3, lihat kebijakan berikut.

Kebijakan cakupan ember tunggal 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/*"
  ]
}

```

Mengatur kontrol akses untuk metadataTransferJob

Untuk mengontrol jenis pekerjaan yang dapat diakses pengguna, tambahkan kebijakan IAM berikut ke peran yang digunakan untuk memanggil AWS IoT TwinMaker.

Note

Kebijakan ini hanya mengizinkan akses ke pekerjaan AWS IoT TwinMaker impor dan ekspor yang mentransfer sumber daya ke dan dari Amazon S3.


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

Jalankan operasi massal

Bagian ini mencakup cara melakukan operasi impor dan ekspor massal.

Impor data dari Amazon S3 ke AWS IoT TwinMaker

1. Tentukan sumber daya yang ingin Anda transfer menggunakan AWS IoT TwinMaker metadataTransferJob skema. Buat dan simpan file skema Anda di bucket Amazon S3 Anda.

Misalnya skema, lihat [Impor templat metadata](#).

2. Buat badan permintaan dan simpan sebagai file JSON. Badan permintaan menentukan sumber dan tujuan untuk pekerjaan transfer. Pastikan untuk menentukan bucket Amazon S3 Anda sebagai sumber dan AWS IoT TwinMaker ruang kerja Anda sebagai tujuan.

Berikut ini adalah contoh badan permintaan:

```
{
  "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"
      }
    }
  }
}

```

Rekam nama file yang Anda berikan pada badan permintaan Anda, Anda akan membutuhkannya di langkah berikutnya. Dalam contoh ini badan permintaan diberi `namacreateMetadataTransferJobImport.json`.

3. Jalankan perintah CLI berikut untuk memanggil `CreateMetadataTransferJob` (ganti nama file `input-json` dengan nama yang Anda berikan pada badan permintaan Anda):

```

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

```

Ini menciptakan `metadataTransferJob` dan memulai proses mentransfer sumber daya yang Anda pilih.

Ekspor data dari AWS IoT TwinMaker ke Amazon S3

1. Buat badan permintaan JSON dengan filter yang sesuai untuk memilih sumber daya yang ingin Anda ekspor. Untuk contoh ini kami menggunakan:

```

{
  "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"
    }
  }
}

```

`filtersArray` memungkinkan Anda menentukan sumber daya mana yang akan diekspor. Dalam contoh ini kita memfilter dengan `entity`, dan `componentType`.

Pastikan untuk menentukan AWS IoT TwinMaker ruang kerja Anda sebagai sumber dan bucket Amazon S3 Anda sebagai tujuan tugas transfer metadata.

Simpan badan permintaan Anda dan catat nama file, Anda akan membutuhkannya di langkah berikutnya. Untuk contoh ini, kami menamai badan permintaan `kamcreateMetadataTransferJobExport.json`.

2. Jalankan perintah CLI berikut untuk memanggil `CreateMetadataTransferJob` (ganti nama file `input-json` dengan nama yang Anda berikan pada badan permintaan Anda):

```

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

```

Ini menciptakan `metadataTransferJob` dan memulai proses mentransfer sumber daya yang Anda pilih.

Untuk memeriksa atau memperbarui status pekerjaan transfer, gunakan perintah berikut:

- Untuk membatalkan pekerjaan, gunakan tindakan [CancelMetadataTransferJobAPI](#). Saat Anda memanggil `CancelMetadataTransferJob`, API hanya membatalkan proses `metadataTransferJob`, dan sumber daya apa pun yang sudah diekspor atau diimpor tidak terpengaruh oleh panggilan API ini.
- Untuk mengambil informasi tentang pekerjaan tertentu, gunakan tindakan [GetMetadataTransferJobAPI](#).

Atau, Anda dapat memanggil `GetMetadataTransferJob` pekerjaan transfer yang ada dengan perintah CLI berikut:

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

Jika Anda memanggil `GetMetadataTransferJob` pekerjaan AWS IoT TwinMaker impor atau ekspor yang tidak ada, Anda mendapatkan `ResourceNotFoundException` kesalahan sebagai tanggapan.

- Untuk membuat daftar pekerjaan saat ini, gunakan tindakan [ListMetadataTransferJobsAPI](#).

Berikut adalah contoh CLI yang memanggil sebagai `ListMetadataTransferJobs` `destinationType` dan AWS IoT TwinMaker sebagai `SourceType: s3`

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

Note

Anda dapat mengubah nilai untuk parameter `SourceType` dan `DestinationType` agar sesuai dengan sumber dan tujuan pekerjaan impor atau ekspor Anda.

Untuk lebih banyak contoh perintah CLI yang menjalankan tindakan API ini, lihat [AWS IoT TwinMaker metadataTransferJob contoh](#)

Jika Anda menemukan kesalahan selama pekerjaan transfer, lihat [Penanganan kesalahan](#).

Penanganan kesalahan

Setelah Anda membuat dan menjalankan pekerjaan transfer, Anda dapat menelepon `GetMetadataTransferJob` untuk mendiagnosis kesalahan yang terjadi:

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

Setelah Anda melihat status pekerjaan beralih keCOMPLETED, Anda dapat memverifikasi hasil pekerjaan. GetMetadataTransferJob mengembalikan sebuah objek [MetadataTransferJobProgress](#) yang disebut yang berisi bidang-bidang berikut:

- FailedCount: Menunjukkan jumlah sumber daya yang gagal selama proses transfer.
- SkippedCount: Menunjukkan jumlah sumber daya yang dilewati selama proses transfer.
- SucceededCount: Menunjukkan jumlah sumber daya yang berhasil selama proses transfer.
- TotalCount: Menunjukkan jumlah total sumber daya yang terlibat dalam proses transfer.

Selain itu, elemen ReportUrl dikembalikan yang berisi URL yang telah ditandatangani sebelumnya. Jika pekerjaan transfer Anda memiliki kesalahan yang ingin Anda selidiki lebih lanjut, maka Anda dapat mengunduh laporan kesalahan lengkap menggunakan URL ini.

Impor templat metadata

Anda dapat mengimpor banyak komponen, ComponentTypes, atau entitas dengan satu operasi impor massal. Contoh di bagian ini menunjukkan bagaimana melakukan ini.

template: Importing entities

Gunakan format templat berikut untuk pekerjaan yang mengimpor entitas:

```
{  
  "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

Gunakan format templat berikut untuk pekerjaan yang mengimpor 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

Gunakan format template berikut untuk pekerjaan yang mengimpor komponen:

```
{
  "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 contoh

Gunakan perintah berikut untuk mengelola transfer metadata Anda:

- [CreateMetadataTransferJob](#) Tindakan API.

Contoh perintah CLI:

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

- Untuk membatalkan pekerjaan, gunakan tindakan [CancelMetadataTransferJob](#) API.

Contoh perintah CLI:

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

Saat Anda menelepon `CancelMetadataTransferJob`, itu hanya membatalkan pekerjaan transfer metadata tertentu, dan sumber daya apa pun yang sudah diekspor atau diimpor tidak terpengaruh.

- Untuk mengambil informasi tentang pekerjaan tertentu, gunakan tindakan [GetMetadataTransferJob](#) API.

Contoh perintah CLI:

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

```

- Untuk membuat daftar pekerjaan saat ini, gunakan tindakan [ListMetadataTransferJobs](#) API.

Anda dapat memfilter hasil yang dikembalikan dengan `ListMetadataTransferJobs` menggunakan file JSON. Lihat prosedur berikut menggunakan CLI:

1. Buat file JSON input CLI untuk menentukan filter yang ingin Anda gunakan:

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

```

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

```

Simpan dan rekam nama file, Anda akan membutuhkannya saat memasukkan perintah CLI.

- Gunakan file JSON sebagai argumen untuk perintah CLI berikut:

```

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

```

AWS IoT TwinMaker skema pekerjaan transfer metadata

metadataTransferJob skema impor: Gunakan skema AWS IoT TwinMaker metadata ini untuk memvalidasi data saat Anda mengunggahnya ke 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/DefaultValue"
  },
  "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
}
}
```



```
}
```

Berikut adalah contoh yang membuat `ComponentType` baru yang dipanggil `component.type.initial` dan menciptakan entitas yang disebut `initial`:

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

Berikut adalah contoh yang memperbarui entitas yang ada:

```
{
  "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 konektor data

AWS IoT TwinMaker menggunakan arsitektur berbasis konektor sehingga Anda dapat menghubungkan data dari penyimpanan data Anda sendiri ke AWS IoT TwinMaker. Ini berarti Anda tidak perlu memigrasi data sebelum menggunakan AWS IoT TwinMaker. Saat ini, AWS IoT TwinMaker mendukung konektor pihak pertama untuk AWS IoT SiteWise. Jika Anda menyimpan data pemodelan dan properti AWS IoT SiteWise, maka Anda tidak perlu mengimplementasikan konektor Anda sendiri. Jika Anda menyimpan data pemodelan atau properti Anda di penyimpanan data lain, seperti Timestream, DynamoDB, atau Snowflake, maka Anda harus menggunakan AWS Lambda untuk mengimplementasikan konektor dengan AWS IoT TwinMaker antarmuka konektor data AWS IoT TwinMaker sehingga dapat memanggil konektor Anda bila diperlukan.

Topik

- [AWS IoT TwinMaker konektor data](#)
- [AWS IoT TwinMaker Konektor data tabular Athena](#)
- [Mengembangkan AWS IoT TwinMaker konektor data deret waktu](#)

AWS IoT TwinMaker konektor data

Konektor memerlukan akses ke penyimpanan data dasar Anda untuk menyelesaikan kueri terkirim dan mengembalikan hasil atau kesalahan.

Untuk mempelajari tentang konektor yang tersedia, antarmuka permintaan mereka, dan antarmuka responsnya, lihat topik berikut.

Untuk informasi selengkapnya tentang properti yang digunakan dalam antarmuka konektor, lihat tindakan [GetPropertyValueHistory](#) API.

Note

Beberapa konektor memiliki dua bidang stempel waktu di antarmuka permintaan dan respons untuk properti waktu mulai dan waktu akhir. Keduanya `startTime` dan `endTime` menggunakan angka panjang untuk mewakili epoch second, yang tidak lagi didukung. Untuk mempertahankan kompatibilitas mundur, kami masih mengirimkan nilai stempel waktu ke bidang itu, tetapi sebaiknya gunakan `start` dan `end` bidang yang konsisten dengan format stempel waktu API kami.

Topik

- [Konektor penginisialisasi skema](#)
- [DataReaderByEntity](#)
- [DataReaderByComponentType](#)
- [DataReader](#)
- [AttributePropertyValueReaderByEntity](#)
- [DataWriter](#)
- [Contoh-contoh](#)

Konektor penginisialisasi skema

Anda dapat menggunakan penginisialisasi skema dalam tipe komponen atau siklus hidup entitas untuk mengambil tipe komponen atau properti komponen dari sumber data yang mendasarinya. Penginisialisasi skema secara otomatis mengimpor tipe komponen atau properti komponen tanpa secara eksplisit memanggil tindakan API untuk disiapkan. `properties`

SchemaInitializer antarmuka permintaan

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

Note

Peta properti dalam antarmuka permintaan ini adalah `filePropertyRequest`. Untuk informasi lebih lanjut, lihat [PropertyRequest](#).

SchemaInitializer antarmuka respons

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

Note

Peta properti dalam antarmuka permintaan ini adalah `filePropertyResponse`. Untuk informasi lebih lanjut, lihat [PropertyResponse](#).

DataReaderByEntity

`DataReaderByEntity` adalah konektor bidang data yang digunakan untuk mendapatkan nilai deret waktu properti dalam satu komponen.

Untuk informasi tentang jenis properti, sintaks, dan format konektor ini, lihat tindakan [GetPropertyValueHistoryAPI](#).

DataReaderByEntity antarmuka permintaan

```
{
  "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"
}

```

DataReaderByEntityantarmuka respons

```

{
  "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

Untuk mendapatkan nilai deret waktu dari properti umum yang berasal dari tipe komponen yang sama, gunakan konektor DataReaderByEntity bidang data. Misalnya, jika Anda menentukan properti deret waktu dalam tipe komponen dan memiliki beberapa komponen yang menggunakan tipe komponen tersebut, Anda dapat menanyakan properti tersebut di semua komponen dalam rentang waktu tertentu. Kasus penggunaan umum untuk ini adalah ketika Anda ingin menanyakan status alarm beberapa komponen untuk tampilan global entitas Anda.

Untuk informasi tentang jenis properti, sintaks, dan format konektor ini, lihat tindakan [GetPropertyValueHistory](#) API.

DataReaderByComponentType antarmuka permintaan

```
{
  "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 antarmuka respons

```
{
  "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 adalah konektor bidang data yang dapat menangani kasus DataReaderByEntity dan DataReaderByComponentType.

Untuk informasi tentang jenis properti, sintaks, dan format konektor ini, lihat tindakan [GetPropertyValueHistory](#) API.

DataReader antarmuka permintaan

Itu EntityId dan componentName bersifat opsional.

```
{
  "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 antarmuka respons

```
{
  "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 adalah konektor bidang data yang dapat Anda gunakan untuk mengambil nilai properti statis dalam satu entitas.

Untuk informasi tentang jenis properti, sintaks, dan format konektor ini, lihat tindakan [GetPropertyValueAPI](#).

AttributePropertyValueReaderByEntity antarmuka permintaan

```

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

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

```

AttributePropertyValueReaderByEntity antarmuka respons

```

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

```

```
}  
}
```

DataWriter

DataWriter adalah konektor bidang data yang dapat Anda gunakan untuk menulis titik data deret waktu kembali ke penyimpanan data yang mendasarinya untuk properti dalam satu komponen.

Untuk informasi tentang jenis properti, sintaks, dan format konektor ini, lihat tindakan [BatchPutPropertyValuesAPI](#).

DataWriter antarmuka permintaan

```
{  
  "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 antarmuka respons

```
{
```

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

Contoh-contoh

Contoh JSON berikut adalah contoh respon dan sintaks permintaan untuk beberapa konektor.

- SchemaInitializer:

Contoh berikut menunjukkan penginisialisasi skema dalam siklus hidup tipe komponen.

Permintaan:

```
{  
  "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"
    }
}
}
}
}

```

Tanggapan:

```

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

```

```
}
```

- Penginisialisasi skema dalam siklus hidup entitas:

Permintaan:

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

Tanggapan:

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

- **DataReaderByEntity** dan **DataReader**:

Permintaan:

```
{
  "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
  }
}
}
}

```

Tanggapan:

```

{
  "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:

Permintaan:

```
{
  "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
    }
  }
}
```

Tanggapan:

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

- **DataWriter:**

Permintaan:

```
{
  "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
        }
      }
    ]
  }
]
}

```

Tanggapan:

```

{
  "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 Konektor data tabular Athena

Dengan konektor data tabular Athena, Anda dapat mengakses dan menggunakan penyimpanan data Athena Anda di AWS IoT TwinMaker. Anda dapat menggunakan data Athena Anda untuk membangun kembar digital tanpa upaya migrasi data yang intensif. Anda dapat menggunakan konektor bawaan atau membuat konektor Athena khusus untuk mengakses data dari sumber data Athena Anda.

AWS IoT TwinMaker Prasyarat konektor data Athena

Sebelum Anda menggunakan konektor data tabular Athena, lengkapi prasyarat berikut:

- Buat tabel Athena terkelola dan sumber daya Amazon S3 yang terkait. [Untuk informasi tentang penggunaan Athena, lihat dokumentasi Athena.](#)
- Buat AWS IoT TwinMaker ruang kerja. Anda dapat membuat ruang kerja di [AWS IoT TwinMaker konsol](#).
- Perbarui peran IAM ruang kerja Anda dengan izin Athena. Untuk informasi selengkapnya, lihat [Ubah peran IAM ruang kerja Anda untuk menggunakan konektor data Athena](#).
- Menjadi akrab dengan AWS IoT TwinMaker sistem komponen entitas dan cara membuat entitas. Untuk informasi selengkapnya, lihat [Buat entitas pertama Anda](#).

- Menjadi akrab dengan AWS IoT TwinMaker konektor data. Untuk informasi selengkapnya, lihat [AWS IoT TwinMaker konektor data](#).

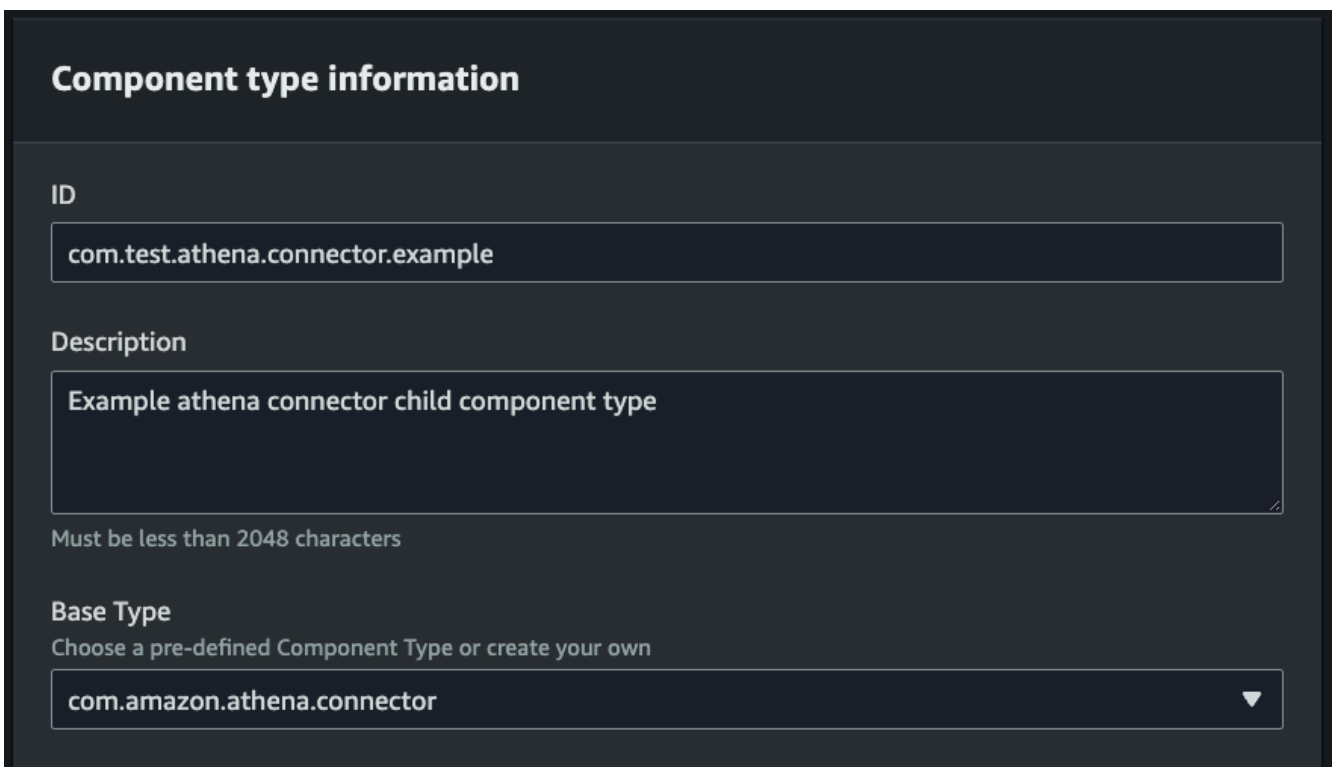
Menggunakan konektor data Athena

Untuk menggunakan konektor data Athena, Anda harus membuat komponen, menggunakan konektor Athena sebagai tipe komponen. Kemudian Anda melampirkan komponen ke entitas dalam adegan Anda untuk digunakan dalam AWS IoT TwinMaker.

Buat tipe komponen dengan konektor data Athena

Gunakan prosedur ini untuk membuat tipe AWS IoT TwinMaker komponen dengan konektor data tabular Athena:

1. Navigasikan ke [konsol AWS IoT TwinMaker](#) tersebut.
2. Buka ruang kerja yang ada atau [buat yang baru](#).
3. Dari menu navigasi sisi kiri, pilih Jenis komponen, dan pilih Buat tipe komponen untuk membuka halaman pembuatan tipe komponen.
4. Pada halaman Buat tipe komponen, isi kolom ID dengan ID yang cocok dengan kasus penggunaan Anda.



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. Pilih tipe Base. Dari daftar dropdown, pilih konektor data tabular Athena yang diberi label sebagai `com.amazon.athena.connector`.
6. Konfigurasi sumber data tipe komponen dengan memilih sumber daya Athena untuk bidang berikut:
 - Pilih sumber data Athena.
 - Pilih database Athena.
 - Pilih nama Tabel.
 - Pilih Athena WorkGroup.
7. Setelah Anda memilih sumber daya Athena yang ingin Anda gunakan sebagai sumber data, pilih kolom mana dari tabel yang ingin Anda sertakan.
8. Pilih nama kolom ID Eksternal. Pilih kolom dari langkah sebelumnya untuk dijadikan kolom ID eksternal. Id eksternal adalah id yang digunakan untuk mewakili aset Athena dan memetakannya ke entitas AWS IoT TwinMaker .

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. (Opsional) Tambahkan AWS tag ke sumber daya ini, sehingga Anda dapat mengelompokkan dan mengaturnya.
10. Pilih Buat tipe komponen untuk menyelesaikan pembuatan tipe komponen.

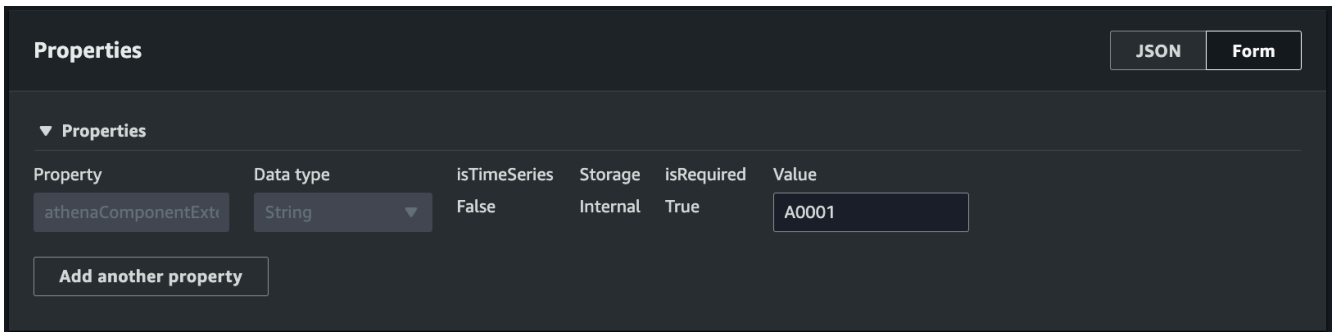
Buat komponen dengan tipe konektor data Athena dan lampirkan ke entitas

Gunakan prosedur ini untuk membuat AWS IoT TwinMaker komponen dengan konektor data tabular Athena dan melampirkannya ke entitas:

 Note

Anda harus memiliki tipe komponen yang ada yang menggunakan konektor data tabular Athena sebagai sumber data untuk menyelesaikan prosedur ini. Lihat prosedur sebelumnya [Buat tipe komponen dengan konektor data Athena](#) sebelum memulai panduan ini.

1. Navigasikan ke [konsol AWS IoT TwinMaker](#) tersebut.
2. Buka ruang kerja yang ada atau [buat yang baru](#).
3. Dari menu navigasi sisi kiri pilih Entitas, dan pilih entitas yang ingin Anda tambahkan komponen atau buat entitas baru.
4. [Buat entitas baru](#).
5. Selanjutnya pilih Tambahkan komponen. , isi kolom Nama komponen dengan nama yang cocok dengan kasus penggunaan Anda.
6. Dari menu tarik-turun tipe Komponen pilih ID tipe komponen yang Anda buat di prosedur sebelumnya.
7. Masukkan informasi Komponen, Nama Komponen, dan pilih anak yang ComponentType dibuat sebelumnya. Ini adalah yang ComponentType Anda buat dengan konektor data Athena.
8. Di bagian Properties, masukkan athenaComponentExternalId untuk komponen.



- Pilih Tambahkan komponen untuk menyelesaikan pembuatan komponen.

Anda sekarang telah berhasil membuat komponen dengan konektor data Athena sebagai tipe komponen dan melampirkannya ke entitas.

Menggunakan referensi JSON konektor data tabular Athena

Contoh berikut adalah referensi JSON lengkap untuk konektor data tabular Athena. Gunakan ini sebagai sumber daya untuk membuat konektor data kustom dan tipe komponen.

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

Menggunakan konektor data Athena

Anda dapat memunculkan entitas Anda yang menggunakan tabel Athena di Grafana. Untuk informasi selengkapnya, lihat Integrasi [dasbor AWS IoT TwinMaker Grafana](#).

Baca [dokumentasi Athena](#) untuk informasi tentang membuat dan menggunakan tabel Athena untuk menyimpan data.

Memecahkan masalah konektor data Athena

Topik ini mencakup masalah umum yang mungkin Anda temui saat mengonfigurasi konektor data Athena.

Lokasi workgroup Athena:

Saat membuat konektor Athena ComponentType, workgroup Athena harus memiliki pengaturan lokasi keluaran. Lihat [Cara kerja kelompok kerja](#).

Izin peran IAM tidak ada:

Peran AWS IoT TwinMaker; ruang kerja mungkin kehilangan izin akses API Athena saat membuat ComponentType, menambahkan komponen Ca ke entitas, atau menjalankan API. GetPropertyValue Untuk memperbarui izin IAM, lihat [Membuat dan mengelola peran layanan](#).
AWS IoT TwinMaker

Visualisasikan data tabular Athena di Grafana

Plugin Grafana juga tersedia untuk memvisualisasikan data tabular Anda di Grafana panel dasbor dengan fitur tambahan seperti penyortiran dan pemfilteran berdasarkan properti yang dipilih tanpa melakukan panggilan API ke, atau interaksi dengan Athena. AWS IoT TwinMaker Topik ini menunjukkan cara mengonfigurasi Grafana untuk memvisualisasikan data tabular Athena.

Prasyarat

Sebelum mengonfigurasi panel Grafana untuk memvisualisasikan data tabular Athena, tinjau prasyarat berikut:

- Anda telah menyiapkan lingkungan Grafana. Untuk informasi lebih lanjut lihat, Integrasi [AWS IoT TwinMaker Grafana](#).
- Anda dapat mengonfigurasi sumber data Grafana. Untuk informasi lebih lanjut lihat, [Grafana AWS IoT TwinMaker](#).
- Anda terbiasa membuat dasbor baru dan menambahkan panel baru.

Visualisasikan data tabular Athena di Grafana

Prosedur ini menunjukkan kepada Anda cara mengatur panel Grafana untuk memvisualisasikan data tabular Athena.

1. Buka dasbor AWS IoT TwinMaker Grafana Anda.
2. Pilih panel Tabel di pengaturan panel.
3. Pilih sumber data Anda dalam konfigurasi kueri.
4. Pilih kueri Dapatkan Nilai Properti.
5. Pilih entitas.
6. Pilih komponen yang memiliki ComponentType yang memperluas tipe komponen dasar Athena.

7. Pilih grup properti dari tabel Athena Anda.
8. Pilih sejumlah properti dari grup properti.
9. Konfigurasi kondisi tabel melalui daftar filter dan pesanan properti. Dengan opsi berikut:
 - Filter: tentukan ekspresi untuk nilai properti untuk memfilter data Anda.
 - OrderBy: tentukan apakah data harus dikembalikan dalam urutan naik atau turun untuk properti.

The screenshot displays a table with the following data:

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

Below the table, the query configuration is shown:

- 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

Mengembangkan AWS IoT TwinMaker konektor data deret waktu

Bagian ini menjelaskan cara mengembangkan konektor data deret waktu dalam suatu step-by-step proses. Selain itu, kami menyajikan contoh konektor data deret waktu berdasarkan seluruh sampel pabrik cookie, yang mencakup model 3D, entitas, komponen, alarm, dan konektor. Sumber sampel pabrik cookie tersedia di [GitHub repositori AWS IoT TwinMaker sampel](#).

Topik

- [AWS IoT TwinMaker prasyarat konektor data deret waktu](#)
- [Latar belakang konektor data deret waktu](#)
- [Mengembangkan konektor data deret waktu](#)
- [Meningkatkan konektor data Anda](#)
- [Menguji konektor Anda](#)
- [Keamanan](#)
- [Menciptakan AWS IoT TwinMaker sumber daya](#)
- [Apa selanjutnya](#)
- [AWS IoT TwinMaker contoh pabrik cookie konektor deret waktu](#)

AWS IoT TwinMaker prasyarat konektor data deret waktu

Sebelum mengembangkan konektor data deret waktu Anda, kami sarankan Anda menyelesaikan tugas-tugas berikut:

- Buat [AWS IoT TwinMaker ruang kerja](#).
- Buat [tipe AWS IoT TwinMaker komponen](#).
- Buat [AWS IoT TwinMaker entitas](#).
- (Opsional) Baca [Menggunakan dan membuat tipe komponen](#).
- (Opsional) Baca [antarmuka konektor AWS IoT TwinMaker data](#) untuk mendapatkan pemahaman umum tentang konektor AWS IoT TwinMaker data.

Note

Untuk contoh konektor yang diimplementasikan sepenuhnya, lihat implementasi contoh pabrik cookie kami.

Latar belakang konektor data deret waktu

Bayangkan Anda bekerja dengan pabrik yang memiliki satu set mixer kue dan tangki air. Anda ingin membangun kembar AWS IoT TwinMaker digital dari entitas fisik ini sehingga Anda dapat memantau status operasional mereka dengan memeriksa berbagai metrik deret waktu.

Anda telah menyiapkan sensor di tempat dan Anda sudah mengalirkan data pengukuran ke database Timestream. Anda ingin dapat melihat dan mengatur data pengukuran AWS IoT TwinMaker dengan overhead minimal. Anda dapat menyelesaikan tugas ini dengan menggunakan konektor data deret waktu. Gambar berikut menunjukkan contoh tabel telemetri, yang diisi melalui penggunaan konektor deret waktu.

Rows returned (1000+)
Results are paginated. Scroll through the result pages to see more query results.

Filter

TelemetryAssetId	TelemetryAssetType	measure_name	time	measure_value:varchar	measure_value:double
Mixer_22_680b5b8e-1afe-4a77-87ab-834f8e5ba01e	Mixer	Temperature	2022-04-19 00:28:00.241000000	-	99.1292877197266
Mixer_20_0568f25f-116c-429c-a974-5ceec065a6ac	Mixer	RPM	2022-04-19 00:28:00.241000000	-	59.4233207702637
Mixer_22_680b5b8e-1afe-4a77-87ab-834f8e5ba01e	Mixer	RPM	2022-04-19 00:28:00.241000000	-	59.9421195983887
Mixer_24_7ff0b75b-f0fa-43f0-bc89-b96337586d00	Mixer	Temperature	2022-04-19 00:28:00.241000000	-	99.1292877197266
Mixer_25_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-d6f3-4804-9fe1-7d2abca82d0	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

[Kumpulan data dan tabel Timestream yang digunakan dalam tangkapan layar ini tersedia di repositori sampel.AWS IoT TwinMaker GitHub](#) Lihat juga [konektor contoh pabrik cookie](#) untuk implementasi, yang menghasilkan hasil yang ditunjukkan pada tangkapan layar sebelumnya.

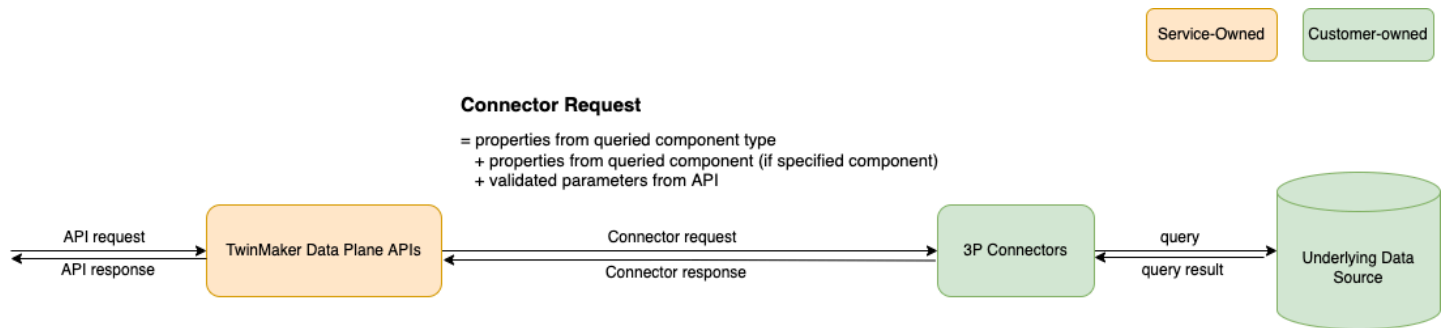
Aliran data konektor data deret waktu

Untuk kueri bidang data, AWS IoT TwinMaker ambil properti yang sesuai dari komponen dan tipe komponen dari definisi komponen dan tipe komponen. AWS IoT TwinMaker meneruskan properti ke AWS Lambda fungsi bersama dengan parameter kueri API apa pun dalam kueri.

AWS IoT TwinMaker menggunakan fungsi Lambda untuk mengakses dan menyelesaikan kueri dari sumber data dan mengembalikan hasil kueri tersebut. Fungsi Lambda menggunakan properti komponen dan tipe komponen dari bidang data untuk menyelesaikan permintaan awal.

Hasil kueri Lambda dipetakan ke respons API dan dikembalikan kepada Anda.

AWS IoT TwinMaker mendefinisikan antarmuka konektor data dan menggunakannya untuk berinteraksi dengan fungsi Lambda. Dengan menggunakan konektor data, Anda dapat melakukan kueri sumber data dari AWS IoT TwinMaker API tanpa upaya migrasi data apa pun. Gambar berikut menguraikan aliran data dasar yang dijelaskan dalam paragraf sebelumnya.



Mengembangkan konektor data deret waktu

Prosedur berikut menguraikan model pengembangan yang secara bertahap dibangun hingga konektor data deret waktu fungsional. Langkah-langkah dasarnya adalah sebagai berikut:

1. Buat tipe komponen dasar yang valid

Dalam tipe komponen, Anda menentukan properti umum yang dibagikan di seluruh komponen Anda. Untuk mempelajari lebih lanjut tentang mendefinisikan tipe komponen, lihat [Menggunakan dan membuat tipe komponen](#).

AWS IoT TwinMaker menggunakan [pola pemodelan entitas-komponen](#) sehingga setiap komponen dilampirkan ke entitas. Kami menyarankan Anda memodelkan setiap item fisik sebagai entitas dan memodelkan sumber data yang berbeda dengan tipe komponennya sendiri.

Contoh berikut menunjukkan tipe komponen template Timestream dengan satu properti:

```

{"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
    }
}
}

```

Elemen kunci dari jenis komponen adalah sebagai berikut:

- `telemetryIdProperti` mengidentifikasi kunci unik dari item fisik di sumber data yang sesuai. Konektor data menggunakan properti ini sebagai kondisi filter untuk hanya menanyakan nilai yang terkait dengan item yang diberikan. Selain itu, jika Anda menyertakan nilai `telemetryId` properti dalam respons API bidang data, maka sisi klien mengambil ID dan dapat melakukan pencarian terbalik jika perlu.
- `lambdaArnBidang` mengidentifikasi fungsi Lambda yang digunakan tipe komponen.
- `isRequiredInEntityBendera` memberlakukan pembuatan ID. Bendera ini diperlukan sehingga ketika komponen dibuat, ID item juga dipakai.
- Ditambahkan ke tipe komponen sebagai id eksternal sehingga item dapat diidentifikasi dalam tabel Timestream. `TelemetryId`

2. Buat komponen dengan tipe komponen

Untuk menggunakan tipe komponen yang Anda buat, Anda harus membuat komponen dan melampirkannya ke entitas tempat Anda ingin mengambil data. Langkah-langkah berikut merinci proses pembuatan komponen itu:

- a. Navigasikan ke [konsol AWS IoT TwinMaker](#) tersebut.

- b. Pilih dan buka ruang kerja yang sama tempat Anda membuat tipe komponen.
 - c. Arahkan ke halaman entitas.
 - d. Buat entitas baru atau pilih entitas yang ada dari tabel.
 - e. Setelah Anda memilih entitas yang ingin Anda gunakan, pilih Add component untuk membuka halaman Add component.
 - f. Beri komponen nama dan untuk Type, pilih jenis komponen yang Anda buat dengan template di 1. Buat tipe komponen dasar yang valid.
3. Buat tipe komponen Anda memanggil konektor Lambda

Konektor Lambda perlu mengakses sumber data dan menghasilkan pernyataan kueri berdasarkan input dan meneruskannya ke sumber data. Contoh berikut menunjukkan template permintaan JSON yang melakukan hal ini.

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

Elemen kunci dari permintaan:

- `selectedProperties` ini adalah daftar yang Anda isi dengan properti yang Anda inginkan pengukuran Timestream.
- Bidang `startDateTime`, `endDateTime`, dan `endTime` bidang menentukan rentang waktu untuk permintaan tersebut. Ini menentukan rentang sampel untuk pengukuran yang dikembalikan.
- `entityId` ini adalah nama entitas dari mana Anda menanyakan data.
- `componentName` ini adalah nama komponen dari mana Anda menanyakan data.
- Gunakan `orderByTime` bidang untuk mengatur urutan di mana hasil ditampilkan.

Dalam permintaan contoh sebelumnya, kami berharap untuk mendapatkan serangkaian sampel untuk properti yang dipilih selama jendela waktu yang diberikan untuk item yang diberikan, dengan urutan waktu yang dipilih. Pernyataan tanggapan dapat diringkas sebagai berikut:

```
{
  "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. Perbarui jenis komponen Anda agar memiliki dua properti

Template JSON berikut menunjukkan tipe komponen yang valid dengan dua properti:

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

```
}
```

5. Perbarui konektor Lambda untuk menangani properti kedua

API bidang AWS IoT TwinMaker data mendukung kueri beberapa properti dalam satu permintaan, dan AWS IoT TwinMaker mengikuti satu permintaan ke konektor dengan menyediakan `daftarselectedProperties`.

Permintaan JSON berikut menunjukkan template yang dimodifikasi yang sekarang mendukung permintaan untuk dua properti.

```
{
  "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
    }
}
}
}

```

Demikian pula, respons yang sesuai juga diperbarui, seperti yang ditunjukkan pada contoh berikut:

```

{
  "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

Dalam hal pagination untuk kasus ini, ukuran halaman dalam permintaan berlaku untuk semua properti. Ini berarti bahwa dengan lima properti dalam kueri dan ukuran halaman 100, jika ada cukup titik data di sumber, Anda harus mengharapkan untuk melihat 100 titik data per properti, dengan total 500 titik data.

Untuk contoh implementasi, lihat [Sampel konektor Snowflake](#) pada GitHub

Meningkatkan konektor data Anda

Menangani pengecualian

Aman bagi konektor Lambda untuk melempar pengecualian. Dalam panggilan API bidang data, AWS IoT TwinMaker layanan menunggu fungsi Lambda mengembalikan respons. Jika implementasi konektor memunculkan pengecualian, AWS IoT TwinMaker menerjemahkan jenis pengecualian menjadi `ConnectorFailure`, membuat klien API sadar bahwa masalah terjadi di dalam konektor.

Penanganan pagination

Dalam contoh, Timestream menyediakan [fungsi utilitas](#) yang dapat membantu mendukung pagination secara native. Namun, untuk beberapa antarmuka kueri lainnya, seperti SQL, mungkin perlu upaya ekstra untuk menerapkan algoritma pagination yang efisien. Ada contoh konektor [Snowflake](#) yang menangani pagination dalam antarmuka SQL.

Ketika token baru dikembalikan AWS IoT TwinMaker melalui antarmuka respons konektor, token dienkripsi sebelum dikembalikan ke klien API. Ketika token disertakan dalam permintaan lain, AWS

IoT TwinMaker dekripsi sebelum meneruskannya ke konektor Lambda. Kami menyarankan Anda menghindari menambahkan informasi sensitif ke token.

Menguji konektor Anda

Meskipun Anda masih dapat memperbarui implementasi setelah Anda menautkan konektor ke jenis komponen, kami sangat menyarankan Anda memverifikasi konektor Lambda sebelum mengintegrasikan dengan AWS IoT TwinMaker

Ada beberapa cara untuk menguji konektor Lambda Anda: Anda dapat menguji konektor Lambda di konsol Lambda atau secara lokal di AWS CDK

[Untuk informasi selengkapnya tentang pengujian fungsi Lambda, lihat Menguji fungsi Lambda dan aplikasi pengujian Lokal. AWS CDK](#)

Keamanan

Untuk dokumentasi tentang praktik terbaik keamanan dengan Timestream, lihat [Keamanan di Timestream](#).

Untuk contoh pencegahan injeksi SQL, lihat skrip [AWS IoT TwinMaker Python](#) berikut di Repositori Sampel. GitHub

Menciptakan AWS IoT TwinMaker sumber daya

Setelah menerapkan fungsi Lambda, Anda dapat membuat AWS IoT TwinMaker sumber daya seperti tipe komponen, entitas, dan komponen melalui [AWS IoT TwinMaker konsol atau API](#).

Note

Jika Anda mengikuti instruksi pengaturan dalam GitHub sampel, semua AWS IoT TwinMaker sumber daya tersedia secara otomatis. Anda dapat memeriksa definisi jenis komponen dalam [AWS IoT TwinMaker GitHub sampel](#). Setelah tipe komponen digunakan oleh komponen apa pun, definisi properti dan fungsi dari tipe komponen tidak dapat diperbarui.

Pengujian integrasi

Sebaiknya lakukan pengujian terintegrasi AWS IoT TwinMaker untuk memverifikasi kueri bidang data berfungsi end-to-end. Anda dapat melakukannya melalui [GetPropertyValueHistory](#) API atau dengan mudah di [AWS IoT TwinMaker konsol](#).

AWS IoT TwinMaker > Workspaces > CookieFactory > Entities > Mixer_22 > MixerComponent

MixerComponent

Delete Edit

Component information

Name
MixerComponent

Type
com.example.cookiefactory.mixer

Status
ACTIVE

Properties JSON Test

Test connector

Run test

Select the properties from "MixerComponent" and a time range to test for time-series properties.

Timeseries properties (max 10 supported)

- Rpm
- Temperature

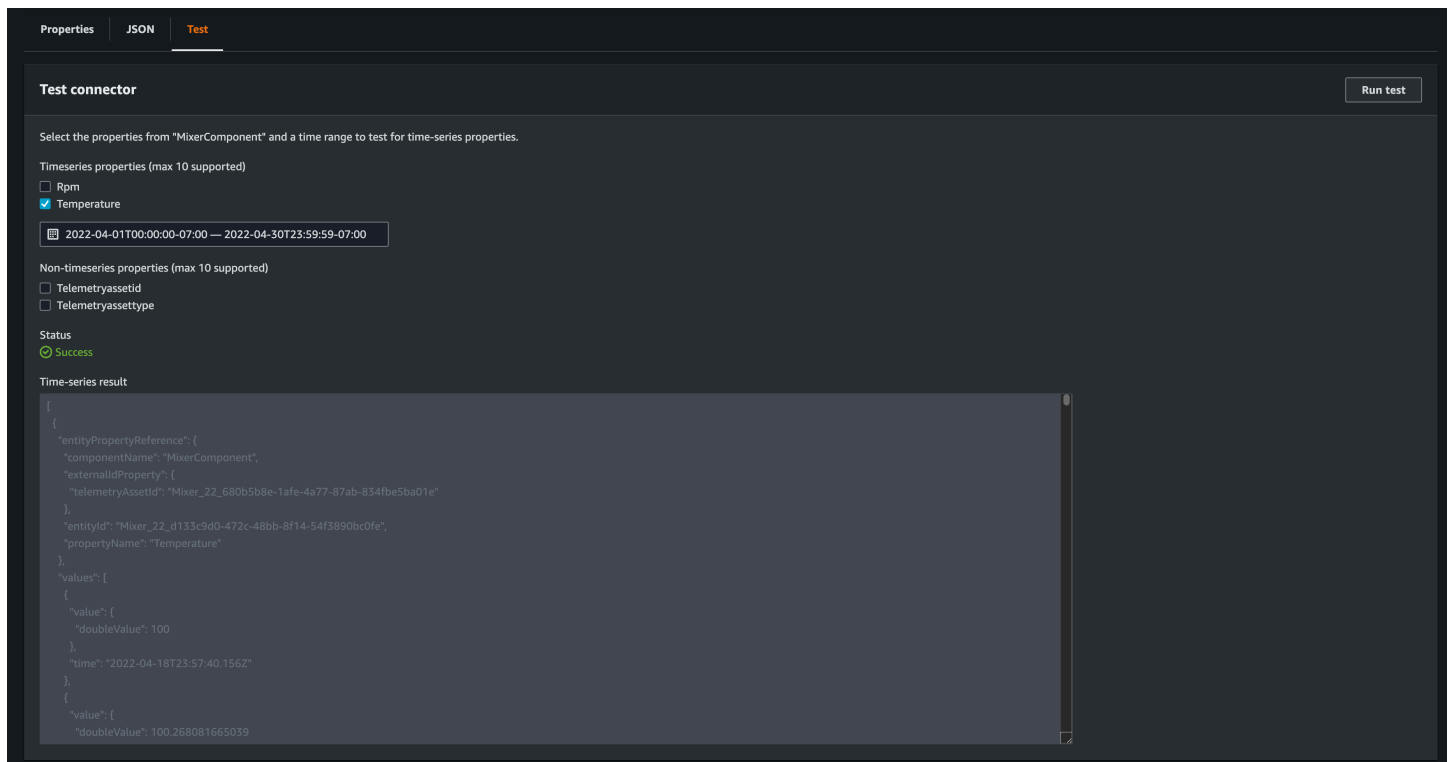
Filter by a date and time range

Non-timeseries properties (max 10 supported)

- Telemetryassetid
- Telemetryassettype

Status
⊖

Di AWS IoT TwinMaker konsol, buka detail komponen dan kemudian di bawah Test, Anda akan melihat semua properti dalam komponen terdaftar di sana. Area Uji konsol memungkinkan Anda menguji properti deret waktu serta non-time-series properti. Untuk properti deret waktu, Anda juga dapat menggunakan [GetPropertyValueHistory](#) API dan untuk non-time-series properti menggunakan [GetPropertyValue](#) API. Jika konektor Lambda Anda mendukung beberapa kueri properti, Anda dapat memilih lebih dari satu properti.



Apa selanjutnya

Anda sekarang dapat mengatur [dasbor AWS IoT TwinMaker Grafana](#) untuk memvisualisasikan metrik. Anda juga dapat menjelajahi sampel konektor data lainnya di [GitHub repositori AWS IoT TwinMaker sampel](#) untuk melihat apakah mereka sesuai dengan kasus penggunaan Anda.

AWS IoT TwinMaker contoh pabrik cookie konektor deret waktu

[Kode lengkap fungsi Lambda pabrik cookie](#) tersedia di GitHub. Meskipun Anda masih dapat memperbarui implementasi setelah Anda menautkan konektor ke jenis komponen, kami sangat menyarankan Anda memverifikasi konektor Lambda sebelum mengintegrasikan dengan AWS IoT TwinMaker. Anda dapat menguji fungsi Lambda Anda di konsol Lambda atau secara lokal di [AWS CDK](#). [Untuk informasi selengkapnya tentang menguji fungsi Lambda Anda, lihat Menguji fungsi Lambda, dan aplikasi pengujian Lokal. AWS CDK](#)

Contoh jenis komponen pabrik cookie

Dalam tipe komponen, kita mendefinisikan properti umum yang dibagi di seluruh komponen. Untuk contoh pabrik cookie, komponen fisik dari jenis yang sama berbagi pengukuran yang sama, sehingga kita dapat menentukan skema pengukuran dalam tipe komponen. Sebagai contoh, jenis mixer didefinisikan dalam contoh berikut.

```
{
  "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
    }
  }
}
```

Misalnya, komponen fisik mungkin memiliki pengukuran dalam database Timestream, catatan pemeliharaan dalam database SQL, atau data alarm dalam sistem alarm. Membuat beberapa komponen dan mengaitkannya dengan entitas menghubungkan sumber data yang berbeda ke entitas dan mengisi grafik komponen entitas. Dalam konteks ini, setiap komponen membutuhkan `telemetryId` properti untuk mengidentifikasi kunci unik komponen dalam sumber data yang sesuai. Menentukan `telemetryId` properti memiliki dua manfaat: properti dapat digunakan dalam konektor data sebagai kondisi filter untuk hanya menanyakan nilai komponen yang diberikan dan, jika Anda menyertakan nilai `telemetryId` properti dalam respons API bidang data, maka sisi klien mengambil ID dan dapat melakukan pencarian terbalik jika perlu.

Jika Anda menambahkan `TelemetryId` ke jenis komponen sebagai id eksternal, itu mengidentifikasi komponen dalam `TimeStream` tabel.

```
{
  "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
    }
}
}
}

```

Demikian pula kita memiliki tipe komponen untuk `WaterTank`, seperti yang ditunjukkan dalam contoh JSON berikut.

```

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

```

```

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

```

TelemetryType ini adalah properti opsional dalam tipe komponen jika ditujukan untuk menanyakan nilai properti dalam lingkup entitas. Sebagai contoh, lihat tipe komponen yang ditentukan dalam [GitHub repositori AWS IoT TwinMaker sampel](#). Ada jenis alarm yang juga disematkan ke dalam tabel yang sama, sehingga TelemetryType didefinisikan dan Anda mengekstrak properti umum seperti TelemetryId dan TelemetryType ke tipe komponen induk untuk tipe anak lain untuk dibagikan.

Contoh Lambda

Konektor Lambda perlu mengakses sumber data dan menghasilkan pernyataan kueri berdasarkan input dan meneruskannya ke sumber data. Contoh permintaan yang dikirim ke Lambda ditampilkan dalam contoh JSON berikut.

```

{
  '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'
}
```


Tujuan dari fungsi Lambda adalah untuk menanyakan data pengukuran historis untuk entitas tertentu. AWS IoT TwinMaker menyediakan peta komponen-properti, dan Anda harus menentukan nilai instantiated untuk ID komponen. Misalnya, untuk menangani kueri tingkat tipe komponen (yang umum untuk kasus penggunaan alarm) dan mengembalikan status alarm semua komponen di ruang kerja, maka peta properti memiliki definisi properti tipe komponen.

Untuk kasus yang paling mudah, seperti pada permintaan sebelumnya, kami menginginkan serangkaian sampel suhu selama jendela waktu yang diberikan untuk komponen yang diberikan, dalam urutan waktu menaik. Pernyataan query dapat diringkas sebagai berikut:

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

Membuat dan mengedit AWS IoT TwinMaker adegan

Adegan adalah visualisasi tiga dimensi dari kembar digital Anda. Mereka adalah cara utama bagi Anda untuk mengedit kembar digital Anda. Pelajari cara menambahkan alarm, data deret waktu, hamparan warna, tag, dan aturan visual ke adegan Anda untuk menyelaraskan visualisasi kembar digital Anda dengan kasus penggunaan dunia nyata Anda.

Bagian ini mencakup topik-topik berikut:

- [Sebelum Anda membuat adegan pertama Anda](#)
- [Unggah sumber daya ke Perpustakaan AWS IoT TwinMaker Sumber Daya](#)
- [Buat adegan Anda](#)
- [Tambahkan kamera tetap ke entitas](#)
- [Pengeditan adegan yang disempurnakan](#)
- [Edit adegan Anda](#)
- [Format model Ubin 3D](#)
- [Adegan dinamis](#)

Sebelum Anda membuat adegan pertama Anda

Adegan mengandalkan sumber daya untuk mewakili kembar digital Anda. Sumber daya ini terdiri dari model 3D, data, atau file tekstur. Ukuran dan kompleksitas sumber daya Anda, elemen dalam adegan seperti pencahayaan, dan perangkat keras komputer Anda, memengaruhi kinerja AWS IoT TwinMaker adegan. Gunakan informasi dalam topik ini untuk mengurangi jeda, waktu pemuatan, dan meningkatkan kecepatan bingkai adegan Anda.

Optimalkan sumber daya Anda sebelum mengimpornya ke AWS IoT TwinMaker

Anda dapat menggunakannya AWS IoT TwinMaker untuk berinteraksi dengan kembar digital Anda secara real time. Untuk pengalaman terbaik dengan adegan Anda, kami sarankan untuk mengoptimalkan sumber daya Anda untuk digunakan dalam lingkungan waktu nyata.

Model 3D Anda dapat memiliki dampak signifikan pada kinerja. Geometri dan jerat model yang kompleks dapat mengurangi kinerja. Misalnya, CAD model industri memiliki tingkat detail yang tinggi.

Kami merekomendasikan untuk mengompresi jerat model ini dan mengurangi jumlah poligonnya sebelum menggunakannya dalam adegan. AWS IoT TwinMaker Jika Anda membuat model 3D baru AWS IoT TwinMaker, Anda harus menetapkan tingkat detail dan mempertahankannya di semua model Anda. Hapus detail dari model yang tidak memengaruhi visualisasi atau interpretasi kasus penggunaan Anda..

Untuk mengompres model dan mengurangi ukuran file, gunakan alat kompresi mesh open source, seperti [kompresi data DRACO 3D](#).

Tekstur yang tidak dioptimalkan juga dapat memengaruhi kinerja. Jika Anda tidak memerlukan transparansi dalam tekstur Anda, pertimbangkan untuk memilih format PEG gambar di atas PNG format. Anda dapat mengompres file tekstur Anda dengan menggunakan alat kompresi tekstur open source, seperti [kompresi tekstur Basis Universal](#).

Praktik terbaik untuk kinerja di AWS IoT TwinMaker

Untuk kinerja terbaik dengan AWS IoT TwinMaker, perhatikan batasan dan praktik terbaik berikut.

- AWS IoT TwinMaker kinerja rendering adegan bergantung pada perangkat keras. Kinerja bervariasi di berbagai konfigurasi perangkat keras komputer.
- Kami merekomendasikan jumlah poligon total di bawah 1 juta di semua objek Anda di AWS IoT TwinMaker.
- Kami merekomendasikan total 200 objek per adegan. Meningkatkan jumlah objek dalam adegan di atas 200 dapat mengurangi frame rate adegan Anda.
- Kami merekomendasikan bahwa ukuran total semua aset 3D unik di adegan Anda tidak melebihi 100 megabyte. Jika tidak, Anda mungkin mengalami waktu pemuatan yang lambat atau kinerja yang menurun tergantung pada browser dan perangkat keras Anda.
- Adegan memiliki pencahayaan sekitar secara default. Anda dapat menambahkan lampu ekstra ke dalam adegan untuk membawa objek tertentu ke dalam fokus, atau melemparkan bayangan pada objek. Kami merekomendasikan menggunakan satu cahaya per adegan. Gunakan lampu jika diperlukan, dan hindari mereplikasi lampu dunia nyata dalam sebuah adegan.

Pelajari selengkapnya

Gunakan sumber daya ini untuk mempelajari lebih lanjut tentang teknik pengoptimalan yang dapat Anda gunakan untuk meningkatkan kinerja dalam adegan Anda.

- [Cara mengonversi dan mengompres OBJ model GLTF untuk digunakan AWS IoT TwinMaker](#)

- [Optimalkan model 3D Anda untuk konten web](#)
- [Mengoptimalkan adegan untuk kinerja WebGL yang lebih baik](#)

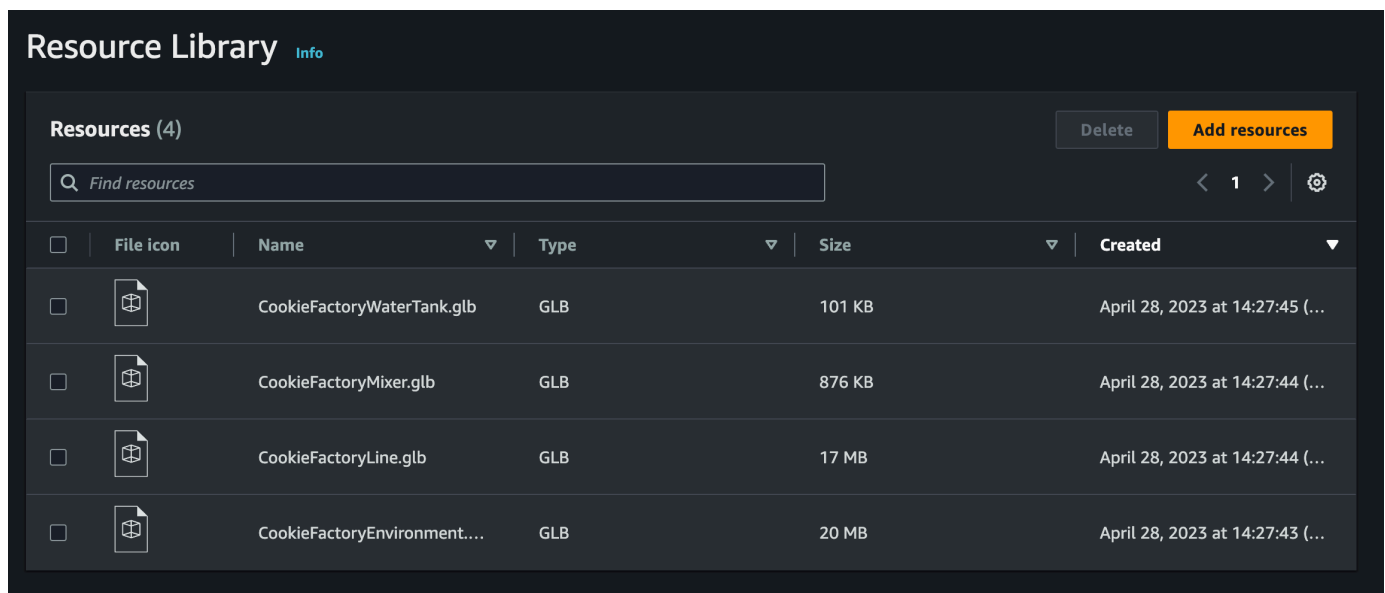
Unggah sumber daya ke Perpustakaan AWS IoT TwinMaker Sumber Daya

Anda dapat menggunakan Perpustakaan Sumber Daya untuk mengontrol dan mengelola sumber daya apa pun yang ingin Anda tempatkan ke dalam adegan untuk aplikasi kembar digital Anda. Untuk AWS IoT TwinMaker mengetahui sumber daya, unggah menggunakan halaman konsol Perpustakaan Sumber Daya.

Unggah file ke Perpustakaan Sumber Daya menggunakan konsol

Ikuti langkah-langkah ini untuk menambahkan file ke Perpustakaan Sumber Daya menggunakan AWS IoT TwinMaker konsol.

1. Di menu navigasi kiri, di bawah Workspaces, pilih Perpustakaan Sumber Daya.
2. Pilih Tambahkan sumber daya dan pilih file yang ingin Anda unggah.



Buat adegan Anda

Di bagian ini, Anda akan mengatur adegan sehingga Anda dapat mengedit kembar digital Anda. Anda dapat mengimpor model 3D yang diunggah ke [perpustakaan sumber daya](#), lalu menambahkan

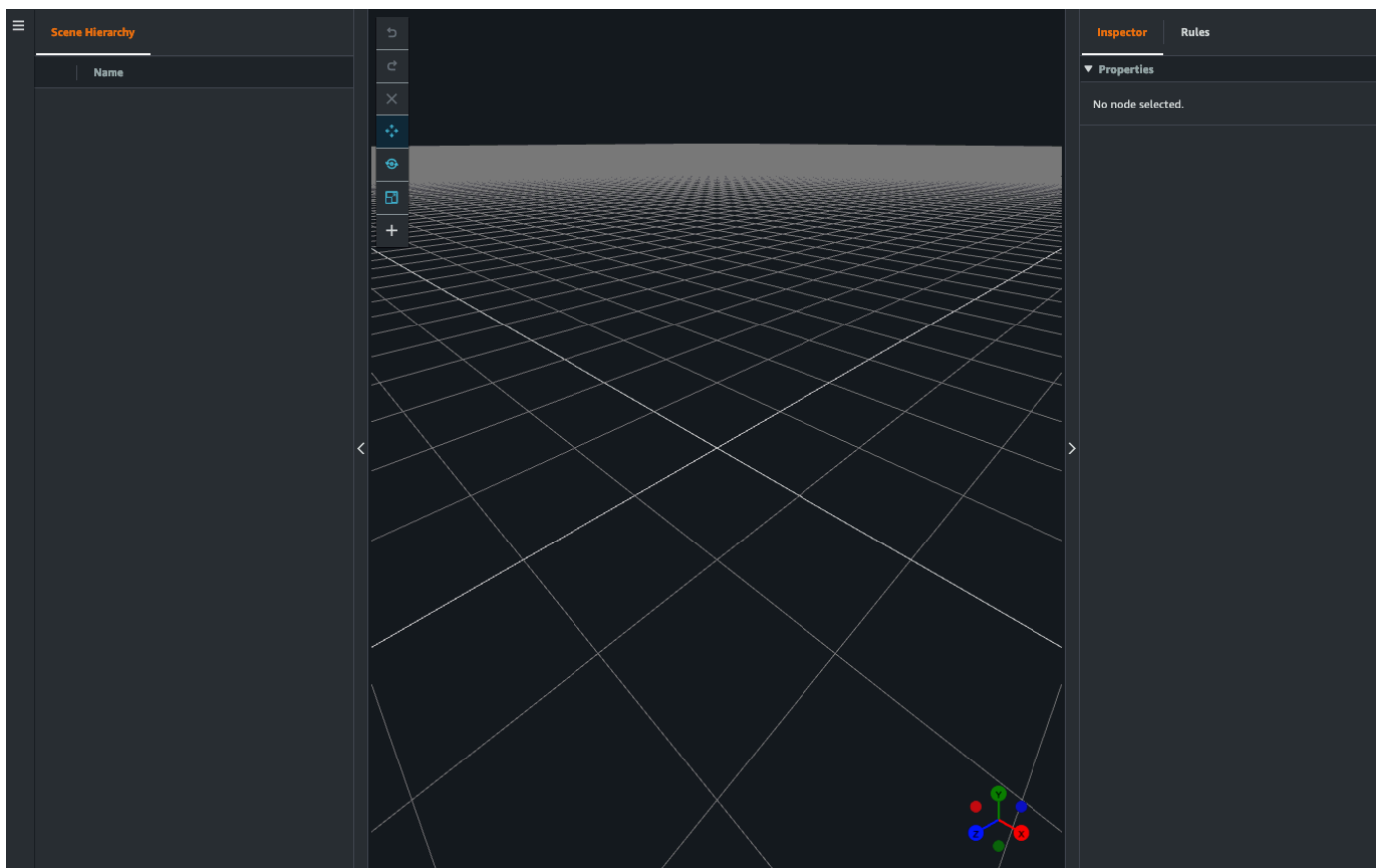
widget dan mengikat data properti ke objek untuk menyelesaikan kembar digital Anda. Objek pemandangan dapat mencakup seluruh bangunan atau ruang, atau masing-masing peralatan yang diposisikan di lokasi fisiknya.

Note

Sebelum Anda membuat adegan, Anda harus membuat ruang kerja.

Gunakan prosedur berikut untuk membuat adegan Anda AWS IoT TwinMaker.

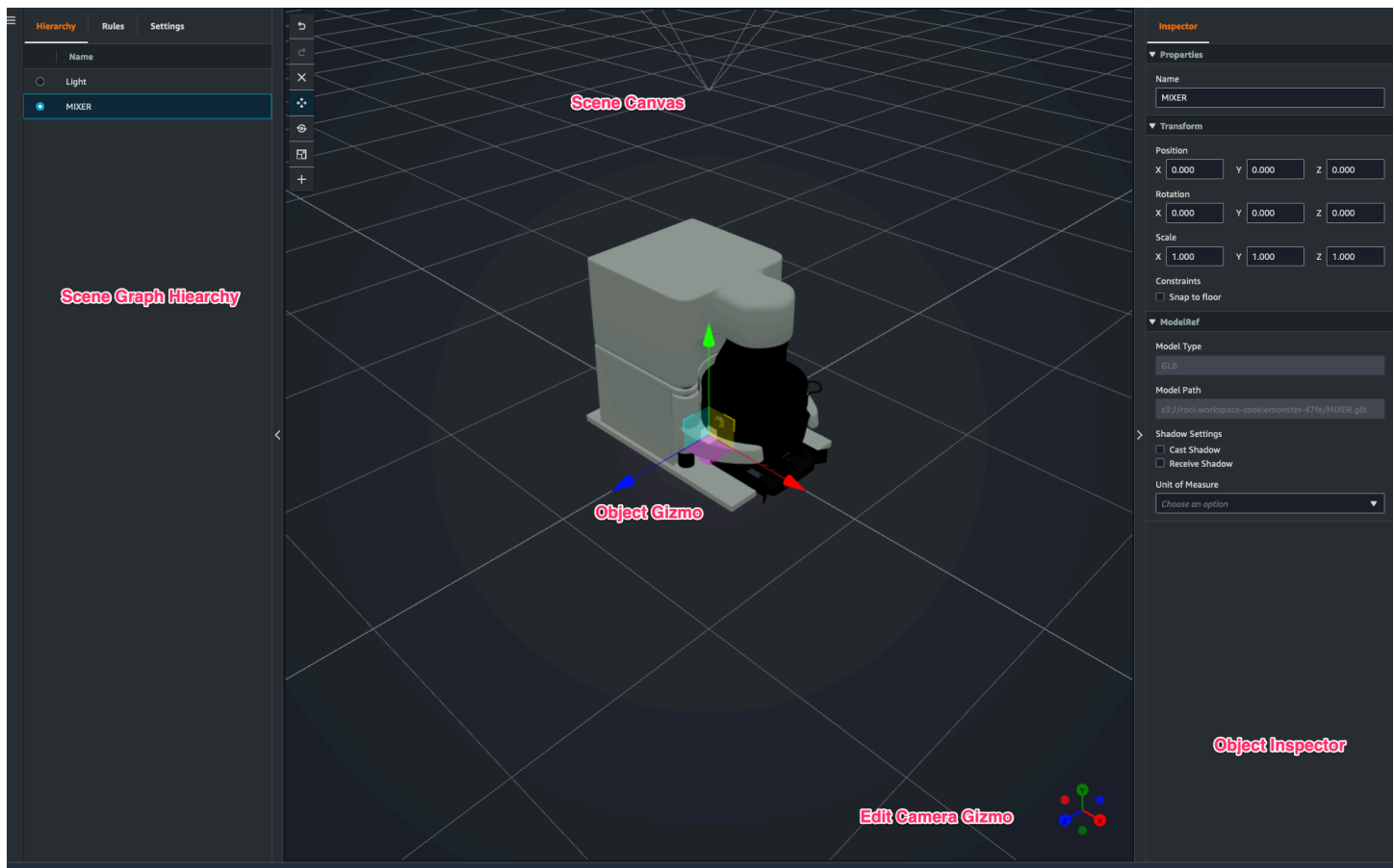
1. Untuk membuka panel adegan, di navigasi kiri ruang kerja Anda, pilih Scenes.
2. Pilih Buat adegan. Panel pembuatan adegan baru terbuka.
3. Di panel pembuatan adegan, masukkan nama dan deskripsi untuk adegan baru Anda. Jika Anda memiliki paket harga bundel standar atau berjenjang, Anda dapat memilih jenis adegan Anda. Disarankan untuk menggunakan [adegan dinamis](#).
4. Saat Anda siap untuk membuat adegan, pilih Buat adegan. Adegan baru terbuka dan siap bagi Anda untuk bekerja dengannya.



Gunakan navigasi 3D AWS IoT TwinMaker dalam adegan Anda

AWS IoT TwinMaker Adegan ini memiliki seperangkat kontrol navigasi yang dapat Anda gunakan untuk menavigasi secara efisien melalui ruang 3D pemandangan Anda. Untuk berinteraksi dengan ruang 3D dan objek yang diwakili oleh adegan Anda, Anda menggunakan widget dan opsi menu berikut.

- **Inspector:** Gunakan jendela Inspector untuk melihat dan mengedit properti dan pengaturan entitas atau komponen yang dipilih dalam hierarki Anda.
- **Scene Canvas:** The Scene Canvas adalah ruang 3D tempat Anda dapat memposisikan dan mengarahkan sumber daya 3D apa pun yang ingin Anda gunakan.
- **Hierarki Grafik Adegan:** Anda dapat menggunakan panel ini untuk melihat semua entitas yang ada di adegan Anda. Itu muncul di sisi kiri jendela.
- **Object gizmo:** Gunakan alat ini untuk memindahkan objek di sekitar kanvas. Itu muncul di tengah objek 3D yang dipilih di Scene Canvas.
- **Edit Alat Kamera:** Gunakan alat Edit Kamera untuk dengan cepat melihat orientasi kamera tampilan pemandangan saat ini dan memodifikasi sudut pandang. Anda dapat menemukan alat ini di sudut kanan bawah tampilan pemandangan.
- **Kontrol zoom:** Untuk menavigasi di Scene Canvas, gunakan klik kanan dan seret ke arah yang ingin Anda pindahkan. Untuk memutar, klik kiri dan seret untuk memutar. Untuk memperbesar, gunakan roda gulir pada mouse Anda, atau cubit dan gerakkan jari-jari Anda terpisah di track pad laptop Anda.



Tombol adegan pada panel hierarki memiliki fungsi berikut yang terdaftar, dalam urutan tata letak tombol:

- Undo: Urungkan perubahan terakhir Anda dalam adegan.
- Redo: Ulangi perubahan terakhir Anda dalam adegan.
- Plus (+): Gunakan tombol ini untuk mendapatkan akses ke tindakan berikut: Tambahkan simpul kosong, Tambahkan model 3D, Tambahkan tag, Tambahkan cahaya, dan Tambahkan shader model.
- Ubah metode navigasi: Dapatkan akses ke opsi navigasi kamera adegan, Orbit dan Pan.
- Tempat sampah (hapus): Gunakan tombol ini untuk menghapus objek yang dipilih di adegan Anda.
- Alat manipulasi objek: Gunakan tombol ini untuk menerjemahkan, memutar, dan menskalakan objek yang dipilih.

Tambahkan kamera tetap ke entitas

Anda dapat melampirkan tampilan kamera tetap ke entitas Anda dalam AWS IoT TwinMaker adegan Anda. Kamera ini memberikan perspektif tetap pada model 3d, memungkinkan Anda untuk dengan cepat dan mudah mengubah perspektif Anda dalam sebuah adegan ke entitas yang ditargetkan.

1. Arahkan ke adegan Anda di [AWS IoT TwinMaker konsol](#).
2. Di menu hierarki adegan, pilih entitas yang ingin Anda lampirkan kamera.
3. Tekan tombol +, dan dari opsi tarik-turun pilih Tambahkan kamera dari tampilan saat ini. Untuk menerapkan kamera dengan perspektif saat ini ke entitas.
4. Di inspektur, Anda dapat mengonfigurasi kamera Anda dan menyesuaikan pengaturan berikut:
 - Nama kamera
 - Posisi dan rotasi kamera
 - Panjang fokus kamera
 - Tingkat zoom
 - Pesawat kliping dekat dan jauh
5. Untuk mengakses kamera Anda setelah Anda meletakkannya. Pilih entitas yang Anda tambahkan kamera dalam hierarki. Cari nama kamera yang tercantum di bawah entitas.
6. Setelah Anda memilih kamera yang ditempatkan dari entitas Anda, tampilan kamera adegan akan snap ke perspektif yang ditetapkan dari kamera yang ditempatkan.

Pengeditan adegan yang disempurnakan

AWS IoT TwinMaker adegan menampilkan seperangkat alat untuk meningkatkan dan mengedit dan memanipulasi sumber daya yang ada di adegan Anda.

Topik berikut mengajarkan Anda cara menggunakan fitur pengeditan yang disempurnakan dalam AWS IoT TwinMaker adegan Anda.

- [Penempatan objek adegan yang ditargetkan](#)
- [Pemilihan submodel](#)
- [Edit entitas dalam hierarki adegan](#)

Penempatan objek adegan yang ditargetkan

AWS IoT TwinMaker memungkinkan Anda untuk secara tepat menempatkan dan menambahkan objek ke dalam adegan Anda. Fitur pengeditan yang disempurnakan ini memberi Anda kontrol yang lebih besar tentang tempat Anda menempatkan tag, lampu entitas, dan model di adegan Anda.

1. Arahkan ke adegan Anda di [AWS IoT TwinMaker konsol](#).
2. Tekan tombol +, dan dari opsi drop-down pilih salah satu opsi. Ini bisa berupa model, lampu, tag, atau apa pun dari menu +.

Ketika Anda memindahkan kursor Anda di ruang 3d adegan Anda, Anda akan melihat target di sekitar kursor Anda.

3. Gunakan target untuk secara tepat menempatkan elemen dalam adegan Anda.

Pemilihan submodel

AWS IoT TwinMaker memungkinkan Anda memilih submodel model 3d dalam adegan dan menerapkan properti standar untuk mereka, seperti tag, lampu, atau aturan.

Format file model 3d berisi metadata yang dapat menentukan sub area model sebagai submodel dalam model yang lebih besar. Misalnya model dapat berupa sistem filtrasi, bagian-bagian individu dari sistem seperti tangki, pipa, atau motor ditandai sebagai submodel dari model 3d filtrasi.

Format file 3D yang didukung dalam adegan:GLB, danGLTF.

1. Arahkan ke adegan Anda di [AWS IoT TwinMaker konsol](#).
2. Jika Anda tidak memiliki model dalam adegan Anda, pastikan untuk menambahkannya dengan memilih opsi dari menu +.
3. Pilih model yang tercantum dalam hierarki adegan Anda, setelah dipilih hierarki harus menampilkan submodel apa pun di bawah model.

Note

Jika Anda tidak melihat submodel apa pun yang terdaftar maka kemungkinan model tersebut tidak dikonfigurasi untuk memiliki submodel apa pun.

4. Untuk mengaktifkan visibilitas submodel, tekan ikon mata, yang terletak di sebelah kanan nama submodel dalam hierarki.

5. Untuk mengedit data submodel, seperti nama atau posisinya, inspektur adegan akan terbuka ketika submodel dipilih. Gunakan menu inspektur untuk memperbarui atau mengubah data submodel.
6. Untuk menambahkan tag, lampu, aturan, atau properti lain ke submodel, tekan +, sementara submodel dipilih dalam hierarki.

Edit entitas dalam hierarki adegan

AWS IoT TwinMaker adegan memungkinkan Anda langsung mengedit properti entitas dalam tabel hierarki. Prosedur berikut menunjukkan kepada Anda tindakan mana yang dapat Anda lakukan pada entitas melalui menu hierarki.

1. Arahkan ke adegan Anda di [AWS IoT TwinMaker konsol](#).
2. Buka hierarki adegan, dan pilih sub elemen dari entitas yang ingin Anda manipulasi.
3. Setelah elemen dipilih, tekan tombol +, dan dari drop-down pilih salah satu opsi:
 - Tambahkan simpul kosong
 - Tambahkan model 3D
 - Tambahkan cahaya
 - Tambahkan kamera dari tampilan saat ini
 - Tambahkan tag
 - Tambahkan model shader
 - Tambahkan indikator gerak
4. Setelah memilih salah satu opsi dari drop-down, pemilihan akan diterapkan ke adegan sebagai anak dari elemen yang dipilih dari langkah 2.
5. Anda dapat menyusun ulang elemen anak dan elemen reparent, dengan memilih elemen anak dan menyeret hierarki ke induk baru.

Tambahkan anotasi ke entitas

Komposer AWS IoT TwinMaker adegan memungkinkan Anda membuat anotasi elemen apa pun dalam hierarki adegan Anda. Anotasi ditulis dalam penurunan harga.

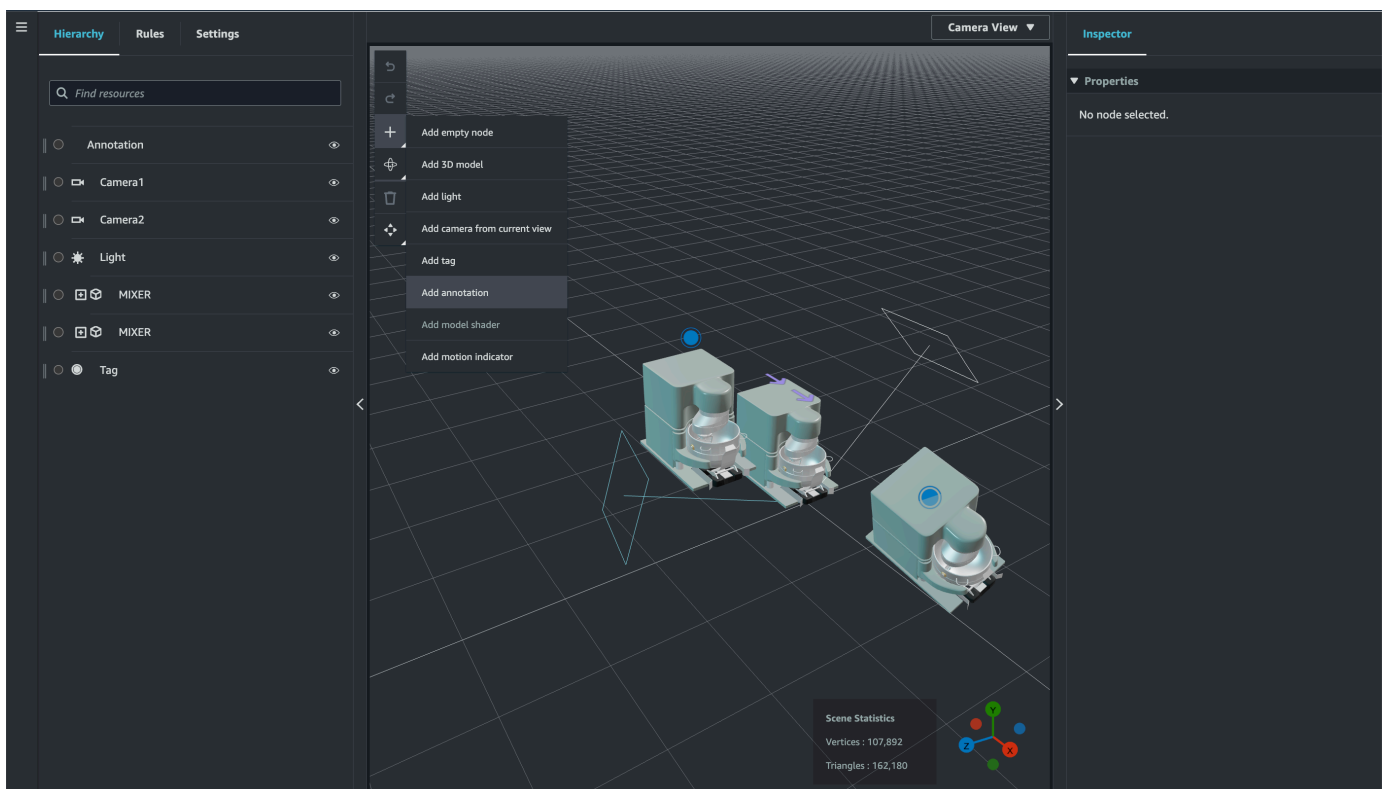
[Untuk informasi lebih lanjut tentang menulis di Markdown, lihat dokumentasi resmi tentang sintaks penurunan harga, Sintaks Dasar.](#)

Note

AWS IoT TwinMaker anotasi dan overlay sintaks Markdown saja dan tidak. HTML

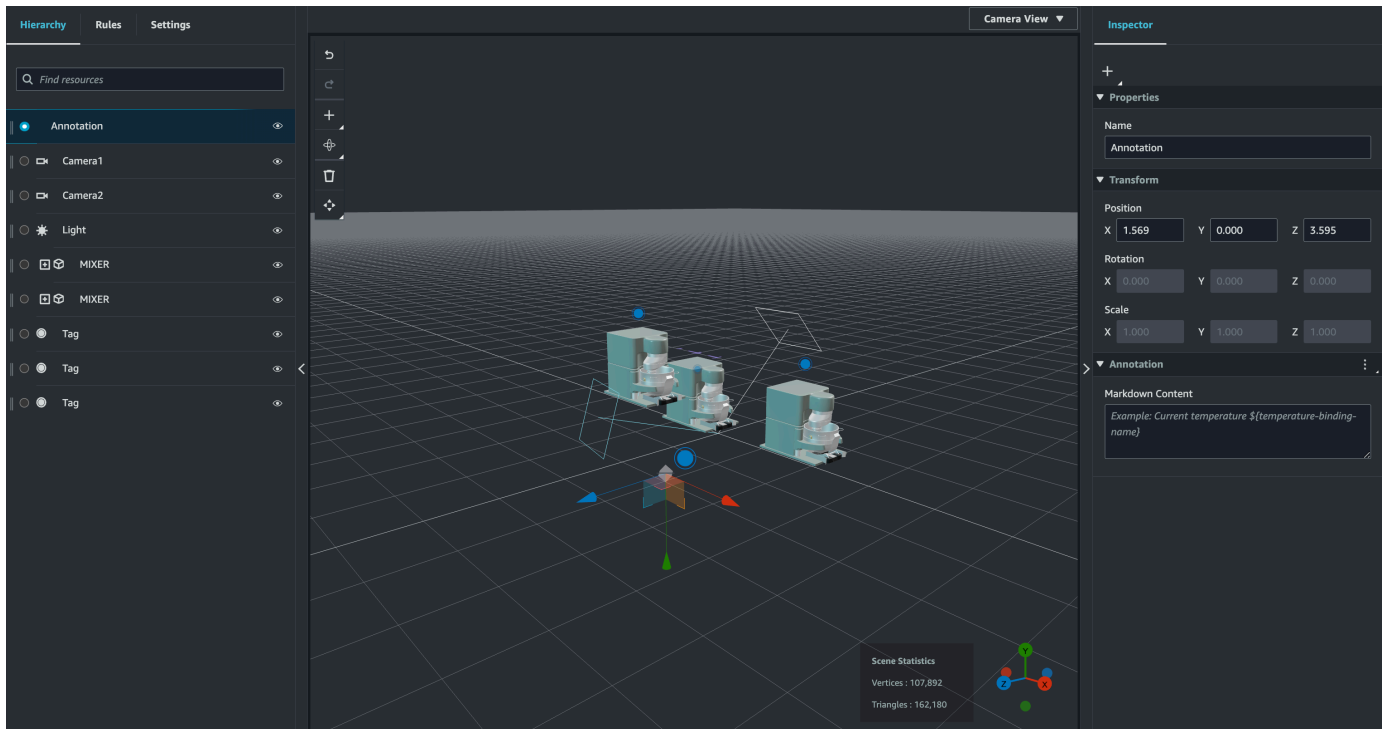
Menambahkan anotasi ke entitas

1. Arahkan ke adegan Anda di [AWS IoT TwinMaker konsol](#).
2. Pilih elemen dari hierarki adegan yang ingin Anda anotasi. Jika tidak ada elemen dalam hierarki yang dipilih, maka Anda dapat menambahkan anotasi ke root.
3. Tekan tombol plus + dan pilih opsi Tambahkan anotasi.



4. Di jendela Inspector di sebelah kiri, gulir ke bawah ke bagian anotasi. Menggunakan sintaks Markdown, tulis teks yang ingin ditampilkan anotasi Anda.

[Untuk informasi lebih lanjut tentang menulis di Markdown, lihat dokumentasi resmi tentang sintaks penurunan harga, Sintaks Dasar.](#)



5. Untuk mengikat data AWS IoT TwinMaker adegan Anda ke anotasi pilih Tambahkan pengikatan data, tambahkan Id Entitas, lalu pilih Nama Komponen dan Nama Properti entitas yang ingin Anda tampilkan datanya. Anda dapat memperbarui nama pengikatan untuk menggunakannya sebagai variabel Markdown, dan memunculkan data dalam anotasi.

Inspector



▼ Properties

Name

Annotation

▼ Transform

Position

X 1.569

Y 0.000

Z 3.595

Rotation

X 0.000

Y 0.000

Z 0.000

Scale

X 1.000

Y 1.000

Z 1.000

▼ Annotation



Add data binding

Markdown Content

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

Inspector



▼ Properties

Name

Annotation

▼ Transform

Position

X 1.569

Y 0.000

Z 3.595

Rotation

X 0.000

Y 0.000

Z 0.000

Scale

X 1.000

Y 1.000

Z 1.000

▼ Annotation ⋮

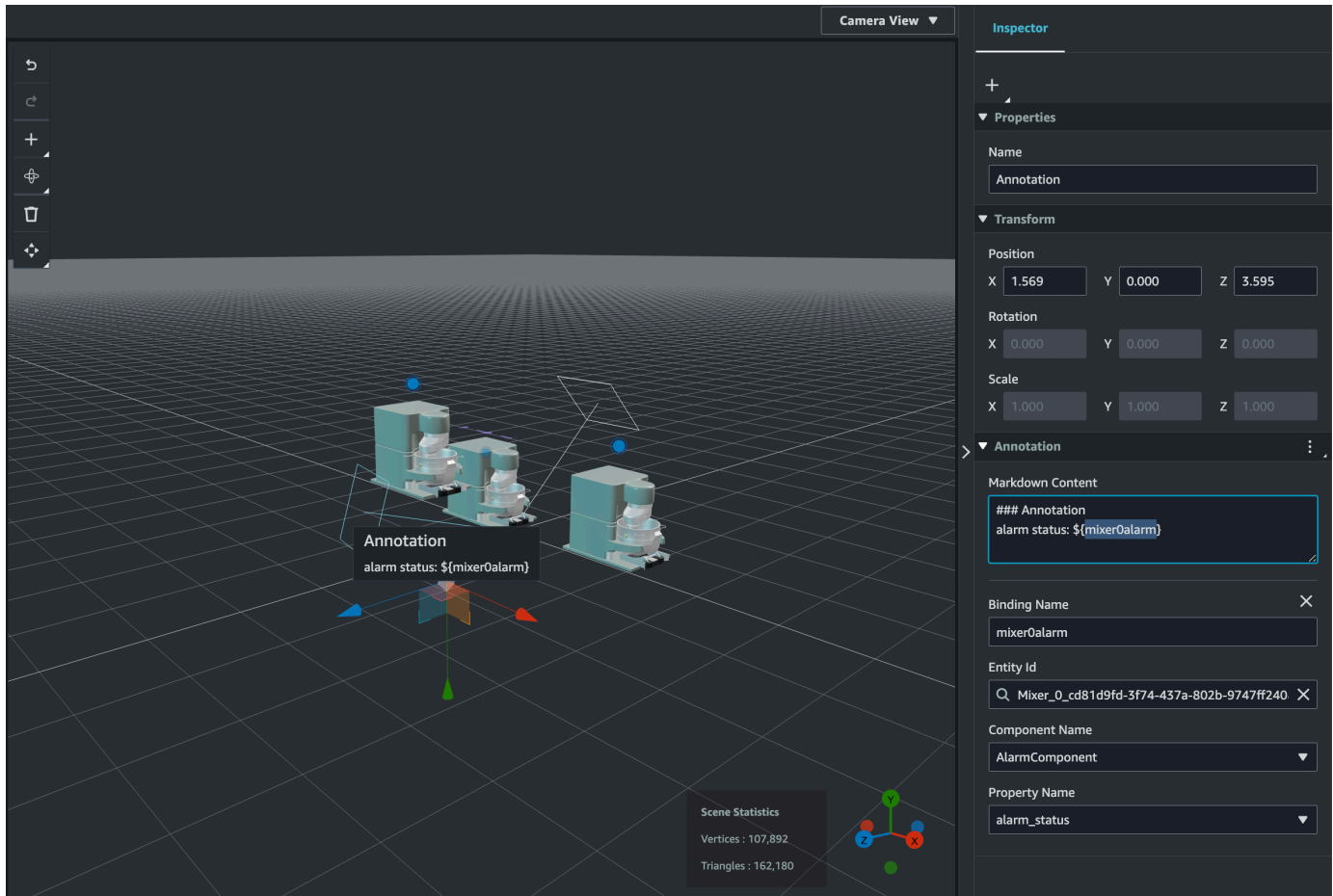
Markdown Content

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

6. Nama Binding digunakan untuk mewakili variabel anotasi.

Masukkan Nama Binding untuk memunculkan nilai historis terbaru dari deret waktu entitas dalam anotasi melalui AWS IoT TwinMaker sintaks variabel: `${variable-name}`

Sebagai contoh, overlay ini menampilkan nilai `mixer0alarm`, dalam anotasi dengan sintaks `${mixer0alarm}`



Tambahkan overlay ke Tag

Anda dapat membuat overlay untuk adegan Anda AWS IoT TwinMaker. Hamparan adegan dikaitkan dengan tag dan dapat digunakan untuk memunculkan data penting yang terkait dengan entitas adegan Anda. Hamparan ditulis dan dirender dalam Markdown.

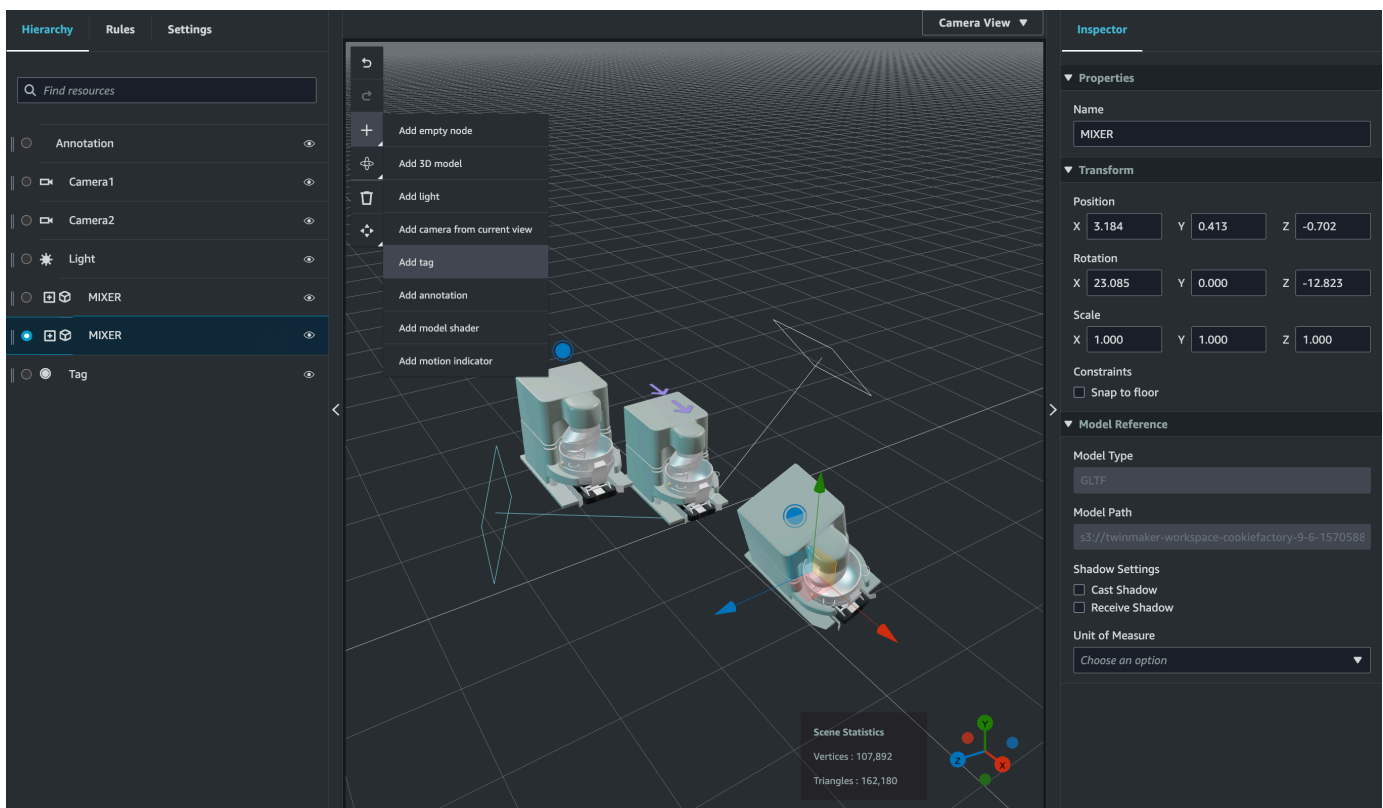
[Untuk informasi lebih lanjut tentang menulis di Markdown, lihat dokumentasi resmi tentang sintaks penurunan harga, Sintaks Dasar.](#)

Note

Secara default, Overlay terlihat dalam adegan hanya ketika tag yang terkait dengannya dipilih. Anda dapat mengaktifkan ini di Pengaturan adegan sehingga semua Hamparan terlihat sekaligus.

1. Arahkan ke adegan Anda di [AWS IoT TwinMaker konsol](#).
2. AWS IoT TwinMaker Hamparan dikaitkan dengan adegan tag, Anda dapat memperbarui tag yang ada atau menambahkan yang baru.

Tekan tombol plus + dan pilih opsi Tambahkan tag.



3. Pada panel Inspector di sebelah kanan, pilih tombol + (plus symbol) lalu pilih Add overlay.

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



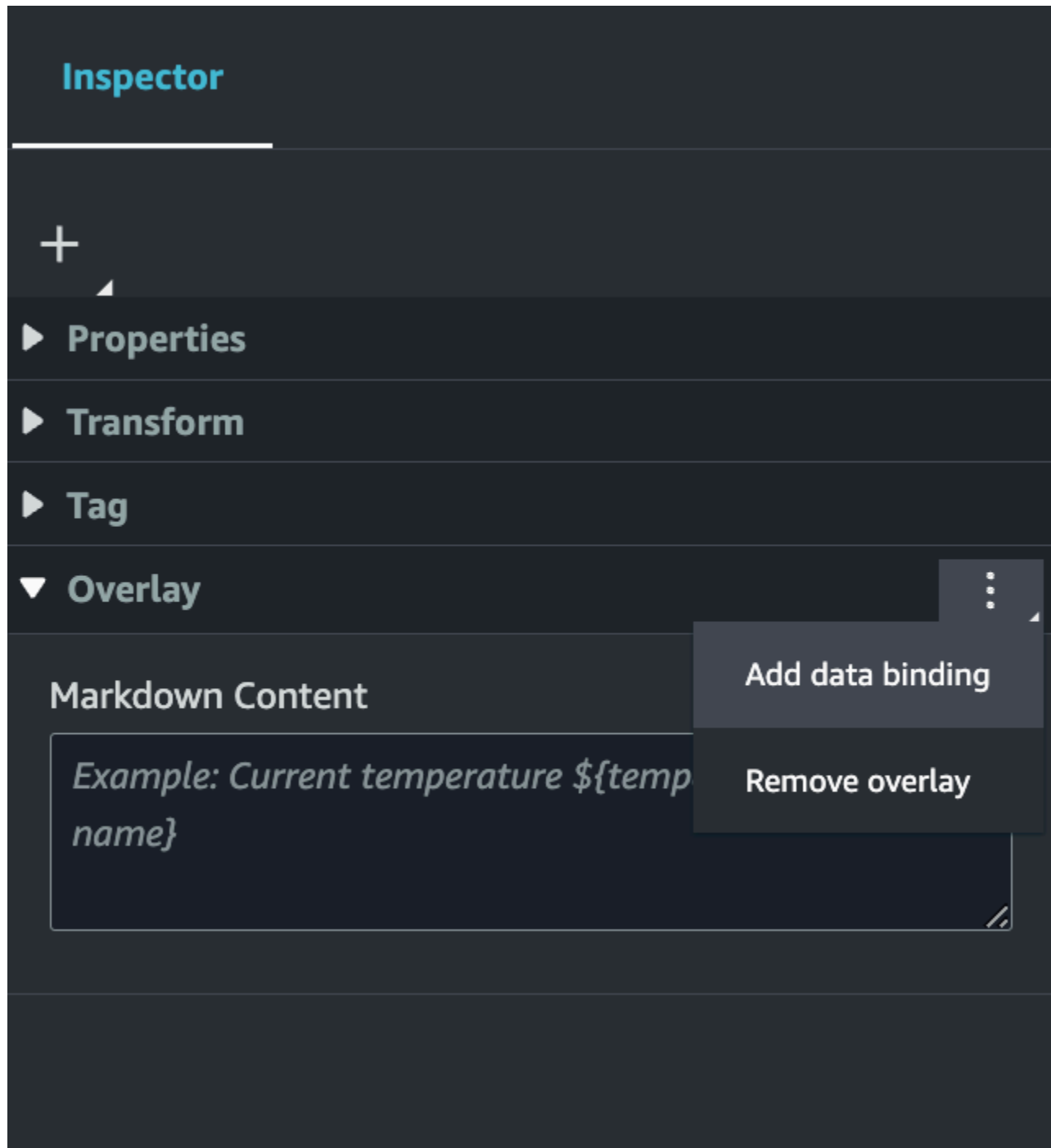
Link Target

Tambahkan overlay ke Tag

4. Dalam sintaks Markdown, tulis teks yang Anda inginkan untuk menampilkan overlay Anda.

[Untuk informasi lebih lanjut tentang menulis di Markdown, lihat dokumentasi resmi tentang sintaks penurunan harga, Sintaks Dasar.](#)

5. Untuk mengikat data AWS IoT TwinMaker adegan Anda ke overlay, pilih Tambahkan pengikatan data.



Tambahkan nama Binding dan Id Entitas, lalu pilih Nama Komponen dan Nama Properti entitas yang ingin Anda tampilkan datanya.

6. Anda dapat memunculkan nilai historis terbaru dari data deret waktu entitas dalam overlay melalui AWS IoT TwinMaker sintaks variabel: `${variable-name}`

Sebagai contoh, overlay ini menampilkan nilai `mixer0alarm`, dalam overlay dengan sintaks `${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. Untuk mengaktifkan visibilitas Overlay, buka tab Pengaturan di kiri atas, dan pastikan sakelar untuk Overlay diaktifkan sehingga semua Overlay terlihat sekaligus.

 Note

Secara default, Overlay terlihat dalam adegan hanya ketika tag yang terkait dengannya dipilih.

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

Tambahkan overlay ke Tag

Edit adegan Anda

Setelah membuat adegan, Anda dapat menambahkan entitas, komponen, dan mengonfigurasi widget tambahan ke dalam adegan Anda. Gunakan komponen entitas dan widget untuk memodelkan kembar digital Anda dan menyediakan fungsionalitas yang sesuai dengan kasus penggunaan Anda.

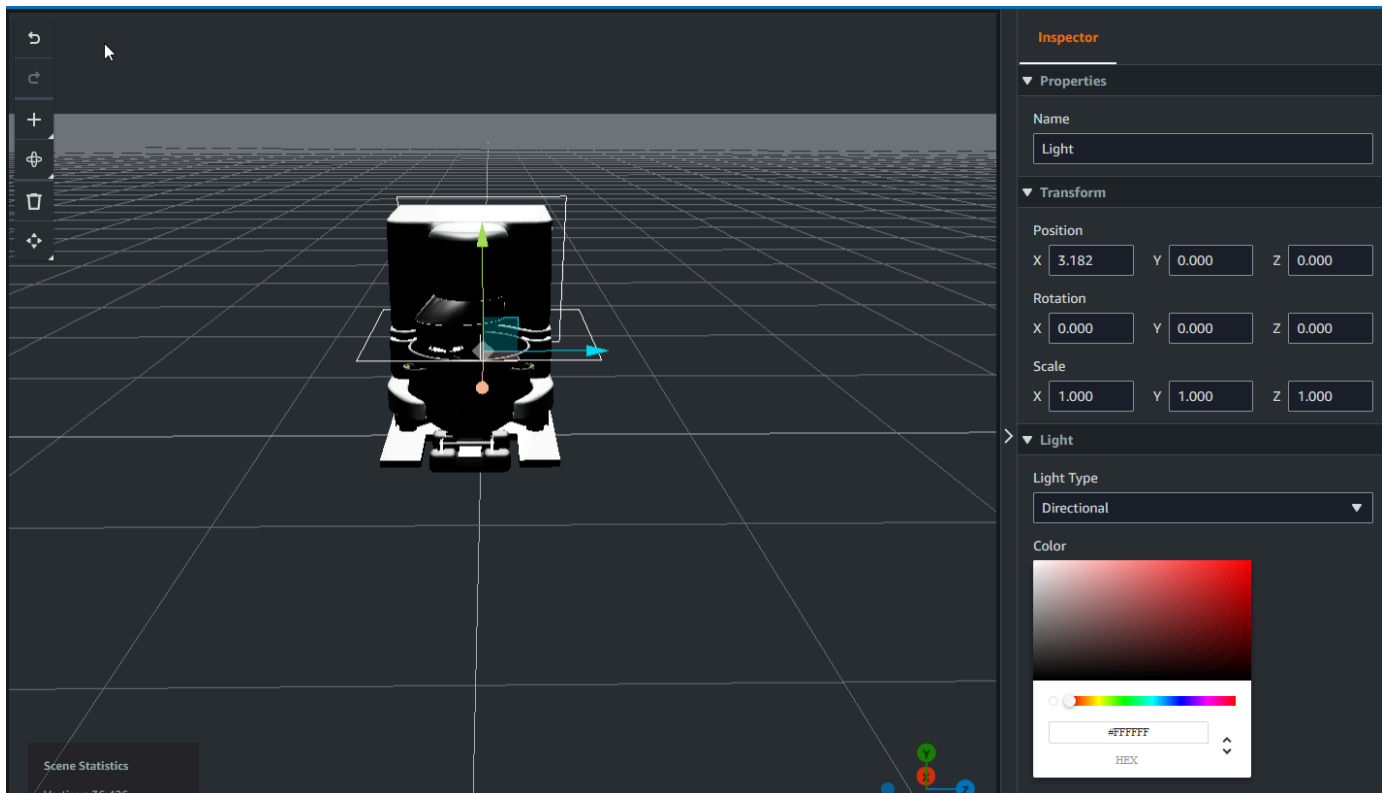
Tambahkan model ke adegan Anda

Untuk menambahkan model ke adegan Anda, gunakan prosedur berikut.

Note

Untuk menambahkan model dalam adegan Anda, Anda harus terlebih dahulu mengunggah model ke Perpustakaan AWS IoT TwinMaker Sumber Daya. Untuk informasi selengkapnya, lihat [Unggah sumber daya ke Perpustakaan AWS IoT TwinMaker Sumber Daya](#).

1. Pada halaman komposer adegan, pilih tanda plus (+), lalu pilih Tambahkan model 3D.
2. Pada jendela Tambahkan sumber daya dari pustaka sumber daya, pilih CookieFactorMixerfile.glb, lalu pilih Tambah. Komposer adegan terbuka.
3. Opsional: Pilih tanda plus (+), lalu pilih Tambahkan cahaya.
4. Pilih setiap opsi cahaya untuk melihat bagaimana pengaruhnya terhadap pemandangan.



Note

Adegan memiliki pencahayaan ambient default. Untuk menghindari kehilangan frame rate, pertimbangkan untuk membatasi jumlah lampu tambahan yang ditempatkan di adegan Anda.

Tambahkan widget UI augmented model shader ke adegan Anda

Widget shader model dapat mengubah warna objek dalam kondisi yang Anda tentukan. Misalnya, Anda dapat membuat widget warna yang mengubah warna mixer cookie di adegan Anda berdasarkan data suhu mixer.

Gunakan prosedur berikut untuk menambahkan widget shader model ke objek yang dipilih.

1. Pilih objek dalam hierarki yang ingin Anda tambahkan widget. Tekan tombol + dan kemudian pilih Model Shader.
2. Untuk menambahkan grup aturan visual baru, pertama ikuti petunjuk di bawah ini untuk membuat ColorRule, lalu di panel Inspector untuk objek ID Aturan, pilih. ColorRule

3. Pilih EntityId, ComponentName, dan PropertyName Anda ingin mengikat shader model ke.

Buat aturan visual untuk adegan Anda


Anda dapat menggunakan peta aturan visual untuk menentukan kondisi berbasis data yang mengubah tampilan visual widget UI tambahan, seperti tag atau shader model. Ada contoh aturan yang disediakan, tetapi Anda juga dapat membuatnya sendiri. Contoh berikut menunjukkan aturan visual.

The screenshot displays the AWS IoT TwinMaker console interface for managing rules. It features a dark-themed sidebar on the left with a hamburger menu icon. The main content area shows a list of three rules, each with an 'Expression' field and a 'Target' section. The first rule has the expression 'temperature >= 40' and a target of 'Error' with a red 'X' icon. The second rule has the expression 'temperature >= 20' and a target of 'Warning' with a yellow exclamation mark icon. The third rule has the expression 'temperature < 20' and a target of 'Info' with a blue circle icon. Below the list are buttons for 'Add new statement', 'Remove Rule', and 'sampleTimeSeriesColorRule'. At the bottom, there is a 'Rule Id' field.

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

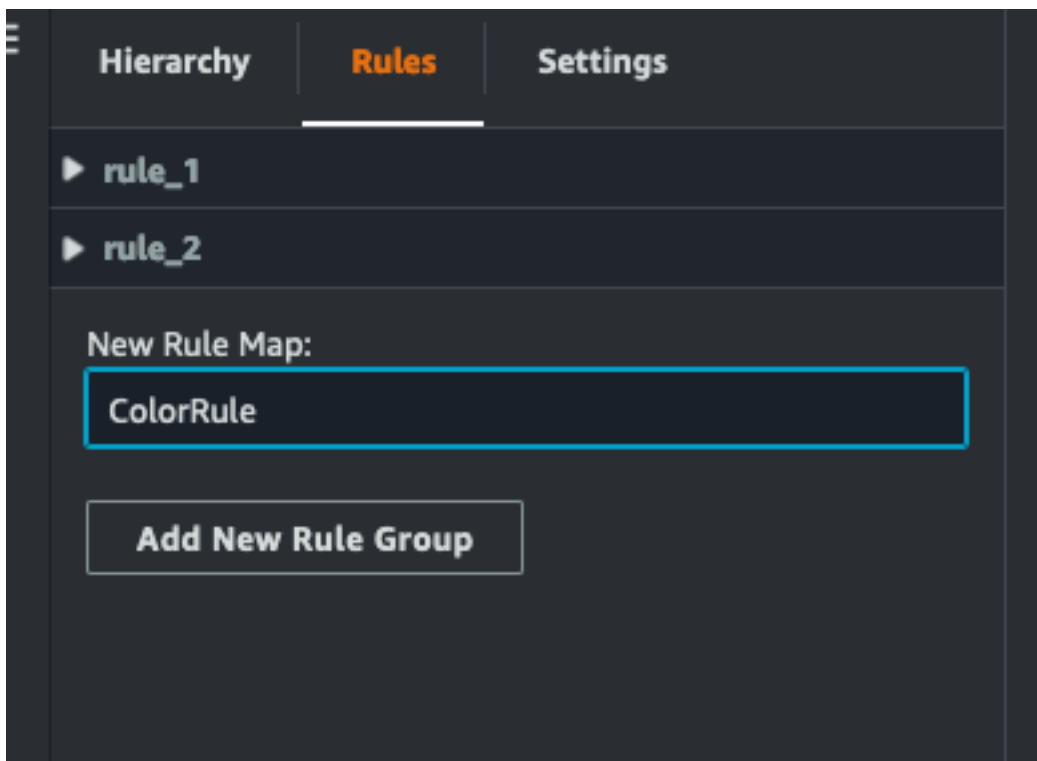
▶ sampleTimeSeriesColorRule

Rule Id

Gambar di atas menunjukkan aturan ketika properti data yang ditentukan sebelumnya dengan 'suhu' ID diperiksa terhadap nilai tertentu. Misalnya, jika 'suhu' lebih besar dari atau sama dengan 40, keadaan akan mengubah tampilan tag menjadi lingkaran merah. Target, ketika dipilih di dasbor Grafana, mengisi panel detail yang dikonfigurasi untuk menggunakan sumber data yang sama.

Prosedur berikut menunjukkan cara menambahkan grup aturan visual baru untuk lapisan UI ditambah pewarnaan mesh.

1. Di bawah tab aturan di konsol, masukkan nama seperti ColorRule di bidang teks dan pilih Tambahkan Grup Aturan Baru.



2. Tentukan aturan baru untuk kasus penggunaan Anda. Misalnya, Anda dapat membuat satu berdasarkan properti data 'suhu', di mana nilai yang dilaporkan kurang dari 20. Gunakan sintaks berikut untuk ekspresi aturan: Kurang dari <, lebih besar dari adalah >, kurang dari atau sama adalah <=, lebih besar dari atau sama adalah >=, dan sama adalah ==. (Untuk informasi selengkapnya, lihat [JEXLsintaks Apache Commons](#).)
3. Tetapkan target ke warna. Untuk menentukan warna, seperti #fcba03, gunakan nilai hex. (Untuk informasi selengkapnya tentang nilai hex, lihat [Heksadesimal](#).)

Membuat tag untuk adegan Anda

Tag adalah anotasi yang ditambahkan ke posisi x , y , z koordinat tertentu dari sebuah adegan. Tag menggunakan properti entitas untuk menghubungkan bagian adegan ke grafik pengetahuan. Anda dapat menggunakan tag untuk mengonfigurasi perilaku atau tampilan visual item dalam adegan, seperti alarm.

Note

Untuk menambahkan fungsionalitas ke tag, Anda menerapkan aturan visual padanya.

Gunakan prosedur berikut untuk menambahkan tag ke adegan Anda.

1. Pilih objek dalam hierarki, pilih tombol +, lalu pilih Tambah Tag.
2. Beri nama tag. Kemudian, untuk menerapkan aturan visual, pilih ID grup visual.
3. Dalam daftar dropdown, pilih EntityId,, dan. ComponentName PropertyName
4. Untuk mengisi bidang Jalur Data, pilih Buat DataFrameLabel.

Format model Ubin 3D

Menggunakan Ubin 3D di adegan Anda

Jika Anda mengalami waktu tunggu yang lama saat memuat adegan 3D AWS IoT TwinMaker atau memiliki kinerja rendering yang buruk saat menavigasi model 3D yang kompleks, maka Anda mungkin ingin mengonversi model Anda ke ubin 3D. Bagian ini menjelaskan format ubin 3D dan alat pihak ketiga yang tersedia. Baca terus untuk memutuskan apakah 3D Tiles tepat untuk kasus penggunaan Anda dan untuk bantuan memulai.

Kasus penggunaan model yang kompleks

Model 3D dalam AWS IoT TwinMaker adegan Anda dapat menyebabkan masalah kinerja seperti waktu pemuatan yang lambat dan navigasi yang tertinggal jika modelnya adalah:

- Besar: ukuran filenya lebih besar dari 100MB.
- Padat: terdiri dari ratusan atau ribuan jerat yang berbeda.

- Kompleks: geometri mesh memiliki jutaan segitiga untuk membentuk bentuk yang kompleks.

Format Ubin 3D

[Format 3D Tiles](#) adalah solusi untuk streaming geometri model dan meningkatkan kinerja rendering 3D. Ini memungkinkan pemuatan instan model 3D dalam sebuah AWS IoT TwinMaker adegan, dan mengoptimalkan interaksi 3D dengan memuat potongan model berdasarkan apa yang terlihat dalam tampilan kamera.

Format 3D Tiles dibuat oleh [Cesium](#). Cesium memiliki layanan terkelola untuk mengubah model 3D menjadi Ubin 3D yang disebut [Cesium Ion](#). Saat ini solusi terbaik untuk membuat Ubin 3D, dan kami merekomendasikan ini untuk model kompleks Anda dalam [format yang didukung](#). Anda dapat mendaftarkan Cesium dan memilih paket berlangganan yang sesuai berdasarkan kebutuhan bisnis Anda di halaman [harga Cesium](#).

Untuk menyiapkan model Ubin 3D yang dapat Anda tambahkan ke AWS IoT TwinMaker adegan, ikuti instruksi yang didokumentasikan oleh Cesium Ion:

- [Impor model ke Cesium Ion](#)

Unggah ubin 3D Cesium ke AWS

Setelah model Anda dikonversi ke 3D Tiles, unduh file model lalu unggah ke bucket Amazon S3 AWS IoT TwinMaker ruang kerja Anda:

1. [Buat dan unduh arsip model Ubin 3D Anda](#).
2. Buka zip arsip ke dalam folder.
3. Unggah seluruh folder 3D Tiles ke bucket Amazon S3 yang terkait dengan ruang kerja Anda AWS IoT TwinMaker . (Lihat [Mengunggah objek](#) di Panduan Pengguna Amazon S3.)
4. Jika model Ubin 3D Anda berhasil diunggah, Anda akan melihat jalur folder Amazon S3 di Perpustakaan [Sumber Daya AWS IoT TwinMaker Anda](#) dengan tipe. Tiles3D

Note

Perpustakaan AWS IoT TwinMaker Sumber Daya tidak mendukung pengunggahan model 3D Tiles secara langsung.

Menggunakan Ubin 3D di AWS IoT TwinMaker

AWS IoT TwinMaker mengetahui model 3D Tiles yang diunggah ke bucket S3 ruang kerja Anda. Model harus memiliki `tileset.json` dan semua file dependen (`.gltf`, `.b3dm`, `.i3dm`, `.cmpt`, `.pnnts`) yang tersedia di direktori Amazon S3 yang sama. Jalur direktori Amazon S3 akan muncul di Perpustakaan Sumber Daya dengan jenisnya `Tiles3D`.

Untuk menambahkan model 3D Tiles ke adegan Anda, ikuti langkah-langkah berikut:

1. Pada halaman komposer adegan, pilih tanda plus (+), lalu pilih Tambahkan model 3D.
2. Pada jendela Tambahkan sumber daya dari pustaka sumber daya, pilih jalur ke model Ubin 3D Anda dengan jenisnya `Tiles3D`, lalu pilih Tambah.
3. Klik pada kanvas untuk menempatkan model di adegan Anda.

Perbedaan Ubin 3D

3D Tiles saat ini tidak mendukung metadata geometris dan semantik, yang berarti bahwa hierarki mesh dari model asli tidak tersedia untuk fitur pemilihan sub-model. Anda masih dapat menambahkan widget ke model Ubin 3D Anda, tetapi Anda tidak dapat menggunakan fitur yang disetel dengan baik ke sub-model: shader model, transformasi 3D terpisah, atau pengikatan entitas untuk mesh sub-model.

Disarankan untuk menggunakan konversi Ubin 3D untuk aset besar yang berfungsi sebagai konteks untuk latar belakang adegan. Jika Anda ingin sub-model dipecah dan dianotasi lebih lanjut maka itu harus diekstraksi sebagai aset GLTF/GLB terpisah dan ditambahkan langsung ke tempat kejadian. Ini dapat dilakukan dengan alat 3D gratis dan umum seperti [Blender](#).

Contoh kasus penggunaan:

- Anda memiliki model pabrik 1GB dengan ruang dan lantai mesin yang terperinci, kotak listrik, dan pipa ledeng. Kotak listrik dan pipa perlu menyala merah ketika data properti terkait melewati ambang batas.
- Anda mengisolasi kotak dan jerat pipa dalam model dan mengekspornya ke glTF terpisah menggunakan Blender.
- Anda mengubah pabrik tanpa elemen listrik dan pipa menjadi model Ubin 3D dan mengunggahnya ke S3.
- Anda menambahkan model 3D Tiles dan model glTF ke AWS IoT TwinMaker adegan di asal (0,0,0).

- Anda menambahkan komponen model shader ke kotak listrik dan sub-model pipa glTF untuk membuat jerat merah berdasarkan aturan properti.

Adekan dinamis

AWS IoT TwinMaker adegan membuka kekuatan [grafik pengetahuan](#) dengan menyimpan node adegan dan pengaturan dalam komponen entitas. Gunakan AWS IoT TwinMaker konsol untuk membuat adegan dinamis agar lebih mudah mengelola, membangun, dan merender adegan 3D.

Fitur utama:

- Semua objek node adegan 3D, pengaturan, dan binding data dirender “dinamis” berdasarkan kueri grafik pengetahuan.
- Jika Anda menggunakan Scene Viewer hanya-baca di Grafana atau aplikasi khusus, Anda bisa mendapatkan pembaruan untuk adegan Anda pada interval 30 detik.

Adekan statis versus dinamis

Adekan statis terdiri dari JSON file adegan yang disimpan di S3 yang memiliki detail semua node dan pengaturan adegan. Setiap perubahan pada adegan harus dilakukan pada JSON dokumen dan disimpan ke S3. Adegan statis adalah satu-satunya pilihan jika Anda memiliki [rencana harga dasar](#).

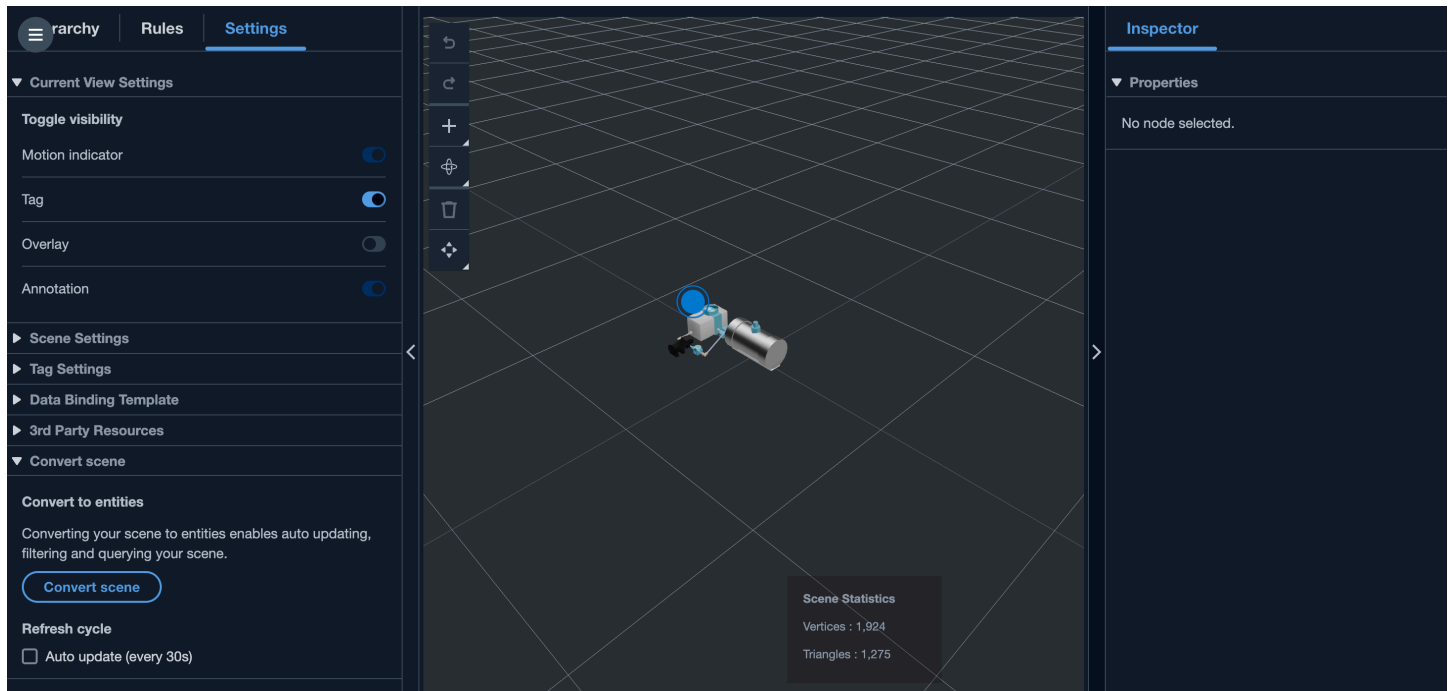
Adekan dinamis terdiri dari JSON file adegan yang memiliki pengaturan global untuk adegan, sementara semua node adegan dan pengaturan node lainnya disimpan sebagai komponen entitas dalam grafik pengetahuan. Pemandangan dinamis hanya didukung dalam paket harga bundel standar dan berjenjang. Lihat [Beralih mode AWS IoT TwinMaker harga](#) untuk informasi tentang cara meningkatkan paket harga Anda).

Anda dapat mengonversi adegan statis yang ada menjadi adegan dinamis dengan mengikuti langkah-langkah berikut:

- Arahkan ke adegan Anda di [AWS IoT TwinMaker konsol](#).
- Di panel sebelah kiri, klik tab Pengaturan.
- Perluas bagian Convert scene di bagian bawah panel.
- Klik tombol Konversi adegan, lalu klik Konfirmasi.

⚠ Warning

Konversi dari adegan statis ke dinamis tidak dapat diubah.



Jenis dan entitas komponen adegan

Untuk membuat komponen entitas khusus adegan, jenis komponen 1P berikut didukung:

- `com.amazon.iottwinmaker.3d.component.camera` Jenis komponen yang menyimpan pengaturan widget [kamera](#).
- `com.amazon.iottwinmaker.3d.component.dataoverlay` Jenis komponen yang menyimpan [pengaturan untuk hamparan anotasi atau widget tag](#).
- `com.amazon.iottwinmaker.3d.component.light` Jenis komponen yang menyimpan pengaturan widget ringan.
- `com.amazon.iottwinmaker.3d.component.modelref` Jenis komponen yang menyimpan pengaturan dan lokasi S3 dari model 3D yang digunakan dalam sebuah adegan.
- [com.amazon.iottwinmaker.3d.component.modelshader](#) Jenis komponen yang menyimpan [pengaturan shader model pada model 3D](#).
- `com.amazon.iottwinmaker.3d.component.motionindicator` Jenis komponen yang menyimpan pengaturan widget indikator gerak.

- `com.amazon.iottwinmaker.3d.component.submodelref` [Jenis komponen yang menyimpan pengaturan submodel model 3D](#).
- `com.amazon.iottwinmaker.3d.component.tag` Jenis komponen yang menyimpan pengaturan widget [tag](#).
- `com.amazon.iottwinmaker.3d.node` Jenis komponen yang menyimpan pengaturan dasar node adegan seperti transformasi 3D, nama, dan properti generiknya.

Konsep adegan dinamis

Entitas adegan dinamis disimpan di bawah entitas global berlabel `$SCENES`. Setiap adegan terdiri dari entitas root dan hierarki entitas turunan yang cocok dengan hierarki node adegan. Setiap node adegan di bawah root memiliki komponen `com.amazon.iottwinmaker.3d.node` dan komponen untuk jenis node (model 3D, widget, dan sebagainya).

Warning

Jangan menghapus entitas adegan secara manual atau adegan Anda mungkin dalam keadaan rusak. Jika Anda ingin menghapus sebagian atau seluruhnya adegan, gunakan halaman komposer adegan untuk menambah dan menghapus node adegan, dan gunakan halaman adegan untuk memilih dan menghapus adegan.

Buat aplikasi web yang disesuaikan menggunakan Komponen AWS IoT TwinMaker UI

AWS IoT TwinMaker menyediakan komponen UI open-source untuk pengembang AWS IoT Aplikasi. Dengan menggunakan komponen UI tersebut, pengembang dapat membangun aplikasi web yang disesuaikan dengan AWS IoT TwinMaker fitur yang diaktifkan untuk kembar digital mereka.

AWS IoT TwinMaker Komponen UI adalah bagian dari AWS IoT Application Kit, pustaka sisi klien open-source yang memungkinkan pengembang aplikasi IoT menyederhanakan pengembangan aplikasi IoT yang kompleks

AWS IoT TwinMaker Komponen UI meliputi:

- AWS IoT TwinMaker sumber:

Komponen konektor data yang memungkinkan Anda mengambil data dan berinteraksi dengan AWS IoT TwinMaker data dan kembar digital Anda.

Untuk informasi selengkapnya, lihat dokumentasi [AWS IoT TwinMaker sumber](#).

- Penampil adegan:

Komponen rendering 3D yang dibangun di atas `@react-three/fiber` yang membuat kembar digital Anda dan memungkinkan Anda untuk berinteraksi dengannya.

Untuk informasi selengkapnya, lihat dokumentasi [Scene Viewer](#).

- Pemutar video:

Komponen pemutar video yang memungkinkan Anda melakukan streaming video dari Kinesis Video AWS IoT TwinMaker Streams.

Untuk informasi selengkapnya, lihat dokumentasi [Pemutar Video](#).

Untuk mempelajari lebih lanjut tentang menggunakan AWS IoT Application Kit, silakan kunjungi halaman [AWS IoT Application Kit Github](#).

Untuk petunjuk tentang cara memulai aplikasi web baru menggunakan AWS IoT Application Kit, silakan kunjungi halaman dokumentasi [IoT App Kit](#) resmi.

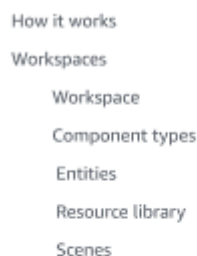
Beralih mode AWS IoT TwinMaker harga

AWS IoT TwinMaker saat ini memiliki tiga mode harga, dasar, standar atau bundel berjenjang. Mode penetapan harga standar ditetapkan sebagai mode harga default.





Anda dapat beralih dari mode harga berbasis penggunaan ke mode harga berbasis berjenjang kapan saja, tetapi perubahan tersebut berlaku di awal siklus penagihan berikutnya. Setelah Anda beralih dari mode harga berbasis penggunaan ke mode harga berbasis berjenjang, Anda tidak dapat beralih kembali ke mode harga berbasis penggunaan untuk tiga siklus penggunaan berikutnya. Jika Anda beralih dari dasar ke standar, perubahan akan segera berlaku. Untuk detail dan informasi biaya, lihat [AWS IoT TwinMaker Harga](#)

Prosedur ini menunjukkan cara mengganti mode harga Anda di [AWS IoT TwinMaker konsol](#):

1. Buka [konsol AWS IoT TwinMaker](#).
2. Di panel navigasi kiri, pilih Pengaturan. Halaman Harga terbuka.



How it works
Workspaces
 Workspace
 Component types
 Entities
 Resource library
 Scenes
Settings

What's new 
Documentation 
FAQ 
Pricing 

3. Pilih mode Ubah harga.
4. Pilih mode bundel Standar atau Berjenjang, seperti yang ditunjukkan pada gambar berikut.

Select price mode

Basic

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

Standard (current price mode)

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

Tiered bundle

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

Standard pricing

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

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

Cancel Save

5. Pilih Simpan untuk mengonfirmasi mode harga baru Anda.
6. Anda sekarang telah mengubah mode harga Anda.

i Note

Anda dapat beralih dari mode harga berbasis penggunaan ke mode harga berbasis berjenjang kapan saja, tetapi perubahan tersebut berlaku di awal siklus penagihan berikutnya. Setelah Anda beralih dari mode harga berbasis penggunaan ke mode harga berbasis berjenjang, Anda tidak dapat beralih kembali ke mode harga berbasis penggunaan untuk tiga siklus penggunaan berikutnya. Jika Anda beralih dari dasar ke standar, perubahan akan segera berlaku.

AWS IoT TwinMaker grafik pengetahuan

Grafik AWS IoT TwinMaker pengetahuan mengatur semua informasi yang terkandung dalam AWS IoT TwinMaker ruang kerja Anda dan menyajikannya dalam format grafik visual. Anda dapat menjalankan kueri terhadap entitas, komponen, dan jenis komponen untuk menghasilkan grafik visual yang menunjukkan hubungan antar sumber daya Anda AWS IoT TwinMaker .

Topik berikut menunjukkan cara menggunakan dan mengintegrasikan grafik pengetahuan.

Topik

- [AWS IoT TwinMaker konsep inti grafik pengetahuan](#)
- [Cara Menjalankan kueri grafik AWS IoT TwinMaker pengetahuan](#)
- [Integrasi adegan grafik pengetahuan](#)
- [Cara menggunakan grafik AWS IoT TwinMaker pengetahuan dengan Grafana](#)
- [AWS IoT TwinMaker grafik pengetahuan sumber daya tambahan](#)

AWS IoT TwinMaker konsep inti grafik pengetahuan

Topik ini mencakup konsep-konsep kunci dan kosakata dari fitur grafik pengetahuan.

Bagaimana grafik pengetahuan bekerja:

Grafik pengetahuan menciptakan hubungan antara entitas dan komponennya dengan yang ada [CreateEntity](#) atau [UpdateEntity](#) API. Hubungan hanyalah properti dari tipe data khusus [HUBUNGAN](#) yang didefinisikan pada komponen entitas. AWS IoT TwinMaker grafik pengetahuan memanggil [ExecuteQuery](#) API untuk membuat kueri berdasarkan data apa pun di entitas atau hubungan di antara mereka. Grafik pengetahuan menggunakan bahasa kueri PartiQL yang fleksibel (digunakan oleh AWS banyak layanan) yang memiliki dukungan sintaks kecocokan grafik yang baru ditambahkan untuk membantu Anda menulis kueri. Setelah panggilan dilakukan, Anda dapat melihat hasilnya sebagai tabel atau memvisualisasikannya sebagai grafik node dan tepi yang terhubung.

Istilah kunci grafik pengetahuan:

- Grafik entitas: Kumpulan node dan tepi dalam ruang kerja.
- Node: Setiap entitas di ruang kerja Anda menjadi simpul dalam grafik entitas.

- Edge: Setiap properti hubungan yang didefinisikan pada komponen entitas menjadi tepi dalam grafik entitas. Selain itu, hubungan orangtua-anak hierarkis yang didefinisikan menggunakan `parentEntityId` bidang entitas juga menjadi keunggulan dalam grafik entitas dengan nama hubungan `isChildOf ""`. Semua tepi adalah tepi terarah.
- AWS IoT TwinMaker Hubungan adalah jenis properti khusus dari komponen Entitas. Anda dapat menggunakan [UpdateEntity](#) API AWS IoT TwinMaker [CreateEntity](#) atau untuk menentukan dan mengedit hubungan. Dalam AWS IoT TwinMaker, hubungan harus didefinisikan dalam komponen entitas. Suatu hubungan tidak dapat didefinisikan sebagai sumber daya yang terisolasi. Suatu hubungan harus terarah dari satu entitas ke entitas lainnya.

Cara Menjalankan kueri grafik AWS IoT TwinMaker pengetahuan

Sebelum Anda menggunakan grafik AWS IoT TwinMaker pengetahuan, pastikan Anda telah menyelesaikan prasyarat berikut:

- Buat AWS IoT TwinMaker ruang kerja. Anda dapat membuat ruang kerja di [AWS IoT TwinMaker konsol](#).
- Menjadi akrab dengan AWS IoT TwinMaker sistem komponen entitas dan cara membuat entitas. Untuk informasi selengkapnya, lihat [Buat entitas pertama Anda](#).
- Menjadi akrab dengan AWS IoT TwinMaker konektor data. Untuk informasi selengkapnya, lihat [AWS IoT TwinMaker konektor data](#).

Note

Untuk menggunakan grafik AWS IoT TwinMaker pengetahuan, Anda harus berada dalam mode harga bundel standar atau berjenjang. Untuk informasi selengkapnya, lihat [Beralih mode AWS IoT TwinMaker harga](#).

Prosedur berikut menunjukkan cara menulis, menjalankan, menyimpan, dan mengedit kueri.

Buka editor kueri

Untuk menavigasi ke editor kueri grafik pengetahuan

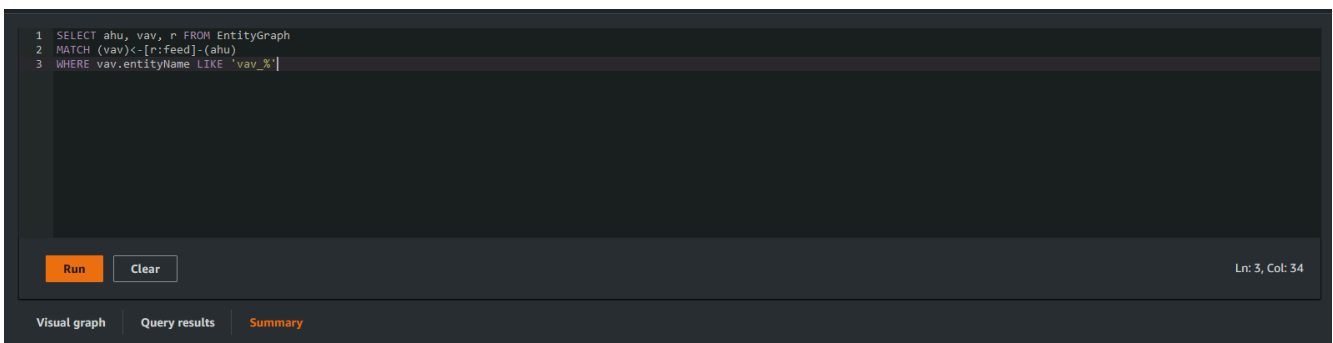
1. Buka [konsol AWS IoT TwinMaker](#).

2. Buka ruang kerja di mana Anda ingin menggunakan grafik pengetahuan.
3. Di menu navigasi kiri, pilih Editor kueri.
4. Editor kueri terbuka. Anda sekarang siap untuk menjalankan kueri pada sumber daya ruang kerja Anda.

Jalankan kueri

Untuk menjalankan kueri dan menghasilkan grafik

1. Di editor kueri, pilih tab Editor untuk membuka editor sintaks.
2. Di ruang editor, tulis kueri yang ingin Anda jalankan terhadap sumber daya ruang kerja Anda.



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

Dalam contoh yang ditampilkan, permintaan mencari entitas yang berisi `vav_%` namanya, lalu mengatur entitas ini berdasarkan feed hubungan di antara mereka, menggunakan kode berikut.

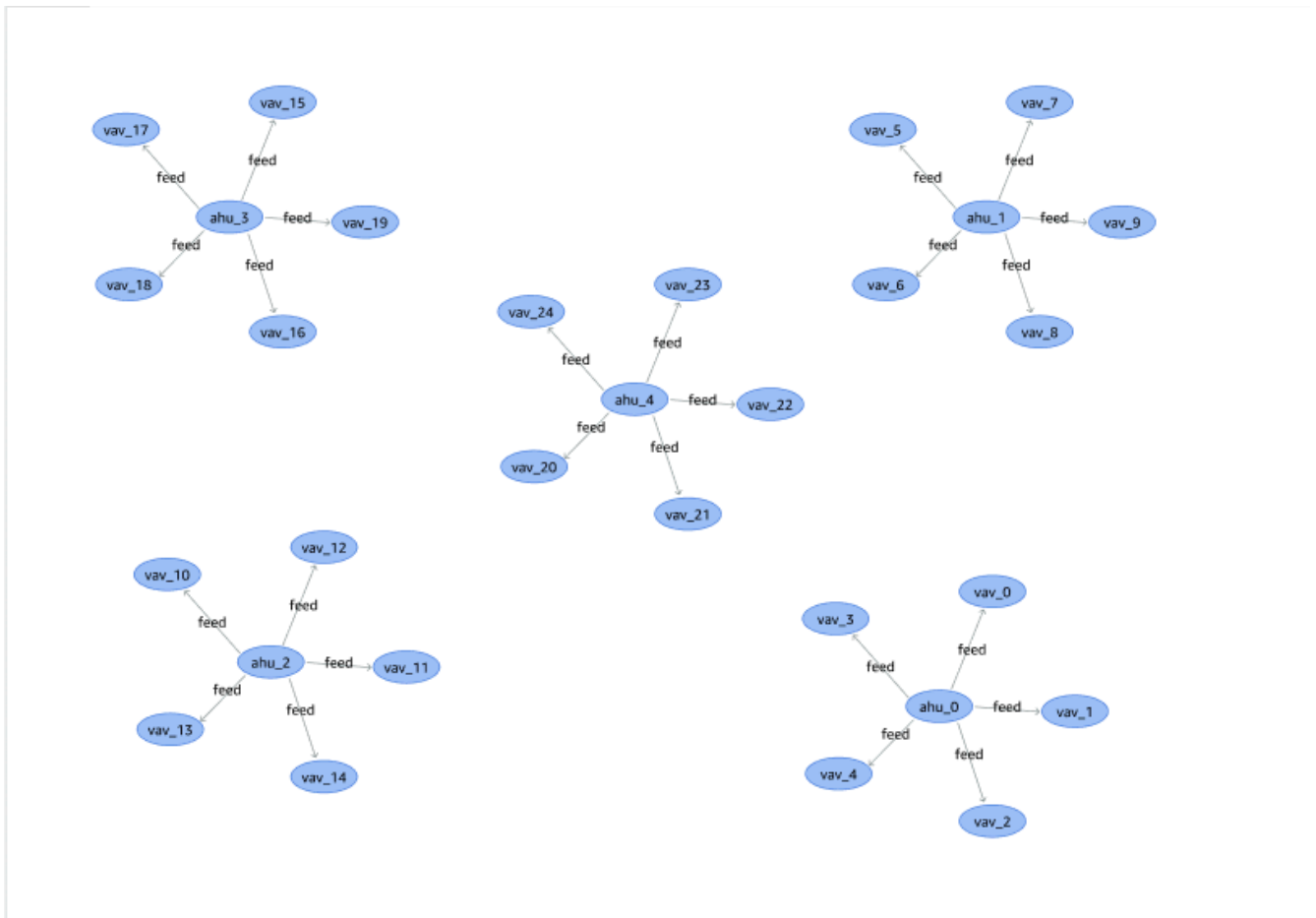
```
SELECT ahu, vav, r FROM EntityGraph
MATCH (vav)-[r:feed]-(ahu)
WHERE vav.entityName LIKE 'vav_%'
```

Note

Sintaks grafik pengetahuan menggunakan [PartiQL](#). Untuk informasi tentang sintaks ini, lihat [AWS IoT TwinMaker grafik pengetahuan sumber daya tambahan](#).

3. Pilih Jalankan kueri untuk menjalankan permintaan yang Anda buat.

Grafik dibuat berdasarkan permintaan Anda.



Grafik contoh yang ditunjukkan di atas didasarkan pada contoh kueri pada langkah 2.

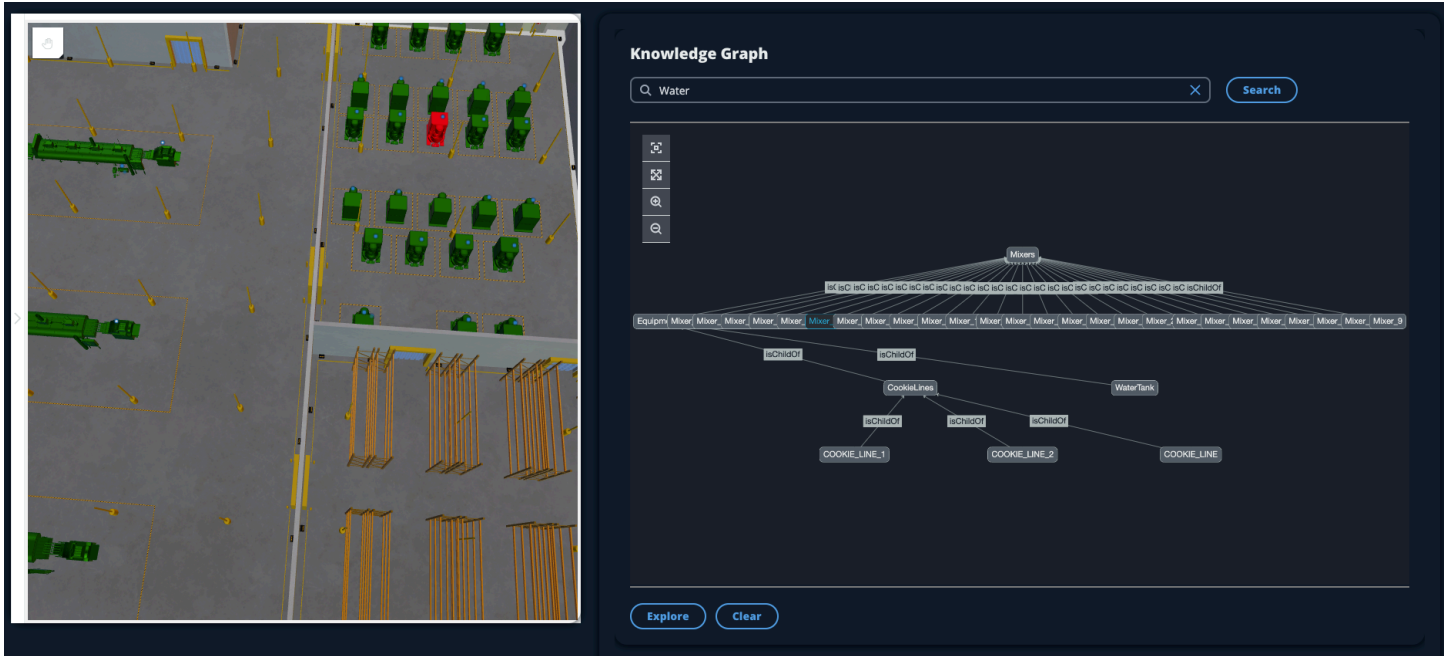
4. Hasil kueri juga disajikan dalam daftar. Pilih hasil untuk melihat hasil kueri dalam daftar.
5. Secara opsional, pilih Ekspor untuk mengeksport hasil kueri dalam format JSON atau CSV.

Ini mencakup penggunaan dasar grafik pengetahuan di konsol. Untuk informasi lebih lanjut dan contoh yang menunjukkan sintaks grafik pengetahuan, lihat [AWS IoT TwinMaker grafik pengetahuan sumber daya tambahan](#)

Integrasi adegan grafik pengetahuan

Anda dapat menggunakan komponen kit AWS IoT aplikasi untuk membuat aplikasi web yang mengintegrasikan grafik pengetahuan ke dalam AWS IoT TwinMaker adegan Anda. Ini memungkinkan Anda untuk menghasilkan grafik berdasarkan node 3D (model 3D yang mewakili peralatan atau sistem Anda) yang ada dalam adegan Anda. Untuk membuat aplikasi yang membuat grafik node 3D dari adegan Anda, pertama-tama ikat node 3D ke entitas di ruang kerja Anda. Dengan

pemetaan ini, AWS IoT TwinMaker buat grafik hubungan antara model 3D yang ada di adegan Anda dan entitas di ruang kerja Anda. Kemudian Anda dapat membuat aplikasi web, memilih model 3D dengan adegan Anda, dan menjelajahi hubungan mereka dengan entitas lain dalam format grafik.



Untuk contoh aplikasi web yang berfungsi yang menggunakan komponen AWS IoT app kit untuk menghasilkan grafik dalam sebuah AWS IoT TwinMaker adegan, lihat [AWS IoT TwinMaker contoh aplikasi react](#) di github.

AWS IoT TwinMaker prasyarat grafik adegan

Sebelum Anda membuat aplikasi web yang menggunakan grafik AWS IoT TwinMaker pengetahuan dalam adegan Anda, lengkapi prasyarat berikut:

- Buat AWS IoT TwinMaker ruang kerja. Anda dapat membuat ruang kerja di [AWS IoT TwinMaker konsol](#).
- Menjadi akrab dengan AWS IoT TwinMaker sistem komponen entitas dan cara membuat entitas. Untuk informasi selengkapnya, lihat [Buat entitas pertama Anda](#).
- Buat AWS IoT TwinMaker adegan yang diisi dengan model 3D.
- Menjadi akrab dengan AWS IoT TwinMaker komponen kit AWS IoT aplikasi. Untuk informasi lebih lanjut tentang AWS IoT TwinMaker komponen, lihat [Buat aplikasi web yang disesuaikan menggunakan Komponen AWS IoT TwinMaker UI](#).
- Menjadi familiar dengan konsep grafik pengetahuan dan terminologi kunci. Lihat [AWS IoT TwinMaker konsep inti grafik pengetahuan](#).

Note

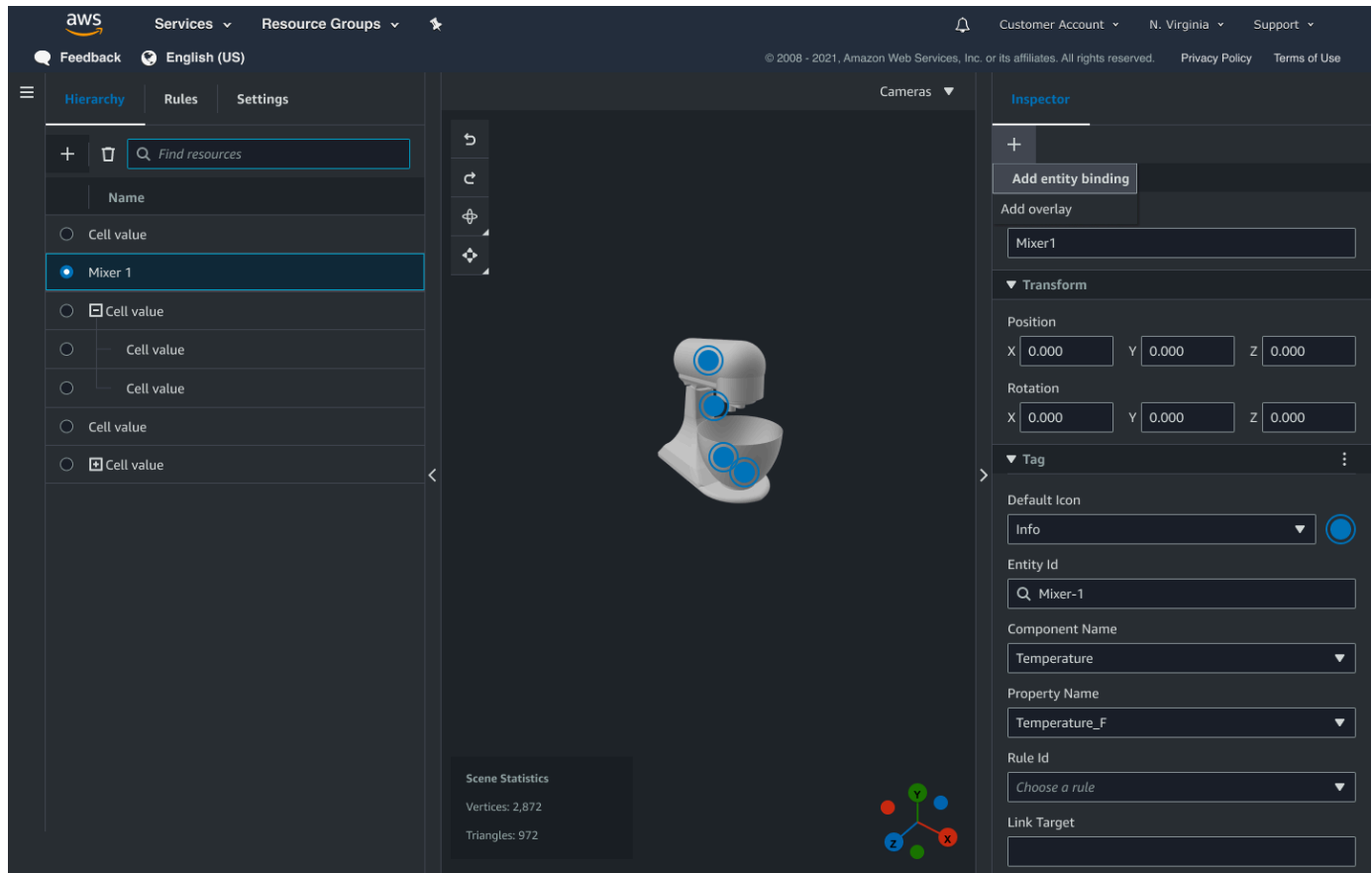
Untuk menggunakan grafik AWS IoT TwinMaker pengetahuan dan fitur terkait, Anda harus berada dalam mode harga bundel standar atau berjenjang. Untuk informasi lebih lanjut tentang AWS IoT TwinMaker harga, lihat [Beralih mode AWS IoT TwinMaker harga](#).

Mengikat node 3D di adegan Anda

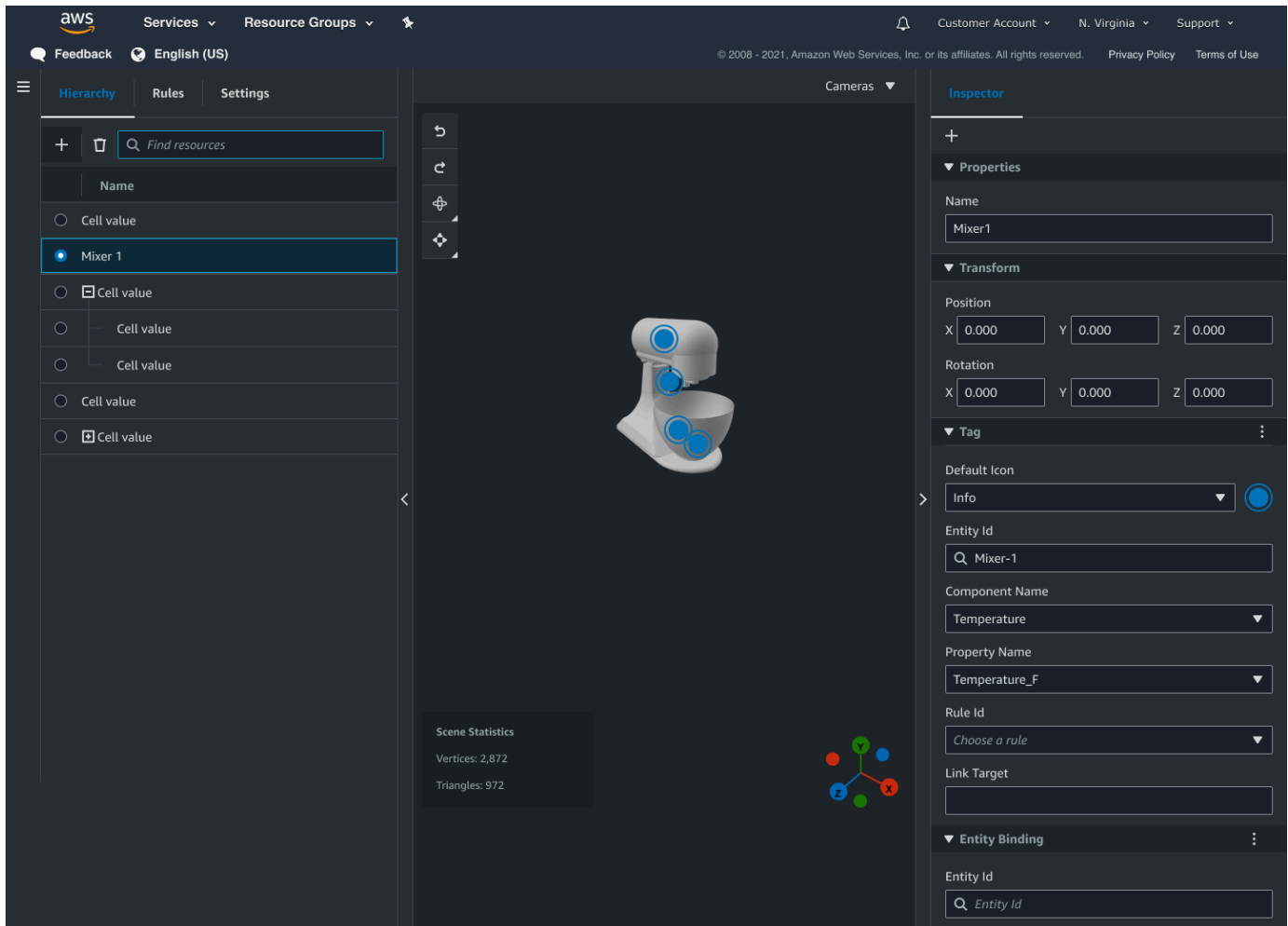
Sebelum Anda membuat aplikasi web yang mengintegrasikan grafik pengetahuan dengan adegan Anda, ikat model 3D, yang disebut sebagai node 3D, yang ada di adegan Anda ke entitas ruang kerja terkait. Misalnya, jika Anda memiliki model peralatan mixer dalam sebuah adegan, dan entitas terkait yang disebut `mixer_0`, buat pengikatan data antara model mixer dan entitas yang mewakili mixer—sehingga model dan entitas dapat dibuat grafik.

Untuk melakukan tindakan pengikatan data

1. Masuk ke [AWS IoT TwinMaker konsol](#).
2. Buka ruang kerja Anda dan pilih adegan dengan node 3D yang ingin Anda ikat.
3. Pilih simpul (model 3D) di komposer adegan. Ketika Anda memilih node, itu akan membuka panel inspektur di sisi kanan layar.
4. Di panel inspektur, arahkan ke bagian atas panel dan pilih tombol +. Kemudian pilih opsi Tambahkan pengikatan entitas. Ini akan membuka drop-down di mana Anda dapat memilih entitas untuk mengikat ke node yang Anda pilih saat ini.



5. Dari menu tarik-turun pengikatan data, pilih id entitas yang ingin Anda petakan ke model 3D. Untuk bidang nama Komponen dan nama Properti, pilih komponen dan properti yang ingin Anda ikat.



Setelah Anda membuat pilihan untuk bidang Entity Id, Component Name dan Property Name, pengikatan selesai.

6. Ulangi proses ini untuk semua model dan entitas yang ingin Anda grafik.

Note

Operasi pengikatan data yang sama dapat dilakukan pada tag adegan Anda, cukup pilih tag alih-alih entitas dan ikuti proses yang sama untuk mengikat tag ke node.

Buat aplikasi web

Setelah Anda mengikat entitas, gunakan library AWS IoT app kit untuk membuat aplikasi web dengan widget grafik pengetahuan yang memungkinkan Anda melihat adegan dan menjelajahi hubungan antara node adegan dan entitas.

Gunakan sumber daya berikut untuk membuat aplikasi Anda sendiri:

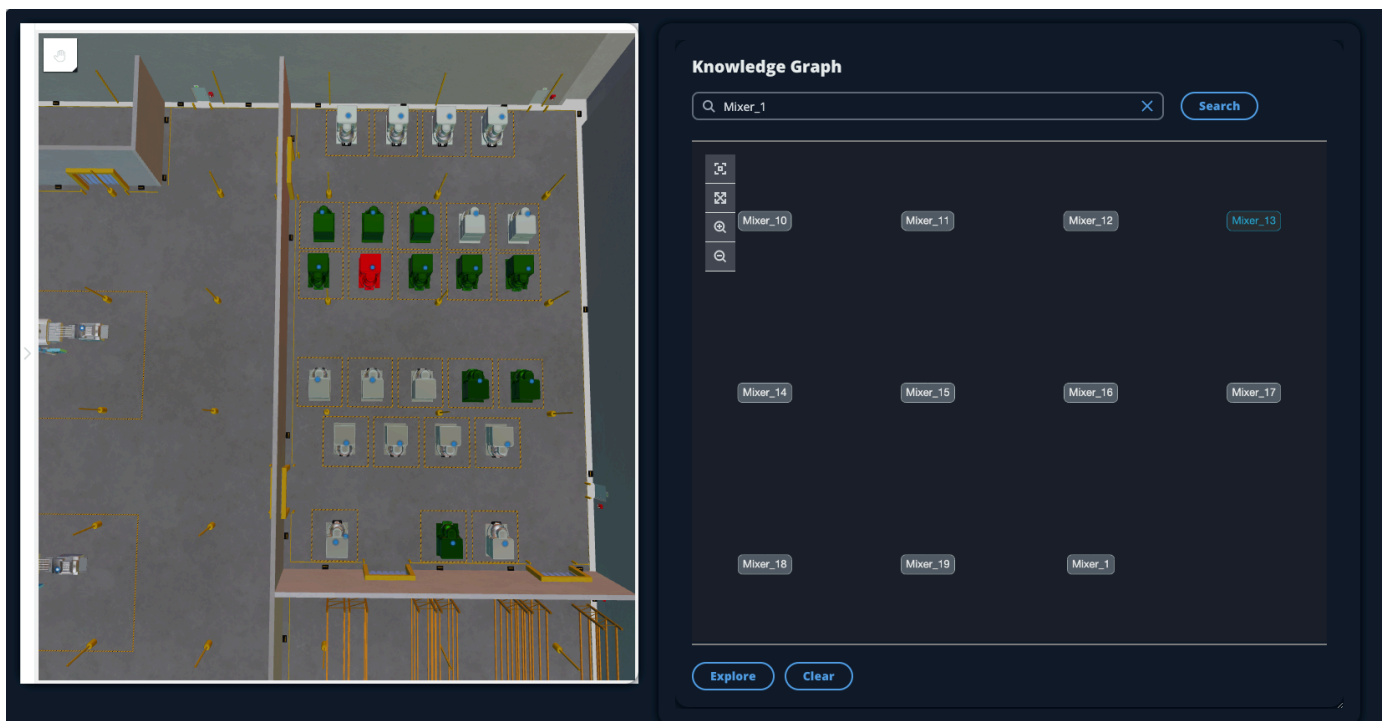
- AWS IoT TwinMaker Contoh aplikasi react github dokumentasi [Readme](#).
- AWS IoT TwinMaker Contoh [sumber](#) aplikasi reaksi di github.
- Kit AWS IoT aplikasi [Memulai](#) dokumentasi.
- Dokumentasi [komponen Pemutar Video](#) kit AWS IoT aplikasi.
- Dokumentasi [komponen Scene Viewer](#) kit AWS IoT aplikasi.

Prosedur berikut menunjukkan fungsionalitas komponen penampil adegan di aplikasi web.

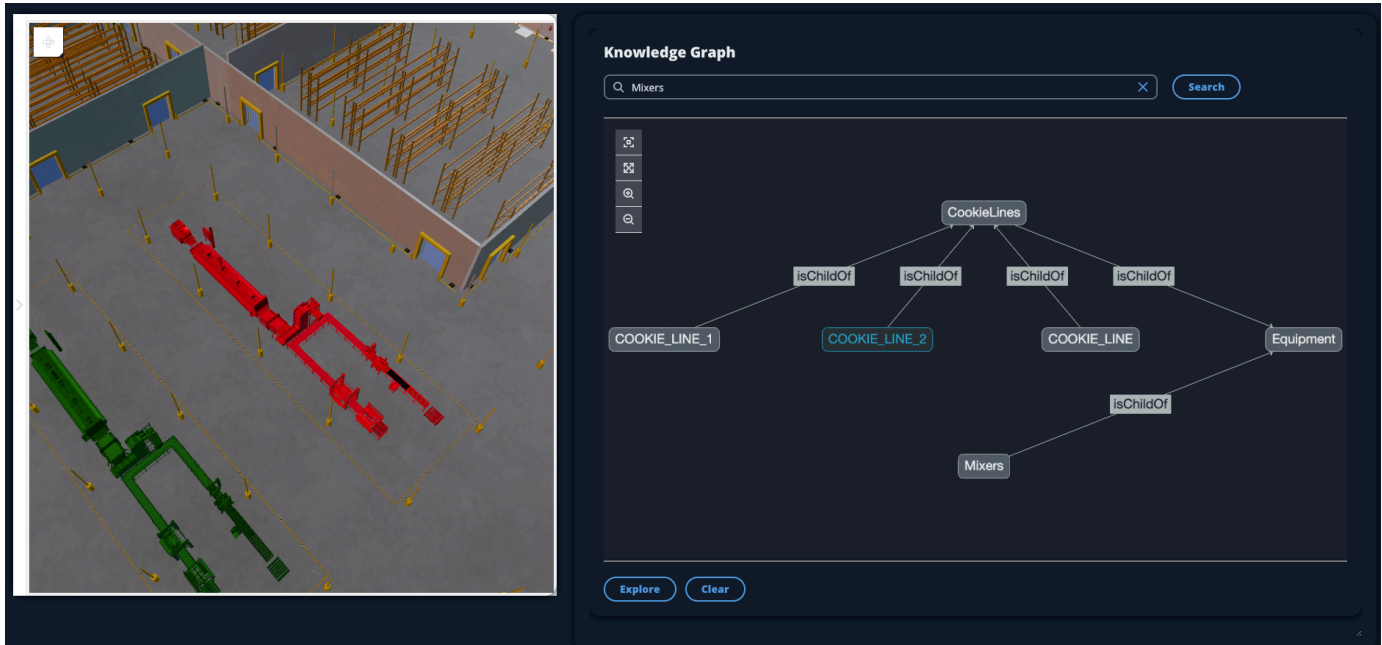
Note

Prosedur ini didasarkan pada implementasi komponen penampil adegan kit AWS IoT aplikasi di aplikasi reaksi AWS IoT TwinMaker sampel.

1. Buka komponen penampil adegan dari aplikasi reaksi AWS IoT TwinMaker sampel. Di bidang pencarian, ketik nama entitas atau nama entitas sebagian (pencarian peka huruf besar/kecil) lalu pilih tombol Cari. Jika model terikat ke id entitas, maka model dalam adegan akan disorot dan simpul entitas akan ditampilkan di panel penampil adegan.



- Untuk menghasilkan grafik dari semua hubungan, pilih simpul di widget penampil adegan dan pilih tombol Jelajahi.



- Tekan tombol Hapus untuk menghapus pilihan grafik Anda saat ini dan mulai dari awal.

Cara menggunakan grafik AWS IoT TwinMaker pengetahuan dengan Grafana

Bagian ini menunjukkan cara menambahkan panel editor kueri ke dasbor AWS IoT TwinMaker Grafana Anda untuk menjalankan dan menampilkan kueri.

AWS IoT TwinMaker prasyarat editor kueri

Sebelum Anda menggunakan grafik AWS IoT TwinMaker pengetahuan di Grafana, lengkapi prasyarat berikut:

- Buat AWS IoT TwinMaker ruang kerja. Anda dapat membuat ruang kerja di [AWS IoT TwinMaker konsol](#).
- Konfigurasi AWS IoT TwinMaker untuk digunakan dengan Grafana. Untuk petunjuk, lihat [AWS IoT TwinMakerIntegrasi dasbor Grafana](#).

Note

Untuk menggunakan grafik AWS IoT TwinMaker pengetahuan, Anda harus berada dalam mode harga bundel standar atau berjenjang. Untuk informasi selengkapnya, lihat [Beralih mode AWS IoT TwinMaker harga](#).

AWS IoT TwinMaker izin editor kueri

Untuk menggunakan editor AWS IoT TwinMaker kueri di Grafana, Anda harus memiliki peran IAM dengan izin untuk tindakan tersebut. `iottwinmaker:ExecuteQuery` Tambahkan izin tersebut ke peran dasbor ruang kerja Anda, seperti yang ditunjukkan dalam contoh ini:

```
{
  "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

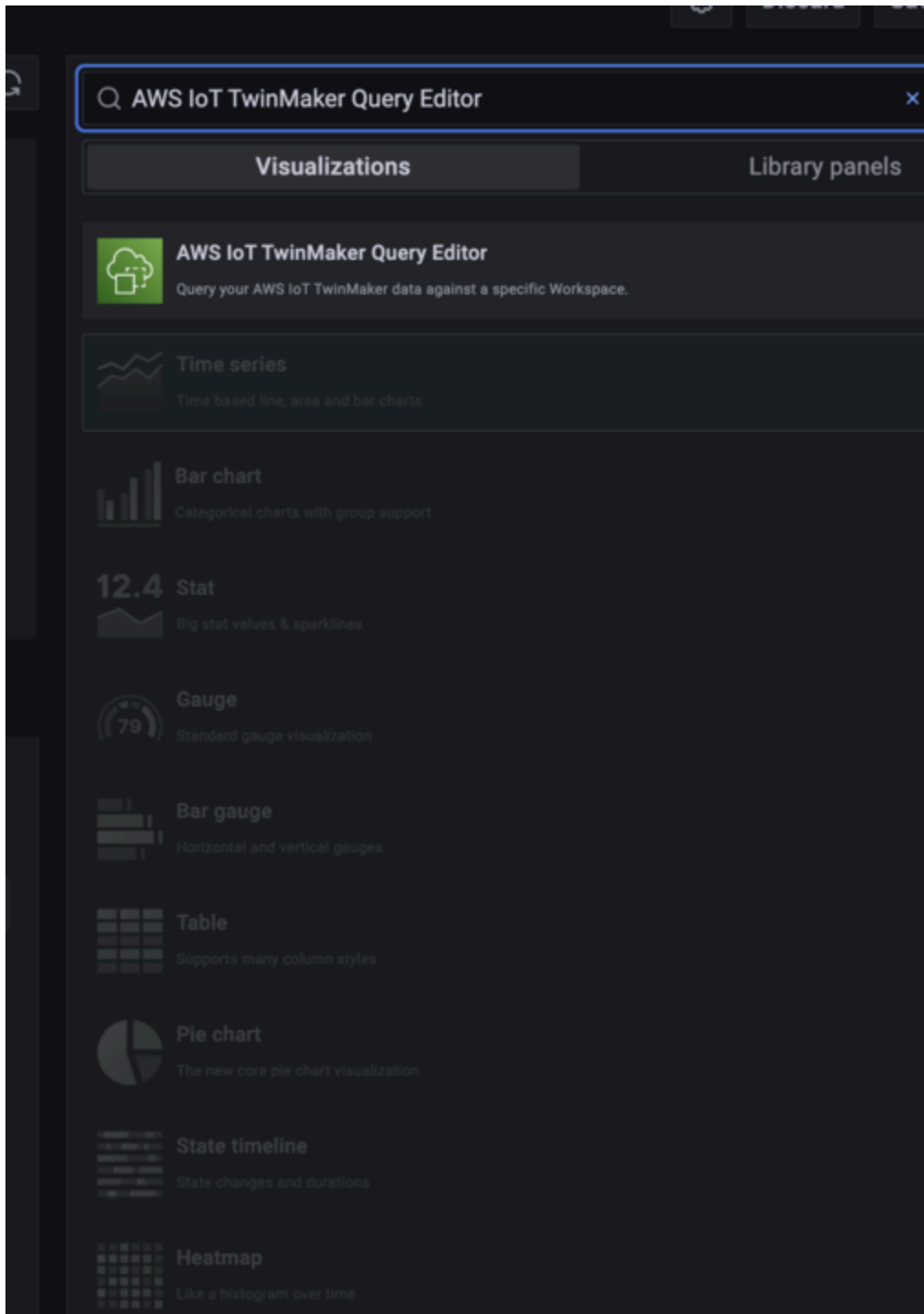
Saat Anda mengonfigurasi sumber data AWS IoT TwinMaker Grafana, pastikan untuk menggunakan peran dengan izin ini untuk bidang Asumsikan peran ARN. Setelah Anda menambahkannya, Anda dapat memilih ruang kerja Anda dari dropdown di sebelah Workspace.

Untuk informasi selengkapnya, lihat [Membuat peran IAM dasbor](#).

Siapkan panel editor AWS IoT TwinMaker kueri

Untuk menyiapkan panel dasbor Grafana baru untuk grafik pengetahuan

1. Buka dasbor AWS IoT TwinMaker Grafana Anda.
2. Buat panel dasbor baru. Untuk langkah-langkah mendetail tentang cara membuat panel, lihat [Membuat dasbor](#) di dokumentasi Grafana.
3. Dari daftar visualisasi, pilih AWS IoT TwinMaker Query Editor.



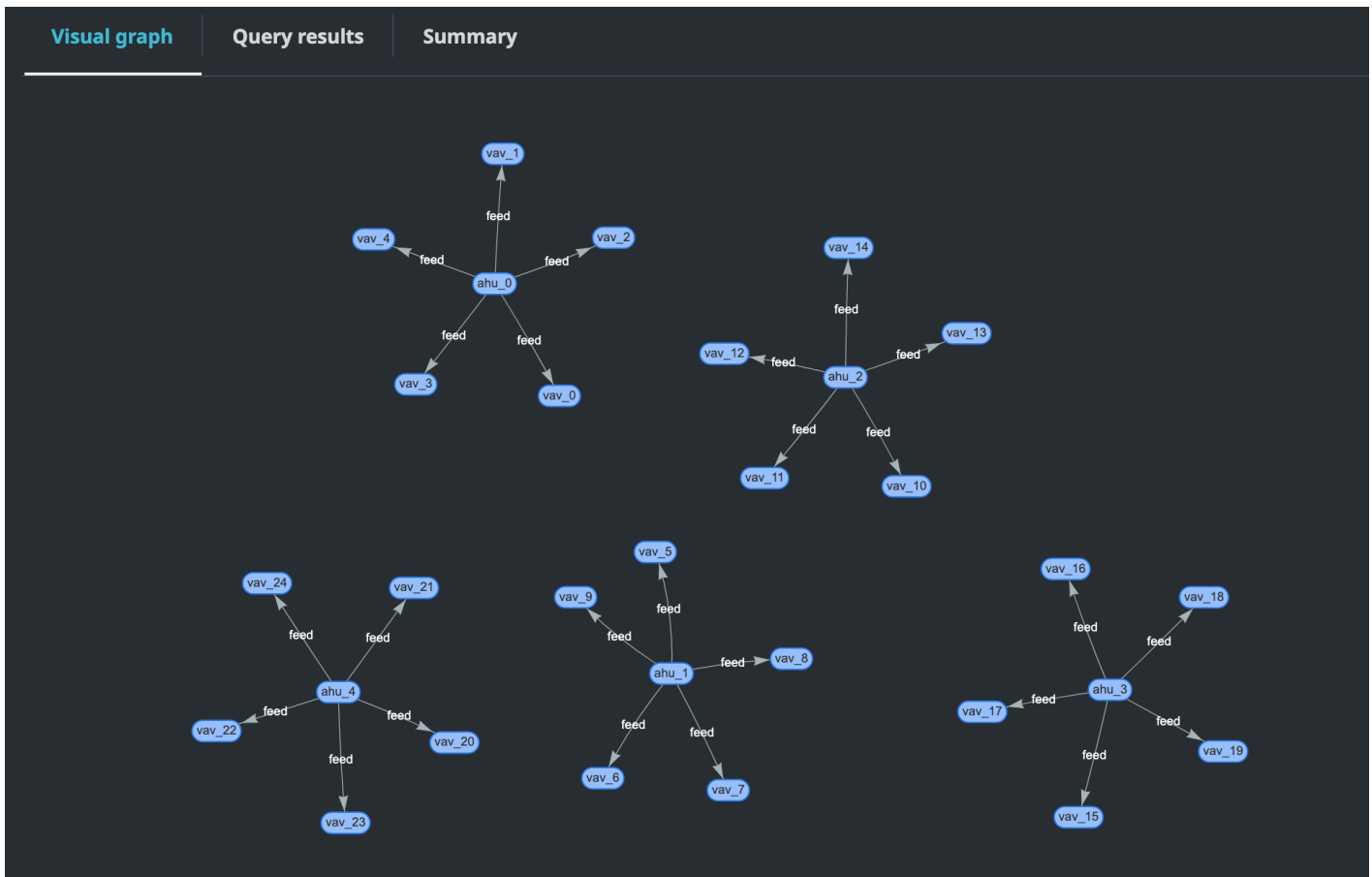
4. Pilih sumber data untuk menjalankan kueri.
5. (Opsional) Tambahkan nama untuk panel baru di bidang yang disediakan.
6. Pilih Terapkan untuk menyimpan dan mengonfirmasi panel baru Anda.

Panel grafik pengetahuan bekerja dengan cara yang sama seperti editor kueri yang disediakan di AWS IoT TwinMaker konsol. Anda dapat menjalankan, menulis, dan menghapus kueri yang Anda buat di panel. Untuk informasi selengkapnya tentang cara menulis kueri, lihat [AWS IoT TwinMaker grafik pengetahuan sumber daya tambahan](#).

Cara menggunakan editor AWS IoT TwinMaker kueri

Hasil kueri Anda ditampilkan dalam tiga cara, seperti yang ditunjukkan pada gambar berikut: divisualisasikan dalam grafik, tercantum dalam tabel, atau disajikan sebagai ringkasan run.

- Visualisasi grafik:



Grafik visual hanya menampilkan data untuk kueri yang memiliki setidaknya satu relasi dalam hasil. Grafik menampilkan entitas sebagai node dan hubungan sebagai tepi yang diarahkan dalam grafik.

- Data tabel:

Visual graph | **Query results** | Summary

Results returned (25) Export as ▾

Q Search query results < 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": "1667895668496", "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": "1667895668496", "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>

Format data tabular menampilkan data untuk semua kueri. Anda dapat mencari tabel untuk hasil tertentu atau himpunan bagian dari hasil. Data dapat diekspor dalam format JSON atau CSV.

- Jalankan ringkasan

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

Ringkasan run menampilkan kueri dan metadata tentang status kueri.

AWS IoT TwinMaker grafik pengetahuan sumber daya tambahan

Bagian ini memberikan contoh dasar sintaks PartiQL yang digunakan untuk menulis kueri dalam grafik pengetahuan, serta tautan ke dokumentasi PartiQL yang memberikan informasi tentang model data grafik pengetahuan.

- [Dokumentasi model data grafik PartiQL](#)
- [Dokumentasi kueri grafik PartiQL](#)

Kumpulan contoh ini menunjukkan kueri dasar dengan tanggapannya. Gunakan ini sebagai referensi untuk menulis pertanyaan Anda sendiri.

Kueri dasar

- Dapatkan semua entitas dengan filter

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

Query ini mengembalikan semua entitas dalam ruang kerja dengan nama `room_0`.

FROM klausa: `EntityGraph` adalah kumpulan grafik yang berisi semua entitas dan hubungannya di ruang kerja. Koleksi ini dibuat dan dikelola secara otomatis AWS IoT TwinMaker berdasarkan entitas di ruang kerja Anda.

MATCH klausa: menentukan pola yang cocok dengan sebagian dari grafik. Dalam hal ini, pola (`entity`) cocok dengan setiap node dalam grafik dan terikat ke variabel entitas. **FROM** Klausul harus diikuti oleh **MATCH** klausa.

WHERE klausa: menentukan filter pada `entityName` bidang node, di mana nilai harus cocok. `room_0`

SELECT klausa: menentukan `entity` variabel sehingga seluruh node entitas dikembalikan.

Tanggapan:

```
{
  "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` Mengembalikan metadata tentang kolom, seperti nama dan jenis. Jenis yang dikembalikan adalah `NODE`. Ini menunjukkan bahwa seluruh node telah dikembalikan. Nilai lain untuk tipe dapat `EDGE` yang akan menunjukkan hubungan atau `VALUE` yang akan menunjukkan nilai skalar seperti bilangan bulat atau string.

`rows` Mengembalikan daftar baris. Karena hanya satu entitas yang cocok, satu `rowData` dikembalikan yang berisi semua bidang dalam entitas.

Note

Tidak seperti SQL di mana Anda hanya dapat mengembalikan nilai skalar, Anda dapat mengembalikan objek (sebagai JSON) menggunakan PartiQL.

Setiap node berisi semua bidang tingkat entitas seperti `entityId`, `arn` dan `components`, bidang tingkat komponen seperti, dan serta bidang tingkat properti `properties` seperti `componentTypeId` dan `componentName`, semua sebagai JSON `propertyName` bersarang `propertyValue`.

- Dapatkan semua hubungan dengan filter:

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

Query ini mengembalikan semua hubungan dalam ruang kerja dengan nama `isLocationOf` hubungan.

MATCHKlausula: menentukan pola yang cocok dengan dua node (ditunjukkan oleh `()`) yang dihubungkan oleh tepi terarah (ditunjukkan oleh `-[]->`) dan terikat ke variabel yang disebut `relationship`

WHEREKlausula: menentukan filter pada `relationshipName` bidang tepi, di mana nilainya `isLocationOf`

SELECTKlausula: menentukan variabel hubungan sehingga seluruh simpul tepi dikembalikan.

Respons

```
{
  "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
]
}

```

Jenis kolom di dalam `columnDescriptions` adalah `EDGE`.

Masing-masing `rowData` mewakili tepi dengan bidang seperti `relationshipName`. Ini sama dengan nama properti hubungan yang didefinisikan pada entitas. `sourceEntityId`, `sourceComponentName` dan `sourceComponentTypeId` memberikan informasi tentang entitas dan komponen mana properti hubungan didefinisikan pada. `targetEntityId` menentukan entitas mana yang ditunjuk oleh hubungan ini.

- Dapatkan semua entitas dengan hubungan tertentu dengan entitas tertentu

```

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

```

Query ini mengembalikan semua nama entitas dari semua entitas yang memiliki `isLocationOf` hubungan dengan `room_0` entitas.

MATCH klausa: menentukan pola yang cocok dengan dua node (`e1`, `e2`) yang memiliki tepi terarah (`r`).

WHERE klausul: menentukan filter pada nama hubungan dan nama entitas sumber.

SELECT klausul: mengembalikan `entityName` bidang di `e2` node.

Respons

```

{
  "columnDescriptions": [
    {
      "name": "entityName",
      "type": "VALUE"
    }
  ]
}

```

```

    ],
    "rows": [
      {
        "rowData": [
          "floor_0"
        ]
      }
    ]
  }
}

```

Dalam ColumnDescriptions, jenis kolom adalah VALUE karena entityName adalah string.

Satu entitas floor_0,, dikembalikan.

PERTANDINGAN

Pola berikut didukung dalam MATCH klausa:

- Cocokkan simpul 'b' yang menunjuk ke simpul 'a':

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

- Cocokkan simpul 'a' yang menunjuk ke simpul 'b':

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

Tidak ada variabel yang terikat pada hubungan dengan asumsi filter tidak perlu ditentukan pada hubungan tersebut.

- Cocokkan simpul 'a' yang menunjuk ke simpul 'b' dan simpul 'b' menunjuk ke simpul 'a':

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

Ini akan mengembalikan dua kecocokan: satu dari 'a' ke 'b' dan satu lagi dari 'b' ke 'a', jadi rekomendasinya adalah menggunakan tepi yang diarahkan sedapat mungkin.

- Nama hubungan juga merupakan label grafik properti EntityGraph, jadi Anda cukup menentukan nama hubungan mengikuti titik dua (:). Alih-alih menentukan filter di rel.relationshipName dalam WHERE klausa.

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

- Rantai: pola dapat dirantai agar sesuai dengan beberapa hubungan.


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

- Pola hop variabel dapat menjangkau beberapa node dan tepi juga:

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

Kueri ini cocok dengan pola apa pun dengan tepi keluar dari simpul 'a' dalam 1 hingga 5 hop. Kuantifier yang diizinkan adalah:

{m, n}- antara m dan n pengulangan

{m, }- m atau lebih pengulangan.

DARI:

Node entitas dapat berisi data bersarang, seperti komponen yang berisi data bersarang lebih lanjut seperti properti. Ini dapat diakses dengan menghapus hasil pola 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'
```

Akses bidang bersarang dengan mendotting . ke dalam variabel. Koma (,) digunakan untuk menghapus (atau menggabungkan) entitas dengan komponen di dalamnya dan kemudian properti di dalam komponen tersebut. AS digunakan untuk mengikat variabel ke variabel unnested sehingga mereka dapat digunakan dalam atau klausa. WHERE SELECT Query ini mengembalikan semua entitas yang berisi properti bernama roomFunction dengan nilai meeting dalam komponen dengan id tipe komponen com.example.query.construction.room

Untuk mengakses beberapa bidang bersarang dari bidang seperti beberapa komponen dalam entitas, gunakan notasi koma untuk melakukan penggabungan.

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

PILIH:

- Kembalikan simpul:

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

- Kembalikan tepi:

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

- Mengembalikan nilai skalar:

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

Format nama bidang output dengan aliasing menggunakan `AS`. Di sini, bukan `propertyValue` sebagai nama kolom dalam respons, `roomfunction` dikembalikan.

- Kembalikan alias:

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

Menggunakan alias sangat disarankan untuk eksplisit, meningkatkan keterbacaan, dan menghindari ambiguitas dalam kueri Anda.

DIMANA:

- Operator logis yang didukung adalah `AND`, `NOT`, dan `OR`.
- Operator perbandingan yang didukung adalah `<`, `<=`, `>`, `=>`, `=`, dan `!=`.
- Gunakan `IN` kata kunci jika Anda ingin menentukan beberapa `OR` kondisi pada bidang yang sama.
- Filter pada bidang entitas, komponen, atau properti:

```
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'
```

- Filter pada configuration properti. Berikut unit adalah kunci dalam peta konfigurasi dan Celsius nilainya.

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

- Periksa apakah properti peta berisi kunci dan nilai yang diberikan:

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

- Periksa apakah properti peta berisi kunci yang diberikan:

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

- Periksa apakah properti list berisi nilai yang diberikan:

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

- Gunakan `lower()` fungsi untuk perbandingan case insensitive. Secara default, semua perbandingan peka huruf besar/kecil.

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

SUKA:

Berguna jika Anda tidak tahu nilai pasti untuk bidang dan dapat melakukan pencarian teks lengkap pada bidang yang ditentukan. %mewakili nol atau lebih.

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

- Pencarian Infix: %room%
- Pencarian awalan: room%
- Pencarian akhiran: %room
- Jika Anda memiliki '%' dalam nilai Anda, maka letakkan karakter escape di LIKE dan tentukan karakter escape dengan ESCAPE.

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

BERBEDA:

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

- **DISTINCT**Kata kunci menghilangkan duplikat dari hasil akhir.

DISTINCTtidak didukung pada tipe data yang kompleks.

MENGHITUNG

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

- **COUNT**Kata kunci menghitung jumlah item dalam hasil query.
- **COUNT**tidak didukung pada bidang kompleks bersarang dan bidang pola grafik.
- **COUNT**agregasi tidak didukung dengan **DISTINCT** dan kueri bersarang.

Misalnya, **COUNT(DISTINCT e.entityId)** tidak didukung.

JALAN

Proyeksi pola berikut didukung dalam kueri menggunakan proyeksi jalur:

- Kueri hop variabel

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

Kueri ini cocok dan memproyeksikan metadata node dari pola apa pun dengan tepi keluar dari simpul a dalam 1 hingga 3 hop.

- Kueri hop tetap

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

Kueri ini mencocokkan dan memproyeksikan metadata entitas dan tepi yang masuk ke b.

- Kueri tidak terarah

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

Kueri ini cocok dan memproyeksikan metadata node dalam 1 pola hop yang menghubungkan a dan c melalui 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"
  }
]
}
]
}
]
}
}

```

Respons *PATH* kueri ini hanya terdiri dari metadata yang mengidentifikasi semua node dan tepi setiap jalur/pola antara a dan c melalui b.

LIMIT dan OFFSET:

```

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

```

LIMIT menentukan jumlah hasil yang akan dikembalikan dalam query, dan **OFFSET** menentukan jumlah hasil untuk dilewati.

LIMIT dan maxResults:

Contoh berikut menunjukkan kueri yang mengembalikan 500 hasil secara total, tetapi hanya menampilkan 50 pada satu waktu per panggilan API. Pola ini dapat digunakan di mana Anda perlu membatasi jumlah hasil yang ditampilkan, misalnya jika Anda hanya dapat menampilkan 50 hasil di UI.

```

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

```

- `LIMIT` kata kunci mempengaruhi query dan membatasi baris yang dihasilkan. Jika Anda perlu mengontrol jumlah hasil yang dikembalikan per panggilan API tanpa membatasi jumlah total hasil yang dikembalikan, gunakan `LIMIT`.
- `max-results` adalah parameter opsional untuk [tindakan ExecuteQuery API](#). `max-result`shanya berlaku untuk API dan bagaimana hasil dibaca dalam batas-batas kueri di atas.

Menggunakan `max-results` dalam kueri memungkinkan Anda untuk mengurangi jumlah hasil yang ditampilkan tanpa membatasi jumlah aktual hasil yang dikembalikan.

Kueri di bawah ini berulang melalui halaman hasil berikutnya. Kueri ini menggunakan panggilan `ExecuteQuery API` untuk mengembalikan baris 51-100, di mana halaman hasil berikutnya ditentukan oleh `next-token` — dalam hal ini tokennya adalah: `"H7kyGmvK376L"`

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

- `next-tokenString` menentukan halaman berikutnya dari hasil. Untuk informasi selengkapnya, lihat tindakan [ExecuteQuery API](#).

AWS IoT TwinMaker kueri grafik pengetahuan memiliki batasan berikut:

Batasi Nama	Kuota	Dapat disesuaikan
Batas waktu eksekusi kueri	10 detik	Tidak
Jumlah hop maksimum	10	Ya
Jumlah maksimum self JOIN s	20	Ya
Jumlah maksimum bidang yang diproyeksikan	20	Ya
Jumlah maksimum ekspresi bersyarat (AND,OR,NOT)	10	Ya

Batasi Nama	Kuota	Dapat disesuaikan
Panjang maksimum pola LIKE ekspresi (termasuk wildcard dan escapes)	20	Ya
Jumlah maksimum item yang dapat ditentukan dalam IN klausa	10	Ya
Nilai maksimum untuk OFFSET	3000	Ya
Nilai maksimum untuk LIMIT	3000	Ya
Nilai maksimum untuk traversal (+) OFFSET LIMIT	3000	Ya

Sinkronisasi aset dengan AWS IoT SiteWise

AWS IoT TwinMaker mendukung sinkronisasi aset (sinkronisasi aset) untuk AWS IoT SiteWise aset dan model aset Anda. Menggunakan tipe AWS IoT SiteWise komponen, sinkronisasi aset mengambil AWS IoT SiteWise aset dan model aset yang ada dan mengubah sumber daya ini menjadi AWS IoT TwinMaker entitas, komponen, dan tipe komponen. Bagian berikut memandu Anda tentang cara mengonfigurasi sinkronisasi aset dan aset serta model AWS IoT SiteWise aset mana yang dapat disinkronkan ke ruang AWS IoT TwinMaker kerja Anda.

Topik

- [Menggunakan sinkronisasi aset dengan AWS IoT SiteWise](#)
- [Perbedaan antara ruang kerja khusus dan default](#)
- [Sumber daya disinkronkan dari AWS IoT SiteWise](#)
- [Menganalisis status sinkronisasi dan kesalahan](#)
- [Hapus pekerjaan sinkronisasi](#)
- [Batas sinkronisasi aset](#)

Menggunakan sinkronisasi aset dengan AWS IoT SiteWise

Topik ini menunjukkan cara mengaktifkan dan mengonfigurasi sinkronisasi AWS IoT SiteWise aset. Ikuti prosedur yang sesuai berdasarkan jenis ruang kerja yang Anda gunakan.

Important

Lihat [the section called “Perbedaan antara ruang kerja khusus dan default”](#) untuk informasi tentang perbedaan antara ruang kerja kustom dan default.

Topik

- [Menggunakan ruang kerja khusus](#)
- [Menggunakan IoT SiteWiseDefaultWorkspace](#)

Menggunakan ruang kerja khusus

Tinjau prasyarat ini sebelum mengaktifkan sinkronisasi aset.

Prasyarat

Sebelum menggunakan AWS IoT SiteWise, hal-hal berikut harus diselesaikan:

- Anda memiliki AWS IoT TwinMaker ruang kerja.
- Anda memiliki aset dan model aset di AWS IoT SiteWise. Untuk informasi selengkapnya, lihat [Membuat model aset](#).
- Peran IAM yang ada dengan izin baca untuk tindakan berikut: AWS IoT SiteWise
 - ListAssets
 - ListAssetModels
 - DescribeAsset
 - DescribeAssetModel
- Peran IAM harus memiliki izin menulis berikut untuk: AWS IoT TwinMaker
 - CreateEntity
 - UpdateEntity
 - DeleteEntity
 - CreateComponentType
 - UpdateComponentType
 - DeleteComponentType
 - ListEntities
 - GetEntity
 - ListComponentTypes

Gunakan peran IAM berikut sebagai templat untuk peran yang diperlukan:

```
// trust relationships
{
  {
    "Version": "2012-10-17",
    "Statement": [
      {
        "Effect": "Allow",
        "Principal": {
          "Service": [
```

```

        "iottwinmaker.amazonaws.com"
    ]
},
    "Action": "sts:AssumeRole"
}
]
}

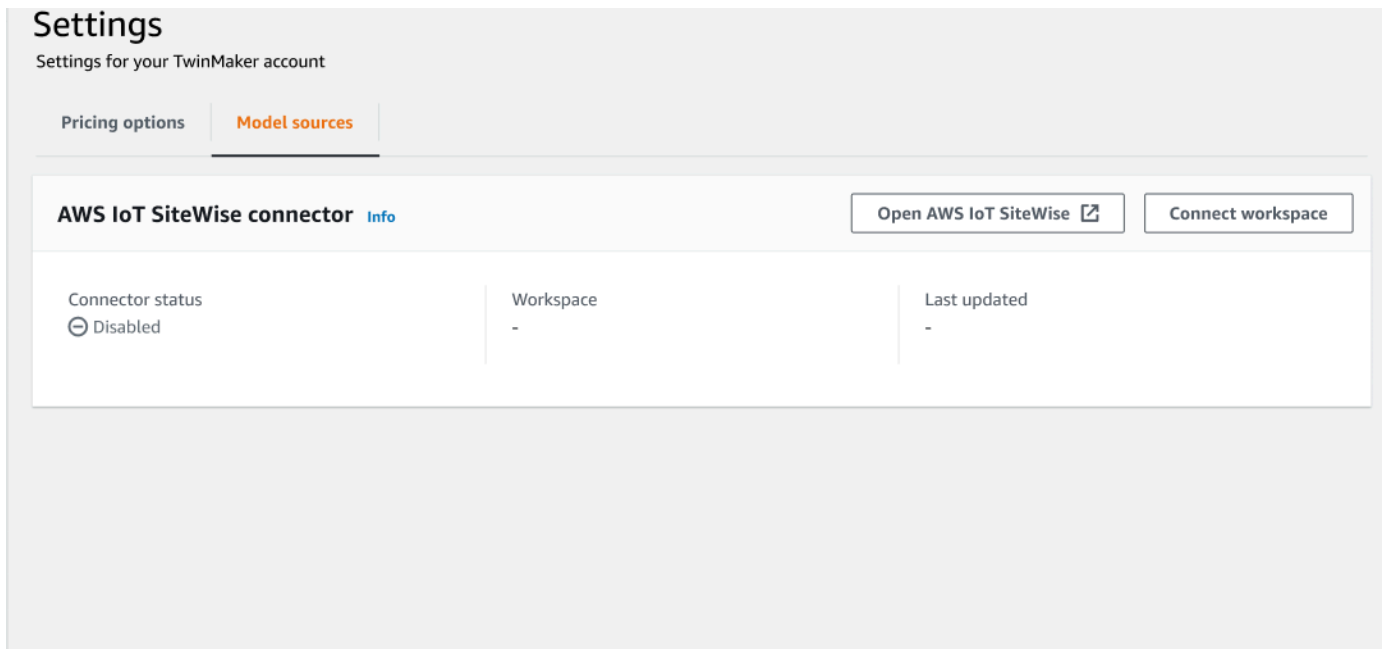
// permissions - replace ACCOUNT_ID, REGION, WORKSPACE_ID with actual values
{
    "Version": "2012-10-17",
    "Statement": [{
        "Sid": "SiteWiseAssetReadAccess",
        "Effect": "Allow",
        "Action": [
            "iotsitewise:DescribeAsset"
        ],
        "Resource": [
          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

```

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

Gunakan prosedur berikut untuk mengaktifkan dan mengonfigurasi sinkronisasi AWS IoT SiteWise aset.

1. Di [AWS IoT TwinMaker konsol](#), navigasikan ke halaman Pengaturan.
2. Buka tab Sumber model.



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	-	-

- Pilih **Connect workspace** untuk menautkan AWS IoT TwinMaker ruang kerja Anda ke aset Anda AWS IoT SiteWise .

Note

Anda hanya dapat menggunakan sinkronisasi aset dengan satu AWS IoT TwinMaker ruang kerja. Anda harus memutuskan sinkronisasi dari satu ruang kerja dan terhubung ke ruang kerja lain jika Anda ingin menyinkronkan di ruang kerja yang berbeda.

- Selanjutnya, navigasikan ke ruang kerja tempat Anda ingin menggunakan sinkronisasi aset.
- Pilih **Tambahkan sumber**. Ini membuka halaman sumber model entitas **Tambah**.

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

Add entity model source

Add an entity model source to your workspace.

Add entity model source

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

AWS IoT SiteWise

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

IAM role
This role will be used for XYZ.

Select IAM role

- Pada halaman Tambah sumber model entitas, konfirmasi bahwa bidang sumber ditampilkan AWS IoT SiteWise. Pilih peran IAM yang Anda buat sebagai prasyarat untuk peran IAM.
- Anda sekarang telah mengaktifkan sinkronisasi AWS IoT SiteWise aset. Anda akan melihat spanduk konformasi muncul di bagian atas halaman Workspace yang dipilih yang mengonfirmasi bahwa sinkronisasi aset aktif. Anda juga sekarang harus melihat sumber sinkronisasi yang tercantum di bagian Sumber model entitas.

cookieFactory [Info](#) View Delete

Workspace information

Edit

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)	

Entity model sources (1)

Add source

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

Menggunakan IoT SiteWiseDefaultWorkspace

Saat Anda memilih [AWS IoT SiteWiseAWS IoT TwinMaker integrasi](#), ruang kerja default bernama `IoTSiteWiseDefaultWorkspace` dibuat dan disinkronkan secara otomatis. AWS IoT SiteWise

Anda juga dapat menggunakan AWS IoT TwinMaker `CreateWorkspace` API untuk membuat ruang kerja bernama `IoTSiteWiseDefaultWorkspace`.

Prasyarat

Sebelum membuat `IoTSiteWiseDefaultWorkspace`, pastikan Anda telah melakukan hal berikut:

- Buat peran AWS IoT TwinMaker terkait layanan. Untuk informasi selengkapnya, lihat [Menggunakan peran terkait layanan untuk AWS IoT TwinMaker](#).
- Buka konsol IAM di <https://console.aws.amazon.com/iam/>.

Tinjau peran atau pengguna dan verifikasi bahwa ia memiliki izin `iot:SiteWiseIntegration:EnableSiteWiseIntegration`.

Jika diperlukan, tambahkan izin ke peran atau pengguna:

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

Perbedaan antara ruang kerja khusus dan default

Important

AWS IoT SiteWise Fitur baru, seperti [CompositionModel](#), hanya tersedia di `IoTSiteWiseDefaultWorkspace`. Kami mendorong Anda untuk menggunakan ruang kerja default alih-alih ruang kerja khusus.

Saat menggunakan `IoTSiteWiseDefaultWorkspace`, ada beberapa perbedaan penting dari menggunakan ruang kerja khusus dengan sinkronisasi aset.

- Saat Anda membuat ruang kerja default, lokasi Amazon S3 dan peran IAM bersifat opsional.

Note

Anda dapat menggunakan `UpdateWorkspace` untuk menyediakan lokasi Amazon S3 dan peran IAM.

- `IoTSiteWiseDefaultWorkspace` Tidak memiliki batas jumlah sumber daya untuk menyinkronkan AWS IoT SiteWise sumber daya AWS IoT TwinMaker.
- Ketika Anda menyinkronkan sumber daya dari AWS IoT SiteWise, mereka `SyncSource` akan menjadi `SITWISE_MANAGED`. Ini termasuk `Entities` dan `ComponentTypes`.
- AWS IoT SiteWise Fitur baru, seperti hanya `CompositionModel` tersedia di `IoTSiteWiseDefaultWorkspace`.

Ada beberapa batasan khusus untuk `IoTSiteWiseDefaultWorkspace`, yaitu:

- Ruang kerja default tidak dapat dihapus.
- Untuk menghapus sumber daya, Anda harus menghapus AWS IoT SiteWise sumber daya terlebih dahulu, kemudian sumber daya yang sesuai AWS IoT TwinMaker akan dihapus.

Sumber daya disinkronkan dari AWS IoT SiteWise

Topik ini mencantumkan aset mana yang dapat Anda sinkronkan AWS IoT SiteWise ke AWS IoT TwinMaker ruang kerja Anda.

Important

Lihat [Perbedaan antara ruang kerja khusus dan default](#) untuk informasi tentang perbedaan antara ruang kerja kustom dan default.

Ruang kerja khusus dan default

Sumber daya berikut disinkronkan dan tersedia di ruang kerja kustom dan default:

Model Aset

AWS IoT TwinMaker membuat tipe komponen baru untuk setiap model aset di AWS IoT SiteWise.

- Komponen `TypeId` untuk model aset akan menggunakan salah satu pola berikut:
 - Ruang kerja khusus - `iotsitewise.assetmodel:assetModelId`
 - Ruang kerja default - `assetModelId`
- Setiap properti dalam model aset adalah properti baru dalam tipe komponen, dengan salah satu pola penamaan berikut:
 - Ruang kerja khusus - `Property_propertyId`
 - Ruang kerja default - `propertyId`

Nama properti di AWS IoT SiteWise disimpan sebagai `displayName` dalam definisi properti.

- Setiap hierarki dalam model aset adalah properti tipe baru `LIST` dan `nestedType` ada `RELATIONSHIP` di tipe komponen. Hirarki dipetakan ke properti dengan nama yang diawali oleh salah satu dari berikut ini:
 - Ruang kerja khusus - `Hierarchy_hierarchyId`
 - Ruang kerja default - `hierarchyId`

Aset

AWS IoT TwinMaker menciptakan entitas baru untuk setiap aset di AWS IoT SiteWise.

- `entityId` sama dengan `assetId` di AWS IoT SiteWise.
- Entitas ini memiliki satu komponen yang disebut `sitewiseBase`, yang memiliki tipe komponen yang sesuai dengan model aset untuk aset ini.
- Setiap penggantian tingkat aset, seperti menetapkan alias properti atau satuan ukuran, tercermin dalam entitas di AWS IoT TwinMaker

Hanya ruang kerja default

Aset berikut disinkronkan dan hanya tersedia di ruang kerja default,.

`IoTSiteWiseDefaultWorkspace`

`AssetModelComponents`

AWS IoT TwinMaker membuat tipe komponen baru untuk masing-masing `AssetModelComponents` di AWS IoT SiteWise.

- Komponen `TypeId` untuk model aset menggunakan pola berikut: `assetModelId`.
- Setiap properti dalam model aset adalah properti baru dalam tipe komponen, dengan nama properti sebagai `propertyId`. Nama properti di AWS IoT SiteWise disimpan sebagai `displayName` dalam definisi properti.
- Setiap hierarki dalam model aset adalah properti tipe baru `LIST` dan `nestedType` ada `RELATIONSHIP` di tipe komponen. Hirarki dipetakan ke properti dengan nama yang diawali oleh `hierarchyId`

AssetModelCompositeModel

AWS IoT TwinMaker membuat tipe komponen baru untuk masing-masing `AssetModelCompositeModel` di AWS IoT SiteWise.

- Komponen `TypeId` untuk model aset menggunakan pola berikut: `assetModelId_assetModelCompositeModelId`.
- Setiap properti dalam model aset adalah properti baru dalam tipe komponen, dengan nama properti sebagai `propertyId`. Nama properti di AWS IoT SiteWise disimpan sebagai `displayName` dalam definisi properti.

AssetCompositeModels

AWS IoT TwinMaker membuat komponen komposit baru untuk masing-masing `AssetCompositeModel` in AWS IoT SiteWise.

- `componentName` itu sama dengan `assetModelCompositeModelId` di AWS IoT SiteWise.

Sumber daya tidak disinkronkan

Sumber daya berikut tidak disinkronkan:

Aset dan model aset yang tidak disinkronkan

- Model alarm akan disinkronkan sebagai `CompositeModels`, tetapi data terkait dalam aset yang terkait dengan alarm tidak disinkronkan.
- [AWS IoT SiteWise aliran data](#) tidak disinkronkan. Hanya properti yang dimodelkan dalam model aset yang disinkronkan.
- Nilai properti untuk atribut, pengukuran, transformasi, agregat, dan perhitungan metadata seperti rumus dan jendela tidak disinkronkan. Hanya metadata tentang properti, seperti alias, satuan ukuran, dan tipe data yang disinkronkan. Nilai-nilai dapat ditanyakan menggunakan API konektor AWS IoT TwinMaker data biasa, [GetPropertyValueHistory](#).

Gunakan entitas dan tipe komponen yang disinkronkan di AWS IoT TwinMaker

Setelah aset disinkronkan AWS IoT SiteWise, tipe komponen yang disinkronkan hanya akan dibaca. AWS IoT TwinMaker Setiap tindakan pembaruan atau penghapusan harus dilakukan AWS IoT SiteWise, dan perubahan tersebut disinkronkan AWS IoT TwinMaker jika SyncJob masih aktif.

Entitas yang disinkronkan dan komponen AWS IoT SiteWise dasar juga hanya dibaca di AWS IoT TwinMaker. Anda dapat menambahkan komponen tambahan yang tidak disinkronkan ke entitas yang disinkronkan, selama tidak ada atribut tingkat entitas seperti deskripsi atau diperbarui. `entityName`

Beberapa batasan berlaku untuk bagaimana Anda dapat berinteraksi dengan entitas yang disinkronkan. Anda tidak dapat membuat entitas turunan di bawah entitas yang disinkronkan dalam hierarki entitas yang disinkronkan. Selain itu, Anda tidak dapat membuat tipe komponen yang tidak disinkronkan yang diperluas dari tipe komponen yang disinkronkan.

Note

Komponen tambahan akan dihapus bersama dengan entitas jika aset dihapus AWS IoT SiteWise atau jika Anda menghapus pekerjaan sinkronisasi.

Anda dapat menggunakan entitas yang disinkronkan ini di dasbor Grafana dan menambahkannya sebagai tag di komposer adegan seperti entitas biasa. Anda juga dapat mengeluarkan kueri grafik pengetahuan untuk entitas yang disinkronkan ini.

Note

Entitas yang disinkronkan tanpa modifikasi tidak dikenakan biaya, tetapi Anda dikenakan biaya untuk entitas tersebut jika perubahan telah dilakukan. AWS IoT TwinMaker Misalnya, jika Anda menambahkan komponen yang tidak disinkronkan ke entitas yang disinkronkan, entitas tersebut sekarang dikenakan biaya. AWS IoT TwinMaker Untuk informasi selengkapnya, silakan lihat [Harga AWS IoT TwinMaker](#).

Menganalisis status sinkronisasi dan kesalahan

Topik ini memberikan panduan tentang cara menganalisis kesalahan dan status sinkronisasi.

⚠ Important

Lihat [the section called “Perbedaan antara ruang kerja khusus dan default”](#) untuk informasi tentang perbedaan antara ruang kerja kustom dan default.

Sinkronkan status pekerjaan

Pekerjaan sinkronisasi memiliki salah satu status berikut tergantung pada statusnya.

- **CREATING**Status pekerjaan sinkronisasi berarti pekerjaan memeriksa izin dan memuat data AWS IoT SiteWise untuk menyiapkan sinkronisasi.
- **INITIALIZING**Status pekerjaan sinkronisasi berarti semua sumber daya yang ada disinkronkan ke AWS IoT TwinMaker. AWS IoT SiteWise Langkah ini bisa memakan waktu lebih lama untuk diselesaikan jika pengguna memiliki sejumlah besar aset dan model aset AWS IoT SiteWise. Anda dapat memantau jumlah sumber daya yang telah disinkronkan dengan memeriksa pekerjaan sinkronisasi di [AWS IoT TwinMaker konsol](#), atau dengan memanggil `ListSyncResources` API.
- **ACTIVE**Status pekerjaan sinkronisasi berarti langkah inisialisasi selesai. Pekerjaan sekarang siap untuk menyinkronkan pembaruan baru dari AWS IoT SiteWise.
- **ERROR**Status pekerjaan sinkronisasi menunjukkan kesalahan dengan salah satu status sebelumnya. Tinjau pesan kesalahan. Mungkin ada masalah dengan pengaturan peran IAM. Jika Anda ingin menggunakan peran IAM baru, hapus pekerjaan sinkronisasi yang memiliki kesalahan dan buat yang baru dengan peran baru.

Kesalahan sinkronisasi muncul di halaman sumber model, yang diakses dari tabel sumber model Entity di ruang kerja Anda. Halaman sumber model menampilkan daftar sumber daya yang gagal disinkronkan. Sebagian besar kesalahan secara otomatis dicoba ulang oleh pekerjaan sinkronisasi, tetapi jika sumber daya memerlukan tindakan, maka itu tetap dalam **ERROR** keadaan. Anda juga dapat memperoleh daftar kesalahan dengan menggunakan [ListSyncResources](#) API.

Untuk melihat semua kesalahan yang tercantum untuk sumber saat ini, gunakan prosedur berikut.

1. Arahkan ke ruang kerja Anda di [AWS IoT TwinMaker konsol](#).
2. Pilih AWS IoT SiteWise sumber yang tercantum dalam modal Sumber model entitas untuk membuka halaman detail sinkronisasi aset.

The screenshot displays the AWS IoT SiteWise source configuration page. At the top, there is a breadcrumb trail: AWS IoT TwinMaker > Workspaces > SWSync > Source: AWS IoT SiteWise. The main heading is 'AWS IoT SiteWise source' with a 'Disconnect' button in the top right corner.

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. Seperti yang ditunjukkan pada tangkapan layar sebelumnya, sumber daya apa pun dengan kesalahan yang bertahan tercantum dalam tabel Kesalahan. Anda dapat menggunakan tabel ini untuk melacak dan memperbaiki kesalahan yang terkait dengan sumber daya tertentu.

Kemungkinan kesalahan termasuk yang berikut:

- Meskipun AWS IoT SiteWise mendukung nama aset duplikat, AWS IoT TwinMaker hanya mendukungnya di ROOT level, bukan di bawah entitas induk yang sama. Jika Anda memiliki dua aset dengan nama yang sama di bawah entitas induk AWS IoT SiteWise, salah satunya gagal disinkronkan. Untuk memperbaiki kesalahan ini, hapus salah satu aset atau pindahkan aset di bawah aset induk yang berbeda AWS IoT SiteWise sebelum Anda menyinkronkan.
- Jika Anda sudah memiliki entitas dengan ID yang sama dengan ID AWS IoT SiteWise aset, aset tersebut gagal disinkronkan hingga Anda menghapus entitas yang ada.

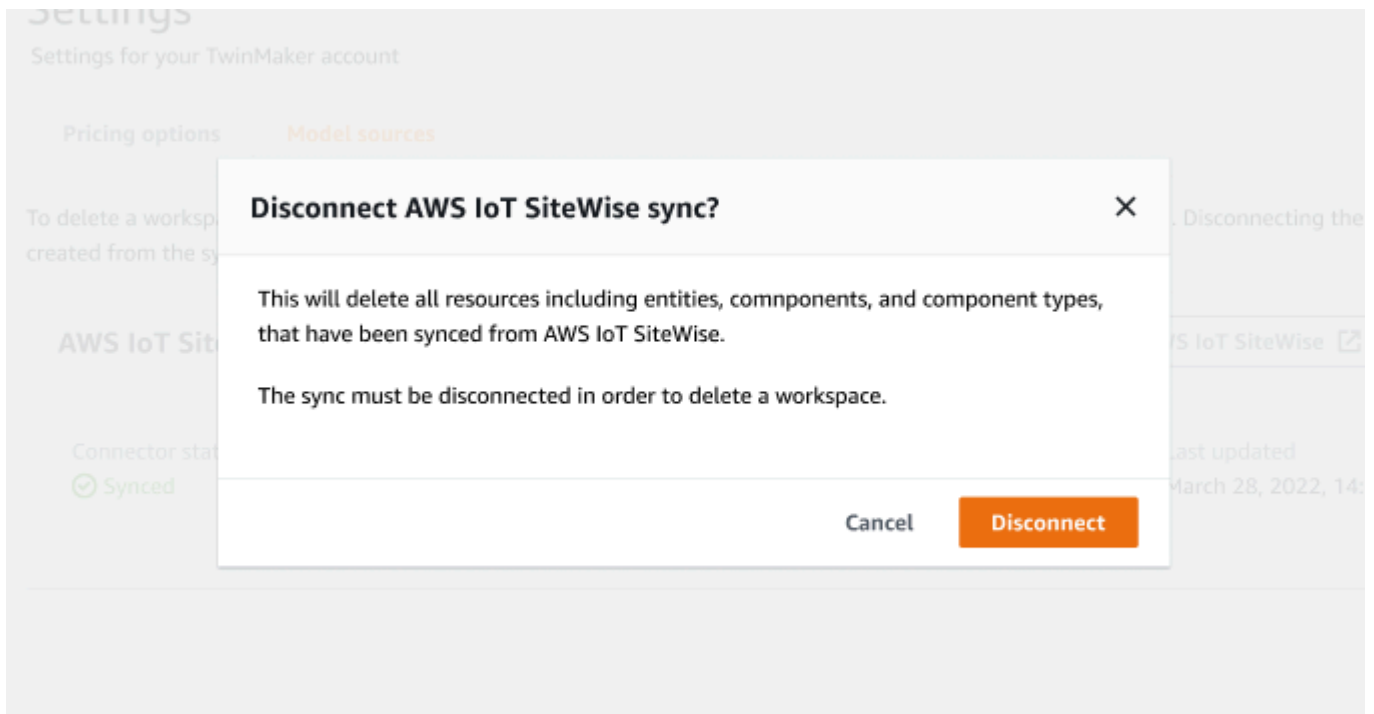
Hapus pekerjaan sinkronisasi

Gunakan prosedur berikut untuk menghapus pekerjaan sinkronisasi.

⚠ Important

Lihat [the section called “Perbedaan antara ruang kerja khusus dan default”](#) untuk informasi tentang perbedaan antara ruang kerja kustom dan default.

1. Navigasikan ke [konsol AWS IoT TwinMaker](#) tersebut.
2. Buka ruang kerja tempat Anda ingin menghapus pekerjaan sinkronisasi.
3. Di bawah Sumber model entitas, pilih AWS IoT SiteWise sumber untuk membuka halaman detail sumber.
4. Untuk menghentikan pekerjaan sinkronisasi, pilih Putuskan sambungan. Konfirmasikan pilihan Anda untuk menghapus sepenuhnya pekerjaan sinkronisasi.



Setelah pekerjaan sinkronisasi dihapus, Anda dapat membuat pekerjaan sinkronisasi lagi di ruang kerja yang sama atau berbeda.

Anda tidak dapat menghapus ruang kerja jika ada pekerjaan sinkronisasi di ruang kerja tersebut. Hapus pekerjaan sinkronisasi terlebih dahulu sebelum menghapus ruang kerja.

Jika ada kesalahan selama penghapusan pekerjaan sinkronisasi, pekerjaan sinkronisasi tetap dalam DELETING status dan secara otomatis dicoba ulang. Anda sekarang dapat secara manual

menghapus salah satu entitas atau tipe komponen yang disinkronkan jika ada kesalahan yang terkait dengan menghapus sumber daya.

Note

Sumber daya apa pun yang disinkronkan AWS IoT SiteWise akan dihapus terlebih dahulu, lalu pekerjaan sinkronisasi itu sendiri dihapus.

Batas sinkronisasi aset

Important

Lihat [the section called “Perbedaan antara ruang kerja khusus dan default”](#) untuk informasi tentang perbedaan antara ruang kerja kustom dan default.

Karena [AWS IoT SiteWise kuota](#) lebih tinggi dari [AWS IoT TwinMaker kuota](#) default, kami meningkatkan batas berikut untuk entitas dan tipe komponen yang disinkronkan dari AWS IoT SiteWise

- 1000 jenis komponen yang disinkronkan di ruang kerja, karena hanya dapat menyinkronkan 1000 model aset dari AWS IoT SiteWise
- 100.000 entitas yang disinkronkan di ruang kerja, karena hanya dapat menyinkronkan 100.000 aset dari AWS IoT SiteWise
- 2000 entitas anak maksimum per entitas induk. Ini menyinkronkan 2000 aset anak per aset induk tunggal.

Note

[GetEntity](#) API hanya menampilkan 50 entitas turunan pertama untuk properti hierarki, tetapi Anda dapat menggunakan [GetPropertyValue](#) API untuk melakukan paginasi dan mengambil daftar semua entitas turunan.

- 600 properti per komponen yang disinkronkan dari AWS IoT SiteWise, yang dapat menyinkronkan model aset dengan 600 total properti dan hierarki.

Note

Batasan ini hanya berlaku untuk entitas yang disinkronkan. Minta peningkatan kuota jika Anda memerlukan peningkatan batas ini untuk sumber daya yang tidak disinkronkan.

AWS IoT TwinMakerIntegrasi dasbor Grafana

AWS IoT TwinMaker mendukung integrasi Grafana melalui plugin aplikasi. Anda dapat menggunakan Grafana versi 8.2.0 dan yang lebih baru untuk berinteraksi dengan aplikasi kembar digital Anda. AWS IoT TwinMakerPlugin ini menyediakan panel khusus, templat dasbor, dan sumber data untuk terhubung ke data kembar digital Anda.

Untuk informasi selengkapnya tentang cara melakukan onboard dengan Grafana dan menyiapkan izin untuk dasbor Anda, lihat topik berikut:

Topik

- [Konfigurasi CORS untuk penampil adegan Grafana](#)
- [Menyiapkan lingkungan Grafana Anda](#)
- [Membuat peran IAM dasbor](#)
- [Membuat kebijakan pemutar AWS IoT TwinMaker video](#)

Note

Anda perlu mengubah konfigurasi CORS (cross-origin resource sharing) bucket Amazon S3 agar antarmuka pengguna Grafana dapat memuat sumber daya dari bucket. Untuk instruksinya, lihat [Konfigurasi CORS untuk penampil adegan Grafana](#).

Untuk informasi selengkapnya tentang plugin AWS IoT TwinMaker Grafana, lihat dokumentasi [AWS IoT TwinMakerAplikasi](#).

Untuk informasi selengkapnya tentang komponen utama plugin Grafana, lihat berikut ini:

- [AWS IoT TwinMakersumber data](#)
- [Templat dasbor](#)
- [Panel Penampil Pemandangan](#)
- [Panel Pemutar Video](#)

Konfigurasi CORS untuk penampil adegan Grafana

Plugin AWS IoT TwinMaker Grafana memerlukan konfigurasi CORS (cross-origin resource sharing), yang memungkinkan antarmuka pengguna Grafana memuat sumber daya dari bucket Amazon S3. Tanpa konfigurasi CORS, Anda akan menerima pesan kesalahan sebagai “Muat Adegan 3D gagal dengan Kegagalan Jaringan” pada penampil Scene karena domain Grafana tidak dapat mengakses sumber daya di bucket Amazon S3.

Untuk mengonfigurasi bucket Amazon S3 Anda dengan CORS, gunakan langkah-langkah berikut:

1. Masuk ke konsol IAM dan buka konsol [Amazon S3](#).
2. Dalam daftar Bucket, pilih nama bucket yang Anda gunakan sebagai bucket sumber daya AWS IoT TwinMaker ruang kerja Anda.
3. Pilih Izin.
4. Di bagian berbagi sumber daya lintas asal, pilih Edit, untuk membuka editor CORS.
5. Di kotak teks editor konfigurasi CORS, ketik atau salin dan tempel konfigurasi JSON CORS berikut dengan mengganti domain ruang kerja Grafana dengan domain Anda. ***GRAFANA-WORKSPACE-DOMAIN***

Note

Anda harus menjaga * karakter tanda bintang di awal elemen "AllowedOrigins": JSON.

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

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

6. Pilih Simpan perubahan untuk menyelesaikan konfigurasi CORS.

Untuk informasi selengkapnya tentang CORS dengan bucket Amazon S3, [lihat Menggunakan berbagi sumber daya lintas asal \(CORS\)](#).

Menyiapkan lingkungan Grafana Anda

Anda dapat menggunakan Grafana Terkelola Amazon untuk layanan yang dikelola sepenuhnya, atau menyiapkan lingkungan Grafana yang Anda kelola sendiri. Dengan Grafana Terkelola Amazon, Anda dapat dengan cepat menerapkan, mengoperasikan, dan menskalakan Grafana open source untuk kebutuhan Anda. Atau, Anda dapat mengatur infrastruktur Anda sendiri untuk mengelola server Grafana.

Untuk informasi selengkapnya tentang kedua opsi lingkungan Grafana, lihat topik berikut:

- [Amazon Managed Grafana](#)
- [Grafana yang dikelola sendiri](#)

Amazon Managed Grafana

Grafana yang Dikelola Amazon menyediakan AWS IoT TwinMaker plugin sehingga Anda dapat dengan cepat berintegrasi dengan AWS IoT TwinMaker Grafana. Karena Grafana yang Dikelola Amazon mengelola server Grafana untuk Anda, Anda dapat memvisualisasikan data Anda tanpa harus membuat, mengemas, atau menerapkan perangkat keras apa pun atau infrastruktur Grafana lainnya. Untuk informasi selengkapnya tentang Grafana yang Dikelola Amazon, lihat [Apa itu Grafana yang Dikelola Amazon?](#) .

Note

Grafana yang Dikelola Amazon saat ini mendukung versi 1.3.1 dari plugin AWS IoT TwinMaker Grafana.

Prasyarat Grafana yang Dikelola Amazon

Untuk digunakan AWS IoT TwinMaker di dasbor Grafana Terkelola Amazon, pertama-tama lengkapi prasyarat berikut:

- Buat AWS IoT TwinMaker ruang kerja. Untuk informasi selengkapnya tentang membuat ruang kerja, lihat [Memulai AWS IoT TwinMaker](#).

Note

Saat pertama kali membuat ruang kerja Grafana Terkelola Amazon di Konsol AWS Manajemen, AWS IoT TwinMaker tidak terdaftar. Namun, plugin sudah diinstal di semua ruang kerja. Anda dapat menemukan AWS IoT TwinMaker plugin di daftar plugin Grafana open source. Anda dapat menemukan sumber AWS IoT TwinMaker data dengan memilih Tambahkan sumber data di halaman Sumber Data.

Saat Anda membuat ruang kerja Grafana Terkelola Amazon, peran IAM dibuat secara otomatis untuk mengelola izin instans Grafana. Ini disebut Workspace IAM Role. Ini adalah opsi penyedia otentikasi yang akan Anda gunakan untuk mengonfigurasi semua sumber AWS IoT TwinMaker data untuk Grafana. Grafana yang Dikelola Amazon tidak mendukung penambahan izin secara otomatis AWS IoT TwinMaker, jadi Anda harus menyiapkan izin ini secara manual. Untuk informasi selengkapnya tentang menyiapkan izin manual, lihat [Membuat peran IAM dasbor](#).

Grafana yang dikelola sendiri

Anda dapat memilih untuk meng-host infrastruktur Anda sendiri untuk menjalankan Grafana. [Untuk informasi tentang menjalankan Grafana secara lokal di mesin Anda, lihat Menginstal Grafana](#). AWS IoT TwinMakerPlugin ini tersedia di katalog Grafana publik. [Untuk informasi tentang menginstal plugin ini di lingkungan Grafana Anda, lihat AWS IoT TwinMaker Aplikasi](#).

Saat Anda menjalankan Grafana secara lokal, Anda tidak dapat dengan mudah berbagi dasbor atau menyediakan akses ke banyak pengguna. [Untuk panduan memulai cepat skrip tentang berbagi dasbor menggunakan Grafana lokal, lihat repositori sampel. AWS IoT TwinMaker](#) Sumber daya ini memandu Anda melalui hosting lingkungan Grafana di Cloud9 dan Amazon EC2 pada titik akhir publik.

Anda harus menentukan penyedia otentikasi mana yang akan Anda gunakan untuk mengonfigurasi sumber data. TwinMaker Anda mengonfigurasi kredensi untuk lingkungan berdasarkan rantai kredensial default (lihat [Menggunakan Rantai Penyedia Kredensial Default](#)). Kredensi default dapat berupa kredensial permanen dari setiap pengguna atau peran. Misalnya, jika Anda menjalankan Grafana di Amazon EC2, rantai kredensial default memiliki akses ke peran [eksekusi Amazon EC2, yang kemudian akan menjadi penyedia otentikasi](#) Anda. Nama Sumber Daya Amazon IAM (ARN) dari penyedia otentikasi diperlukan dalam langkah-langkah untuk [Membuat peran IAM dasbor](#)

Membuat peran IAM dasbor

Dengan AWS IoT TwinMaker, Anda dapat mengontrol akses data di dasbor Grafana Anda. Pengguna dasbor Grafana harus memiliki cakupan izin yang berbeda untuk melihat data, dan dalam beberapa kasus, menulis data. Misalnya, operator alarm mungkin tidak memiliki izin untuk melihat video, sementara admin memiliki izin untuk semua sumber daya. Grafana mendefinisikan izin melalui sumber data, di mana kredensial dan peran IAM disediakan. Sumber AWS IoT TwinMaker data mengambil AWS kredensial dengan izin untuk peran itu. Jika peran IAM tidak disediakan, Grafana menggunakan cakupan kredensialnya, yang tidak dapat dikurangi dengan AWS IoT TwinMaker

Untuk menggunakan AWS IoT TwinMaker dasbor di Grafana, Anda membuat peran IAM dan melampirkan kebijakan. Anda dapat menggunakan templat berikut untuk membantu Anda membuat kebijakan ini.

Buat kebijakan IAM

Buat kebijakan IAM yang disebut *YourWorkspaceId*DashboardPolicy di Konsol IAM. Kebijakan ini memberi ruang kerja Anda akses ke bucket AWS IoT TwinMaker dan sumber daya Amazon S3. Anda juga dapat memutuskan untuk menggunakan [AWS IoT Greengrass Edge Connector untuk Amazon Kinesis Video](#) Streams, yang memerlukan izin untuk Kinesis AWS IoT SiteWise Video Streams dan aset yang dikonfigurasi untuk komponen. Agar sesuai dengan kasus penggunaan Anda, pilih salah satu templat kebijakan berikut.

1. Tidak ada kebijakan izin video

Jika Anda tidak ingin menggunakan [panel Pemutar Video](#) Grafana, buat kebijakan menggunakan templat berikut.

```
{
  "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": "*"
    }
  ]
}
```

Bucket Amazon S3 dibuat untuk setiap ruang kerja. Ini berisi model 3D dan adegan untuk dilihat di dasbor. [SceneViewer](#) Panel memuat item dari ember ini.

2. Kebijakan izin video yang dicakup

Untuk membatasi akses pada panel Pemutar Video di Grafana, kelompokkan sumber daya Konektor AWS IoT Greengrass Edge Anda untuk Amazon Kinesis Video Streams berdasarkan tag. Untuk informasi selengkapnya tentang melingkupi izin untuk sumber daya video Anda, lihat [Membuat kebijakan pemutar AWS IoT TwinMaker video](#)

3. Semua izin video

Jika Anda tidak ingin mengelompokkan video Anda, Anda dapat membuat semuanya dapat diakses dari Pemutar Video Grafana. Siapa pun yang memiliki akses ke ruang kerja Grafana dapat memutar video untuk streaming apa pun di akun Anda, dan hanya membaca akses ke aset apa pun. AWS IoT SiteWise Ini termasuk sumber daya apa pun yang dibuat di masa depan.

Buat kebijakan dengan templat berikut:

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

```

Templat kebijakan ini memberikan izin berikut:

- Akses hanya baca ke bucket S3 untuk memuat adegan.
- Akses hanya baca AWS IoT TwinMaker untuk semua entitas dan komponen di ruang kerja.
- Akses hanya baca untuk streaming semua video Kinesis Video Streams di akun Anda.
- Baca hanya akses ke riwayat nilai properti dari semua AWS IoT SiteWise aset di akun Anda.
- Penyerapan data ke properti apa pun dari AWS IoT SiteWise aset yang ditandai dengan kunci `EdgeConnectorForKVS` dan nilainya `workspaceId`

Menandai AWS IoT SiteWise aset kamera Anda meminta upload video dari edge

Menggunakan Pemutar Video di Grafana, pengguna dapat secara manual meminta agar video diunggah dari cache tepi ke Kinesis Video Streams. Anda dapat mengaktifkan fitur ini untuk AWS IoT SiteWise aset apa pun yang terkait dengan Konektor AWS IoT Greengrass Edge Anda untuk Amazon Kinesis Video Streams dan yang ditandai dengan kunci. `EdgeConnectorForKVS`

Nilai tag dapat berupa daftar `workspaceID` yang dibatasi oleh salah satu karakter berikut: `. : + = @ _ / -`. Misalnya, jika Anda ingin menggunakan AWS IoT SiteWise aset yang terkait dengan Konektor AWS IoT Greengrass Edge untuk Amazon Kinesis Video Streams di seluruh ruang kerja, Anda dapat menggunakan tag yang mengikuti pola ini: `WorkspaceA/WorkspaceB/WorkspaceC`. Plugin Grafana memberlakukan bahwa AWS IoT TwinMaker `workspaceID` digunakan untuk mengelompokkan konsumsi data aset. AWS IoT SiteWise

Tambahkan lebih banyak izin ke kebijakan dasbor Anda

Plugin AWS IoT TwinMaker Grafana menggunakan penyedia otentikasi Anda untuk memanggil `AssumeRole` peran dasbor yang Anda buat. Secara internal, plugin membatasi cakupan izin tertinggi yang dapat Anda akses dengan menggunakan kebijakan sesi dalam panggilan. `AssumeRole` Untuk informasi selengkapnya tentang kebijakan sesi, lihat [Kebijakan sesi](#).

Ini adalah kebijakan permisif maksimum yang dapat Anda miliki di peran dasbor Anda untuk ruang AWS IoT TwinMaker kerja:

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

```

Jika Anda menambahkan pernyataan bahwa Allow lebih banyak izin, mereka tidak akan berfungsi pada AWS IoT TwinMaker plugin. Ini dirancang untuk memastikan izin minimum yang diperlukan digunakan oleh plugin.

Namun, Anda dapat mengurangi izin lebih lanjut. Untuk informasi, lihat [Membuat kebijakan pemutar AWS IoT TwinMaker video](#).

Membuat peran IAM Dasbor Grafana

Di Konsol IAM, buat peran IAM yang disebut. *YourWorkspaceId*DashboardRole Lampirkan *YourWorkspaceId*DashboardPolicy ke peran.

Untuk mengedit kebijakan kepercayaan peran dasbor, Anda harus memberikan izin kepada penyedia otentikasi Grafana untuk memanggil peran AssumeRole dasbor. Perbarui kebijakan kepercayaan dengan templat berikut:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "ARN of Grafana authentication provider"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Untuk informasi selengkapnya tentang membuat lingkungan Grafana dan menemukan penyedia autentikasi Anda, lihat. [Menyiapkan lingkungan Grafana Anda](#)

Membuat kebijakan pemutar AWS IoT TwinMaker video

Berikut ini adalah template kebijakan dengan semua izin video yang Anda butuhkan untuk AWS IoT TwinMaker plugin di 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*"
    }
  }
}
]
}

```

Untuk informasi selengkapnya tentang kebijakan lengkap, lihat Templat kebijakan Semua izin video dalam [Buat kebijakan IAM](#) topik.

Cakupan ke bawah akses ke sumber daya Anda

Panel Pemutar Video di Grafana langsung memanggil Kinesis Video Streams dan SiteWise IoT untuk memberikan pengalaman pemutaran video yang lengkap. Untuk menghindari akses tidak sah ke sumber daya yang tidak terkait dengan AWS IoT TwinMaker ruang kerja Anda, tambahkan kondisi ke kebijakan IAM untuk peran dasbor ruang kerja Anda.

Cakupan ke bawah DAPATKAN izin

Anda dapat mengurangi akses Amazon Kinesis Video AWS IoT SiteWise Streams dan aset dengan menandai sumber daya. Anda mungkin telah menandai aset AWS IoT SiteWise kamera berdasarkan AWS IoT TwinMaker WorkspaceID untuk mengaktifkan fitur permintaan unggah video, lihat topik [Unggah video dari tepi](#). Anda dapat menggunakan pasangan nilai kunci tag yang sama untuk membatasi akses GET ke AWS IoT SiteWise aset, dan juga untuk menandai Kinesis Video Streams Anda dengan cara yang sama.

Anda kemudian dapat menambahkan kondisi ini ke pernyataan kinesisvideo dan iotsitewise di:

*YourWorkspaceId*DashboardPolicy

```

"Condition": {
  "StringLike": {
    "aws:ResourceTag/EdgeConnectorForKVS": "*workspaceId*"
  }
}

```

Kasus penggunaan kehidupan nyata: Mengelompokkan kamera

Dalam skenario ini, Anda memiliki sejumlah besar kamera yang memantau proses memanggang kue di pabrik. Batch adonan kue dibuat di Ruang Adonan, adonan dibekukan di Ruang Freezer, dan kue dipanggang di Ruang Kue. Ada kamera di masing-masing ruangan ini dengan tim operator yang berbeda secara terpisah memantau setiap proses. Anda ingin setiap grup operator diberi wewenang untuk kamar masing-masing. Saat membangun kembar digital untuk pabrik cookie, satu ruang kerja digunakan, tetapi izin kamera harus dicakup oleh kamar.

Anda dapat mencapai pemisahan izin ini dengan menandai grup kamera berdasarkan GroupingID mereka. Dalam skenario ini, GroupingIds adalah BatterRoom, FreezerRoom, dan BakingRoom. Kamera di setiap ruangan terhubung ke Kinesis Video Streams dan harus memiliki tag dengan: EdgeConnectorForKVS Key =, Value =. BatterRoom Nilai dapat berupa daftar pengelompokan yang dibatasi oleh salah satu karakter berikut: . : + = @ _ / -

Untuk mengubah *YourWorkspaceId* DashboardPolicy, gunakan pernyataan kebijakan berikut:

```

...
{
  "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*"
    }
  }
}

```

```

}
},
...

```

Pernyataan ini membatasi pemutaran video streaming dan akses riwayat AWS IoT SiteWise properti ke sumber daya tertentu dalam pengelompokan. *groupingId* ini ditentukan oleh kasus penggunaan Anda. Dalam skenario kami, itu akan menjadi RoomID.

Cakupan AWS IoT SiteWise BatchPutAssetPropertyValue izin ke bawah

Memberikan izin ini mengaktifkan [fitur permintaan unggah video di Pemutar Video](#). Saat Anda mengunggah video, Anda dapat menentukan rentang waktu dan mengirimkan permintaan dengan memilih Kirim di panel di dasbor Grafana.

Untuk memberikan BatchPutAssetPropertyValue izin iotsitewise:, gunakan kebijakan default:

```

...,
{
  "Effect": "Allow",
  "Action": [
    "iotsitewise:BatchPutAssetPropertyValue"
  ],
  "Resource": "*",
  "Condition": {
    "StringLike": {
      "aws:ResourceTag/EdgeConnectorForKVS": "*workspaceId*"
    }
  }
},
...

```

Dengan menggunakan kebijakan ini, pengguna dapat memanggil BatchPutAssetPropertyValue properti apa pun di aset AWS IoT SiteWise kamera. Anda dapat membatasi otorisasi untuk PropertyId tertentu dengan menentukannya dalam kondisi pernyataan.

```

{
  ...
  "Condition": {
    "StringEquals": {
      "iotsitewise:propertyId": "propertyId"
    }
  }
}

```

```
}  
...  
}
```

Panel Pemutar Video di Grafana menyerap data ke dalam properti pengukuran, bernama `VideoUploadRequest`, untuk memulai pengunggahan video dari cache tepi ke Kinesis Video Streams. Temukan `propertyId` dari properti ini di Konsol. AWS IoT SiteWise Untuk mengubah `YourWorkspaceId` `DashboardPolicy`, gunakan pernyataan kebijakan berikut:

```
...,  
{  
  "Effect": "Allow",  
  "Action": [  
    "iotsitewise:BatchPutAssetPropertyValue"  
  ],  
  "Resource": "*",  
  "Condition": {  
    "StringLike": {  
      "aws:ResourceTag/EdgeConnectorForKVS": "*workspaceId"  
    },  
    "StringEquals": {  
      "iotsitewise:propertyId": "VideoUploadRequestPropertyId"  
    }  
  }  
},  
...
```

Pernyataan ini membatasi pengambilan data ke properti tertentu dari aset kamera Anda yang ditandai AWS IoT SiteWise. Untuk informasi selengkapnya, lihat [Cara AWS IoT SiteWise kerja dengan IAM](#).

Hubungkan AWS IoT SiteWise Alarm ke dasbor AWS IoT TwinMaker Grafana

Note

Fitur ini dalam pratinjau publik dan dapat berubah sewaktu-waktu.

AWS IoT TwinMaker mampu mengimpor AWS IoT SiteWise dan alarm Acara ke dalam AWS IoT TwinMaker komponen. Ini memungkinkan Anda untuk dapat menanyakan status alarm dan mengonfigurasi ambang alarm tanpa menerapkan konektor data khusus untuk migrasi AWS IoT SiteWise data. Anda dapat menggunakan plugin AWS IoT TwinMaker Grafana untuk memvisualisasikan status alarm dan mengonfigurasi ambang alarm di Grafana, tanpa melakukan panggilan API ke AWS IoT TwinMaker atau berinteraksi langsung dengan alarm. AWS IoT SiteWise

AWS IoT SiteWise prasyarat konfigurasi alarm

Sebelum membuat alarm dan mengintegrasikannya ke dasbor Grafana Anda, pastikan Anda telah meninjau prasyarat berikut:

- Menjadi akrab dengan AWS IoT SiteWise model dan sistem aset. Untuk informasi selengkapnya, lihat [Membuat model aset](#) dan [Membuat aset](#) di Panduan AWS IoT SiteWise Pengguna.
- Kenali model alarm IoT Events dan cara memasangnya ke AWS IoT SiteWise model. Untuk informasi selengkapnya, lihat [Mendefinisikan alarm AWS IoT Acara](#) di AWS IoT SiteWise Panduan Pengguna.
- Integrasikan AWS IoT TwinMaker dengan Grafana sehingga Anda dapat mengakses AWS IoT TwinMaker sumber daya Anda di Grafana. Untuk informasi lebih lanjut lihat, [AWS IoT TwinMakerIntegrasi dasbor Grafana](#).

Tentukan AWS IoT SiteWise peran IAM komponen alarm

AWS IoT TwinMaker menggunakan peran IAM ruang kerja untuk menanyakan dan mengonfigurasi ambang alarm di Grafana. Izin berikut diperlukan dalam peran AWS IoT TwinMaker ruang kerja, untuk berinteraksi dengan AWS IoT SiteWise alarm di Grafana:

```
{
  "Effect": "Allow",
  "Action": [
    "iotevents:DescribeAlarmModel",
  ],
  "Resource": ["{IoTEventsAlarmModelArn}"]
},{
  "Effect": "Allow",
  "Action": [
    "iotsitewise:BatchPutAssetPropertyValue"
  ],
  "Resource": ["{IoTSitewiseAssetArn}"]
}
```

Di [AWS IoT TwinMaker konsol](#), buat entitas yang mewakili AWS IoT SiteWise aset Anda. Pastikan Anda menambahkan komponen untuk entitas yang menggunakan `com.amazon.iotsitewise.alarm` sebagai tipe komponen, dan pilih model aset dan alarm yang sesuai.

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

Screenshot di atas adalah contoh membuat entitas ini dengan tipe `com.amazon.iotsitewise.alarm`.

Saat Anda membuat komponen ini, AWS IoT TwinMaker secara otomatis mengimpor properti alarm terkait dari AWS IoT SiteWise dan AWS IoT Events. Anda dapat mengulangi pola jenis komponen alarm ini untuk membuat komponen alarm untuk semua aset yang dibutuhkan di ruang kerja Anda.

Kueri dan perbarui melalui AWS IoT TwinMaker API

Setelah membuat komponen alarm, Anda dapat menanyakan status alarm, ambang batas, dan memperbarui ambang alarm melalui API. AWS IoT TwinMaker

Di bawah ini adalah contoh permintaan untuk menanyakan status alarm:

```
aws iottwinmaker get-property-value-history --cli-input-json \  
{  
  "workspaceId": "{workspaceId}",  
  "entityId": "{entityId}",  
  "componentName": "{componentName}",  
  "selectedProperties": ["alarm_status"],  
  "startTime": "{startTimeIsoString}",  
  "endTime": "{endTimeIsoString}"  
}
```

Di bawah ini adalah contoh permintaan untuk menanyakan ambang alarm.

```
aws iottwinmaker get-property-value-history --cli-input-json \  
{  
  "workspaceId": "{workspaceId}",  
  "entityId": "{entityId}",  
  "componentName": "{componentName}",  
  "selectedProperties": ["alarm_threshold"],  
  "startTime": "{startTimeIsoString}",  
  "endTime": "{endTimeIsoString}"  
}
```

Di bawah ini adalah contoh permintaan untuk memperbarui ambang alarm:

```
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}"
        }
    ]
}
]
}'

```

Konfigurasi dasbor Grafana Anda untuk alarm

Peran IAM dasbor yang diaktifkan penulisan kedua perlu dibuat, itu adalah peran normal tetapi dengan izin agar tindakan ditambahkan `iottwinmaker:BatchPutPropertyValues` ke TwinMaker ruang kerja seperti pada contoh di bawah ini.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iottwinmaker:Get*",
        "iottwinmaker:List*",
        "iottwinmaker:BatchPutPropertyValues"
      ],
      "Resource": [
        "{workspaceArn}",
        "{workspaceArn}/*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": "iottwinmaker:ListWorkspaces",
      "Resource": "*"
    }
  ]
}

```

```
}
```

Atau Anda dapat menambahkan pernyataan ini di akhir peran IAM Anda sebagai gantinya:

```
{  
  "Effect": "Allow",  
  "Action": [  
    "iottwinmaker:BatchPutPropertyValues"  
  ],  
  "Resource": [  
    "{workspaceArn}",  
    "{workspaceArn}/*"  
  ]  
}
```

Sumber data harus disetel arn tulis dengan peran tulis dasbor yang Anda buat.

Setelah Anda mengubah peran IAM Anda, masuk ke dasbor Grafana Anda untuk mengambil peran yang diperbarui arn. Pilih kotak centang untuk Tentukan izin tulis untuk Panel Konfigurasi Alarm dan salin di arn untuk peran Tulis.

Settings Dashboards

Name Default

Connection Details

Authentication Provider	<input type="text" value="AWS SDK Default"/>
Assume Role ARN	<input type="text" value="arn:aws:iam:*"/>
External ID	<input type="text" value="External ID"/>
Endpoint	<input type="text" value="https://{service}.{region}.amazonaws.com"/>
Default Region	<input type="text" value="us-east-1"/>

Assume Role ARN

Specify an IAM role to narrow the permission scope of this datasource. Follow the documentation [here](#) to create policies and a role with minimal permissions for your TwinMaker workspace.

TwinMaker settings

Workspace	<input type="text" value="enter workspace ID"/>
<input checked="" type="checkbox"/> Define write permissions for Alarm Configuration Panel	
Assume Role ARN Write	<input type="text" value="arn:aws:iam:*"/>

Gunakan dasbor Grafana untuk visualisasi alarm

Gunakan prosedur berikut untuk menambahkan panel konfigurasi alarm ke dasbor Anda dan konfigurasi:

1. Pilih ruang kerja di opsi panel.

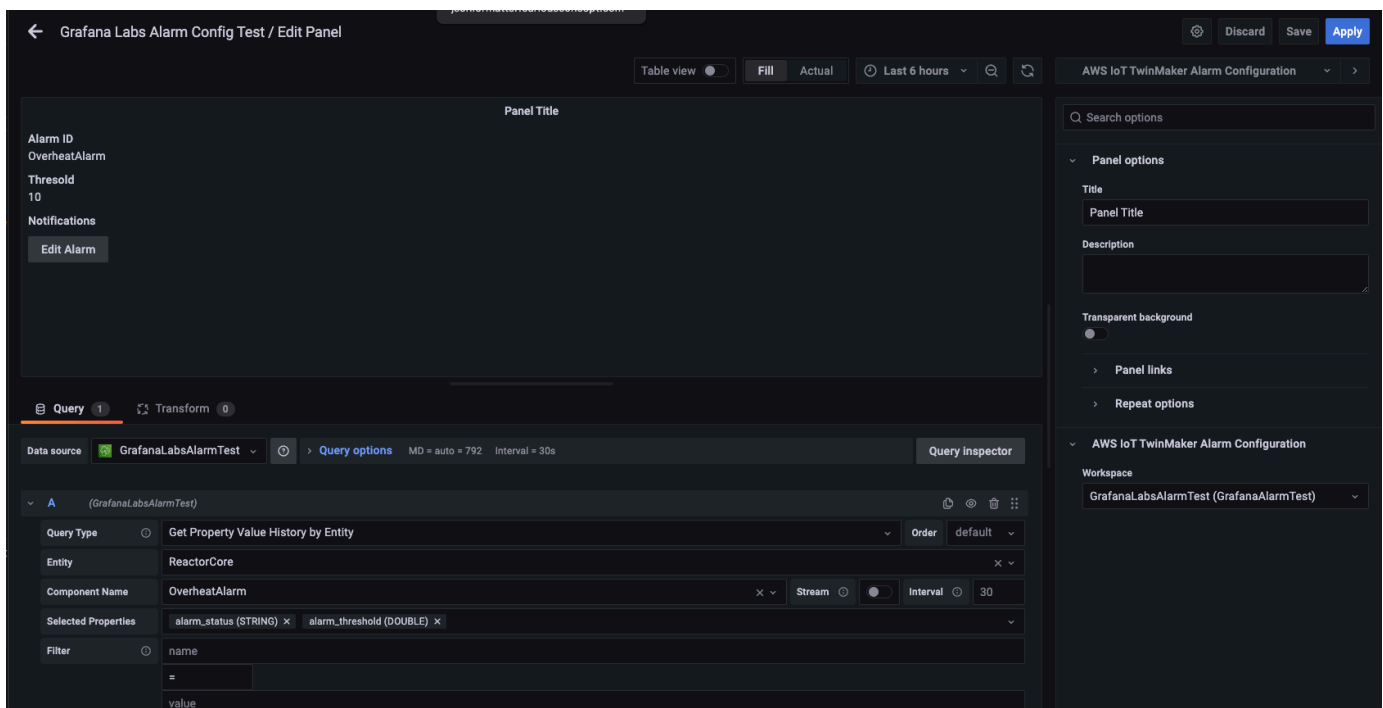
2. Tetapkan sumber data Anda dalam konfigurasi kueri.
3. Gunakan jenis kueri berikut: `Get Property Value History by Entity`.
4. Pilih entitas atau variabel entitas, Anda ingin menambahkan alarm ke.
5. Setelah Anda memilih entitas, pilih komponen atau variabel komponen, untuk menerapkan properti ke.
6. Untuk properti, pilih: `alarm_status` dan `alarm_threshold`.

Ketika terhubung, Anda akan melihat Id untuk Id alarm dan ambang batas saat ini.

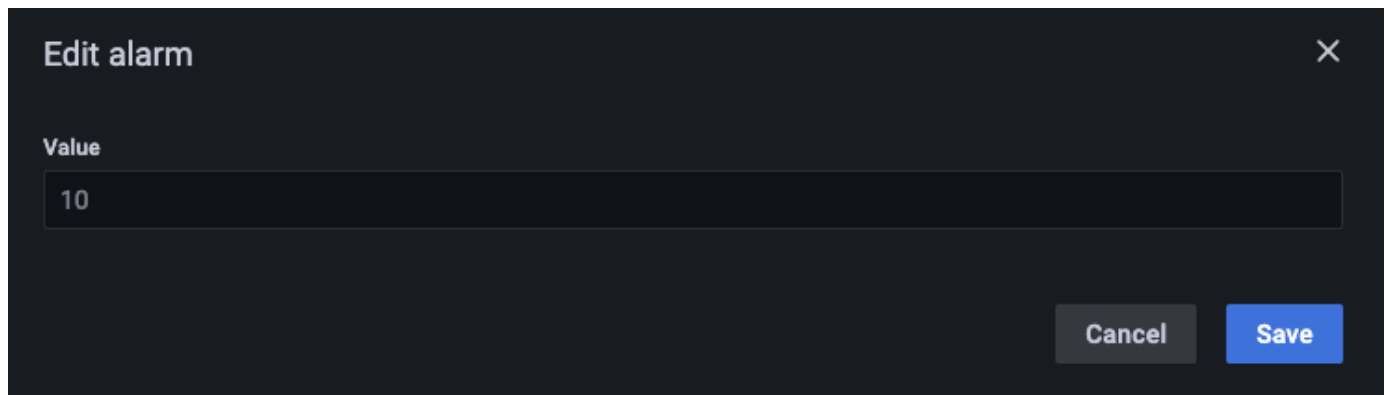
Note

Untuk pratinjau publik, tidak ada pemberitahuan yang ditampilkan. Anda harus meninjau status alarm dan ambang batas untuk memastikan properti diterapkan dengan benar.

7. Urutan Kueri default dari Ascending harus digunakan sehingga nilai terbaru ditampilkan.
8. Bagian filter dari Query dapat dibiarkan kosong. Konfigurasi lengkap digambarkan di bawah ini:



9. Dengan menggunakan tombol Edit Alarm Anda dapat memunculkan dialog untuk mengubah ambang alarm saat ini.
10. Pilih Simpan untuk menetapkan nilai ambang baru.



Edit alarm ✕

Value

10

Cancel Save

Note

Panel ini hanya boleh digunakan dengan rentang waktu langsung yang mencakup masa kini. Menggunakannya dengan rentang waktu yang berakhir dan dimulai di masa lalu dapat menunjukkan nilai yang tidak terduga saat mengedit ambang alarm sebagai ambang batas saat ini selalu.

AWS IoT TwinMaker Integrasi Matterport

Matterport menyediakan berbagai opsi pengambilan untuk memindai lingkungan dunia nyata dan membuat model 3D imersif, juga dikenal sebagai kembar digital Matterport. Model-model ini disebut ruang Matterport. AWS IoT TwinMaker mendukung integrasi Matterport, memungkinkan Anda untuk mengimpor kembar digital Matterport Anda ke dalam adegan Anda. AWS IoT TwinMaker Dengan memasang kembar digital Matterport dengan AWS IoT TwinMaker, Anda dapat memvisualisasikan dan memantau sistem kembar digital Anda di lingkungan virtual.



[Untuk informasi lebih lanjut tentang penggunaan Matterport, baca dokumentasi Matterport dan halaman AWS IoT TwinMaker Matterport.](#)

Topik integrasi

- [Ikhtisar integrasi](#)
- [Prasyarat integrasi Matterport](#)
- [Hasilkan dan catat kredensi Matterport Anda](#)
- [Simpan kredensi Matterport Anda di AWS Secrets Manager](#)
- [Impor ruang Matterport ke dalam adegan AWS IoT TwinMaker](#)
- [Gunakan spasi Matterport di dasbor AWS IoT TwinMaker Grafana Anda](#)

- [Gunakan spasi Matterport di aplikasi web Anda AWS IoT TwinMaker](#)

Ikhtisar integrasi

Integrasi ini memungkinkan Anda untuk melakukan hal berikut:

- Gunakan tag dan spasi Matterport Anda di AWS IoT TwinMaker app kit.
- Lihat data matterport yang Anda impor di dasbor Grafana AWS IoT TwinMaker Anda. Untuk informasi lebih lanjut tentang penggunaan AWS IoT TwinMaker dan Grafana, baca dokumentasi integrasi dasbor [Grafana](#).
- Impor ruang Matterport Anda ke dalam adegan Anda AWS IoT TwinMaker .
- Pilih dan impor tag Matterport yang ingin Anda ikat ke data di adegan Anda. AWS IoT TwinMaker
- Secara otomatis memunculkan ruang Matterport Anda dan menandai perubahan dalam AWS IoT TwinMaker adegan Anda dan menyetujui mana yang akan disinkronkan.

Proses integrasi terdiri dari 3 langkah penting.

1. [Hasilkan dan catat kredensi Matterport Anda](#)
2. [Simpan kredensi Matterport Anda di AWS Secrets Manager](#)
3. [Impor ruang Matterport ke dalam adegan AWS IoT TwinMaker](#)

Anda memulai integrasi Anda di [AWS IoT TwinMaker konsol](#). Di halaman Pengaturan konsol, di bawah sumber daya pihak ke-3, buka integrasi Matterport untuk menavigasi antara berbagai sumber daya yang diperlukan untuk integrasi.

The screenshot shows the AWS IoT TwinMaker console interface. The top navigation bar includes the AWS logo, 'Services', 'Resource Groups', 'Customer Account', 'N. Virginia', and 'Support'. The left sidebar contains navigation options like 'How it works', 'Workspaces', '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, there is a section for '3rd party software integration' with a sub-section for 'Matterport integration (1)'. This section contains a 'How it works' guide with four steps: 1. Contact Matterport, 2. Record your Matterport SDK credentials, 3. Add your Matterport credentials into AWS secrets manager, and 4. Select your Matterport account in a scene composer scene. Below the steps is a 'Connected accounts' table which is currently empty, showing 'No connections'. A button for 'AWS Secret Manager' is visible below the table.

Prasyarat integrasi Matterport

Sebelum mengintegrasikan Matterport dengan AWS IoT TwinMaker pastikan Anda memenuhi prasyarat berikut:

- Anda telah membeli akun Matterport tingkat Perusahaan dan [produk Matterport](#) yang diperlukan untuk integrasi. AWS IoT TwinMaker
- Anda memiliki AWS IoT TwinMaker ruang kerja. Untuk informasi selengkapnya, lihat [Memulai dengan AWS IoT TwinMaker](#).

- Anda telah memperbarui peran AWS IoT TwinMaker ruang kerja Anda. Untuk informasi selengkapnya tentang membuat peran ruang kerja, lihat [Membuat dan mengelola peran layanan AWS IoT TwinMaker](#)

Tambahkan yang berikut ini ke peran ruang kerja Anda:

```
{
  "Effect": "Allow",
  "Action": "secretsmanager:GetSecretValue",
  "Resource": [
    "AWS Secrets Manager secret ARN"
  ]
}
```

- Anda harus menghubungi Matterport untuk mengonfigurasi lisensi yang diperlukan untuk mengaktifkan integrasi. Matterport juga akan mengaktifkan Private Model Embed (PME) untuk integrasi.

Jika Anda sudah memiliki manajer akun Matterport, hubungi mereka secara langsung.

Gunakan prosedur berikut untuk menghubungi Matterport dan meminta integrasi jika Anda tidak memiliki titik kontak Matterport:

1. Buka [Matterport dan AWS IoT TwinMaker halaman](#).
2. Tekan tombol Hubungi kami, untuk membuka formulir kontak.
3. Isi informasi yang diperlukan pada formulir.
4. Saat Anda siap, pilih SAY HELLO untuk mengirim permintaan Anda ke Matterport.

Setelah Anda meminta integrasi, Anda dapat menghasilkan kredensial Matterport SDK dan Private Model Embed (PME) yang diperlukan untuk melanjutkan proses integrasi.

Note

Ini mungkin melibatkan Anda dikenakan biaya untuk membeli produk atau layanan baru.

Hasilkan dan catat kredensi Matterport Anda

Untuk mengintegrasikan Matterport dengan AWS IoT TwinMaker, Anda harus memberikan kredensi AWS Secrets Manager Matterport. Gunakan prosedur berikut untuk menghasilkan kredensial Matterport SDK.

1. Masuk ke akun [Matterport](#) Anda.
2. Arahkan ke halaman pengaturan akun Anda.
3. Setelah di halaman pengaturan, pilih opsi Alat pengembang.
4. Pada halaman Alat pengembang, buka bagian Manajemen Kunci SDK.
5. Setelah berada di bagian Manajemen Kunci SDK, pilih opsi untuk menambahkan kunci SDK baru.
6. Setelah Anda memiliki kunci Matterport SDK, tambahkan domain ke kunci untuk dan server Grafana AWS IoT TwinMaker Anda. Jika Anda menggunakan kit AWS IoT TwinMaker aplikasi, pastikan untuk menambahkan domain khusus Anda juga.
7. Selanjutnya, temukan bagian Manajemen Integrasi Aplikasi, Anda akan melihat aplikasi PME Anda terdaftar. Catat informasi berikut:
 - ID Klien
 - Rahasia Klien

Note

Karena Rahasia Klien hanya disajikan kepada Anda sekali, kami sangat menyarankan agar Anda mencatat Rahasia Klien Anda. Anda harus menunjukkan Rahasia Klien Anda di AWS Secrets Manager konsol untuk melanjutkan integrasi Matterport.

Kredensi ini secara otomatis dibuat ketika Anda telah membeli komponen yang diperlukan dan PME untuk akun Anda telah diaktifkan oleh Matterport. Jika kredensi ini tidak muncul, hubungi Matterport. Untuk meminta kontak, lihat [Matterport dan formulir AWS IoT TwinMaker](#) kontak.

[Untuk informasi selengkapnya tentang kredensial Matterport SDK, lihat Ikhtisar Dokumen SDK dokumentasi SDK resmi Matterport.](#)

Simpan kredensi Matterport Anda di AWS Secrets Manager

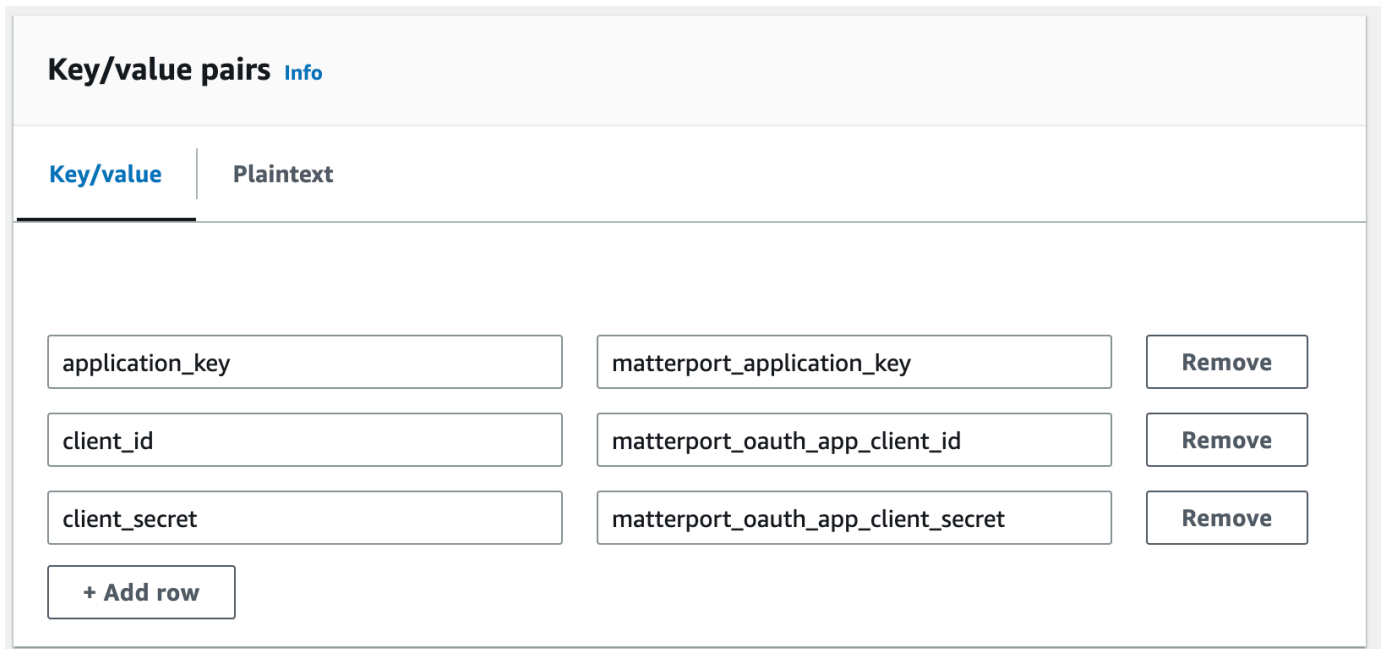
Gunakan prosedur berikut untuk menyimpan kredensial Matterport Anda di AWS Secrets Manager

Note

Anda memerlukan Rahasia Klien yang dibuat dari prosedur dalam [Hasilkan dan catat kredensi Matterport Anda](#) topik untuk melanjutkan integrasi Matterport.

1. Masuk ke AWS Secrets Manager konsol.
2. Arahkan ke halaman Rahasia dan pilih Simpan rahasia baru.
3. Untuk tipe Rahasia, pilih Jenis rahasia lainnya.
4. Di bagian pasangan kunci/nilai, tambahkan pasangan kunci-nilai berikut, dengan kredensial Matterport Anda sebagai nilainya:
 - <your Matterport credentials>Buat pasangan kunci-nilai, dengan Key:application_key, dan Value:.
 - <your Matterport credentials>Buat pasangan kunci-nilai, dengan Key:client_id, dan Value:.
 - <your Matterport credentials>Buat pasangan kunci-nilai, dengan Key:client_secret, dan Value:.

Setelah selesai, Anda harus memiliki konfigurasi yang mirip dengan contoh berikut:



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
+ Add row		

5. Untuk kunci Enkripsi, Anda dapat membiarkan kunci enkripsi default `aws/secretsmanager` dipilih.
6. Pilih Berikutnya untuk melanjutkan ke halaman Konfigurasi rahasia.
7. Isi kolom untuk nama Rahasia dan Deskripsi.
8. Tambahkan tag ke rahasia ini di bagian Tag.

Saat membuat tag, tetapkan kunci `AWSIoTtwinMaker_Matterport` seperti yang ditunjukkan pada gambar berikut:

AWS Secrets Manager > Secrets > Store a new secret

Step 1
Choose secret type

Step 2
Configure secret

Step 3
Configure rotation - optional

Step 4
Review

Configure secret

Secret name and description [Info](#)

Secret name
A descriptive name that helps you find your secret later.

Secret name must contain only alphanumeric characters and the characters /_+@-

Description - optional

Maximum 250 characters.

Tags - optional

Key	Value - optional	
<input type="text" value="AWSIoTtwinMaker_Matterport"/>	<input type="text" value="Enter value"/>	<input type="button" value="Remove"/>
<input type="button" value="Add"/>		

Note

Anda harus menambahkan tag. Tag diperlukan saat menambahkan rahasia pihak ke-3 ke dalam AWS Secrets Manager, meskipun Tag terdaftar sebagai opsional.

Bidang Nilai adalah opsional. Setelah Anda memberikan Kunci, Anda dapat memilih Tambah untuk melanjutkan ke langkah berikutnya.

- Pilih Berikutnya untuk melanjutkan ke halaman Konfigurasi rotasi. Menyiapkan rotasi rahasia adalah opsional. Jika Anda ingin menyelesaikan menambahkan rahasia Anda dan tidak perlu rotasi, pilih Berikutnya lagi. Untuk informasi selengkapnya tentang rotasi rahasia, lihat [Memutar AWS Secrets Manager rahasia](#).
- Konfirmasikan konfigurasi rahasia Anda di halaman Tinjauan. Setelah Anda siap untuk menambahkan rahasia Anda, pilih Store.

Untuk informasi selengkapnya tentang penggunaan AWS Secrets Manager, lihat AWS Secrets Manager dokumentasi berikut:

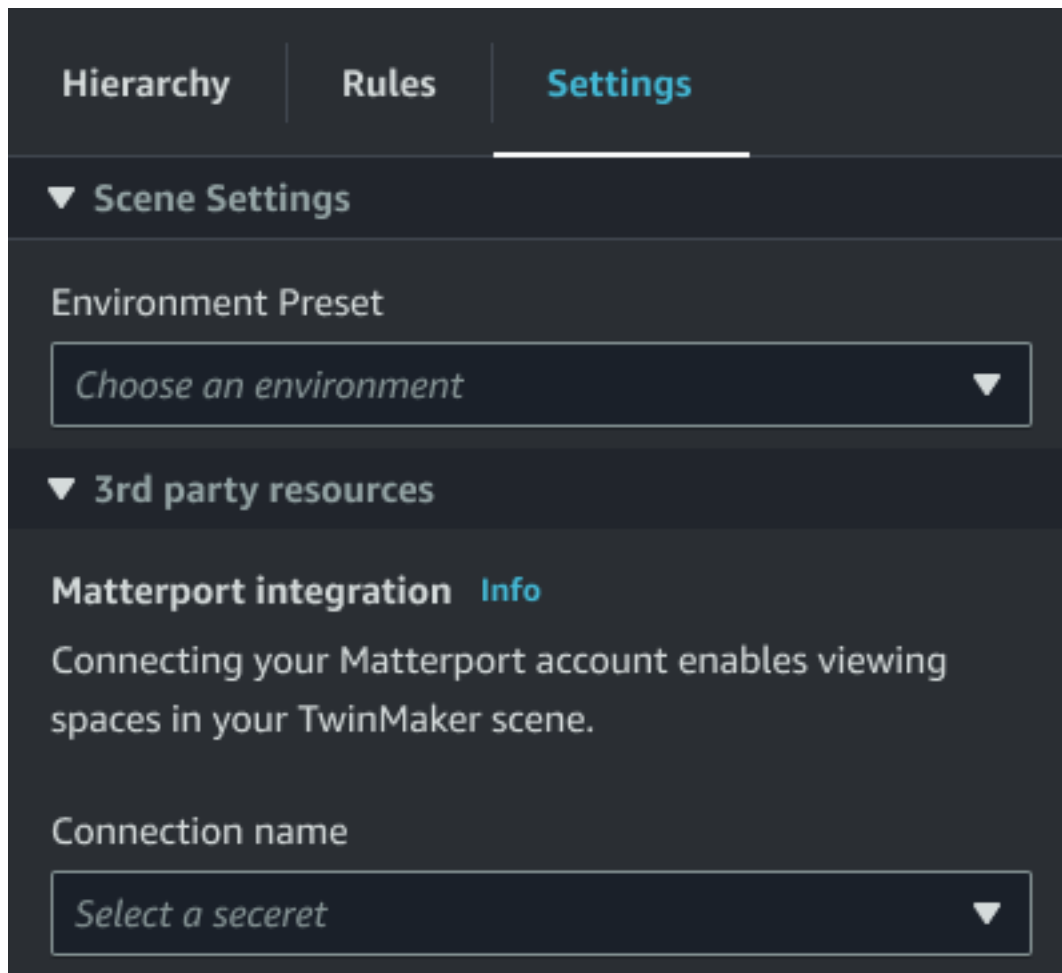
- [Buat dan kelola rahasia dengan AWS Secrets Manager](#)
- [Apa itu AWS Secrets Manager?](#)
- [Putar AWS Secrets Manager rahasia](#)

Sekarang Anda siap untuk mengimpor aset Matterport Anda ke dalam AWS IoT TwinMaker adegan. Lihat prosedur di bagian berikut, [Impor ruang Matterport ke dalam adegan AWS IoT TwinMaker](#)

Impor ruang Matterport ke dalam adegan AWS IoT TwinMaker

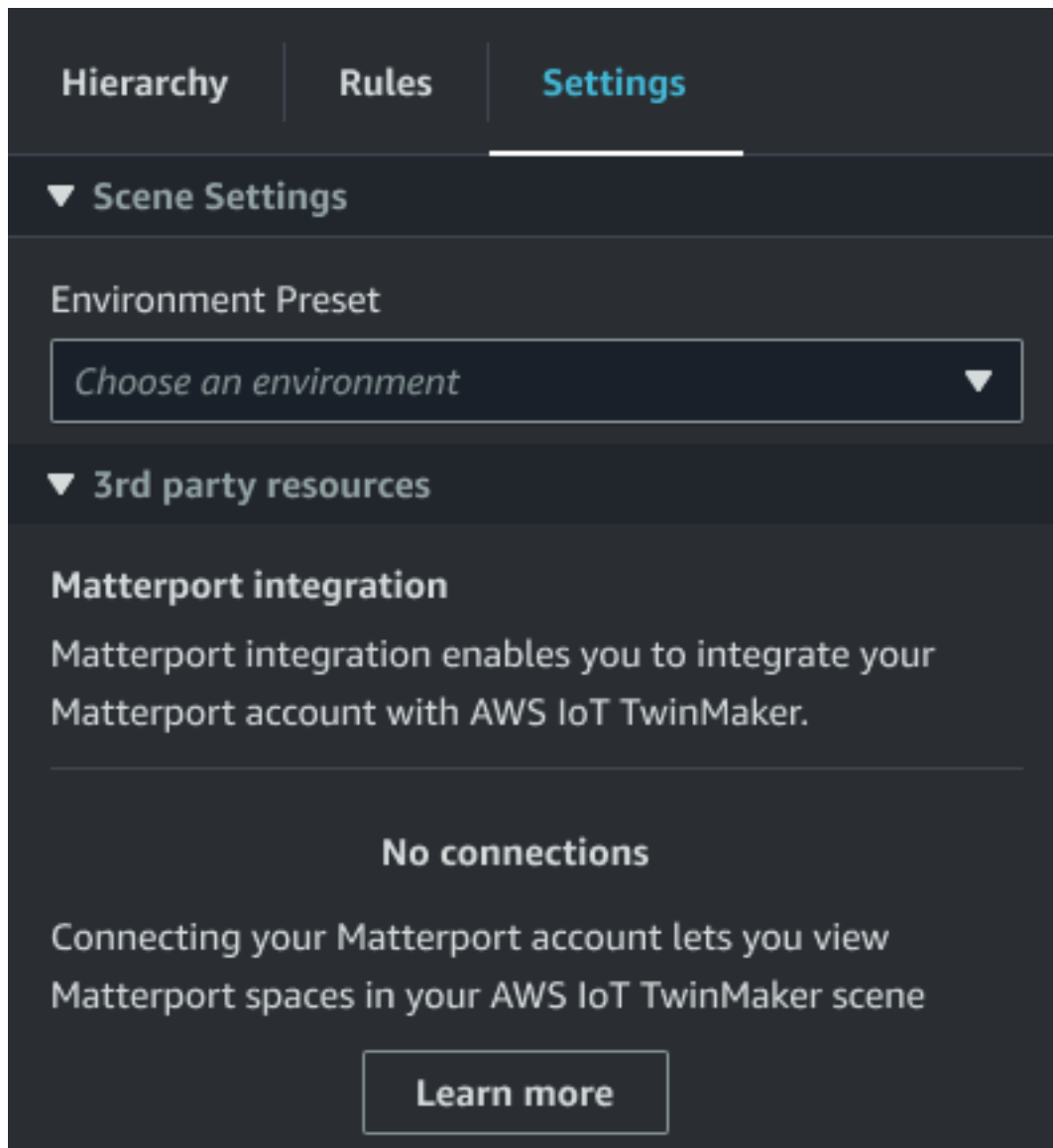
Tambahkan pemindaian Matterport ke adegan Anda dengan memilih akun Matterport yang terhubung dari dalam halaman pengaturan adegan. Gunakan prosedur berikut untuk mengimpor pemindaian dan tag Matterport Anda:

1. Masuk ke [AWS IoT TwinMaker konsol](#).
2. Buat atau buka AWS IoT TwinMaker adegan yang ada di mana Anda ingin menggunakan ruang Matterport.
3. Setelah adegan terbuka, navigasikan ke tab Pengaturan.
4. Di Pengaturan, di bawah sumber daya pihak ke-3, temukan nama Koneksi dan masukkan rahasia yang Anda buat dalam prosedur dari [Simpan kredensi Matterport Anda di AWS Secrets Manager](#).

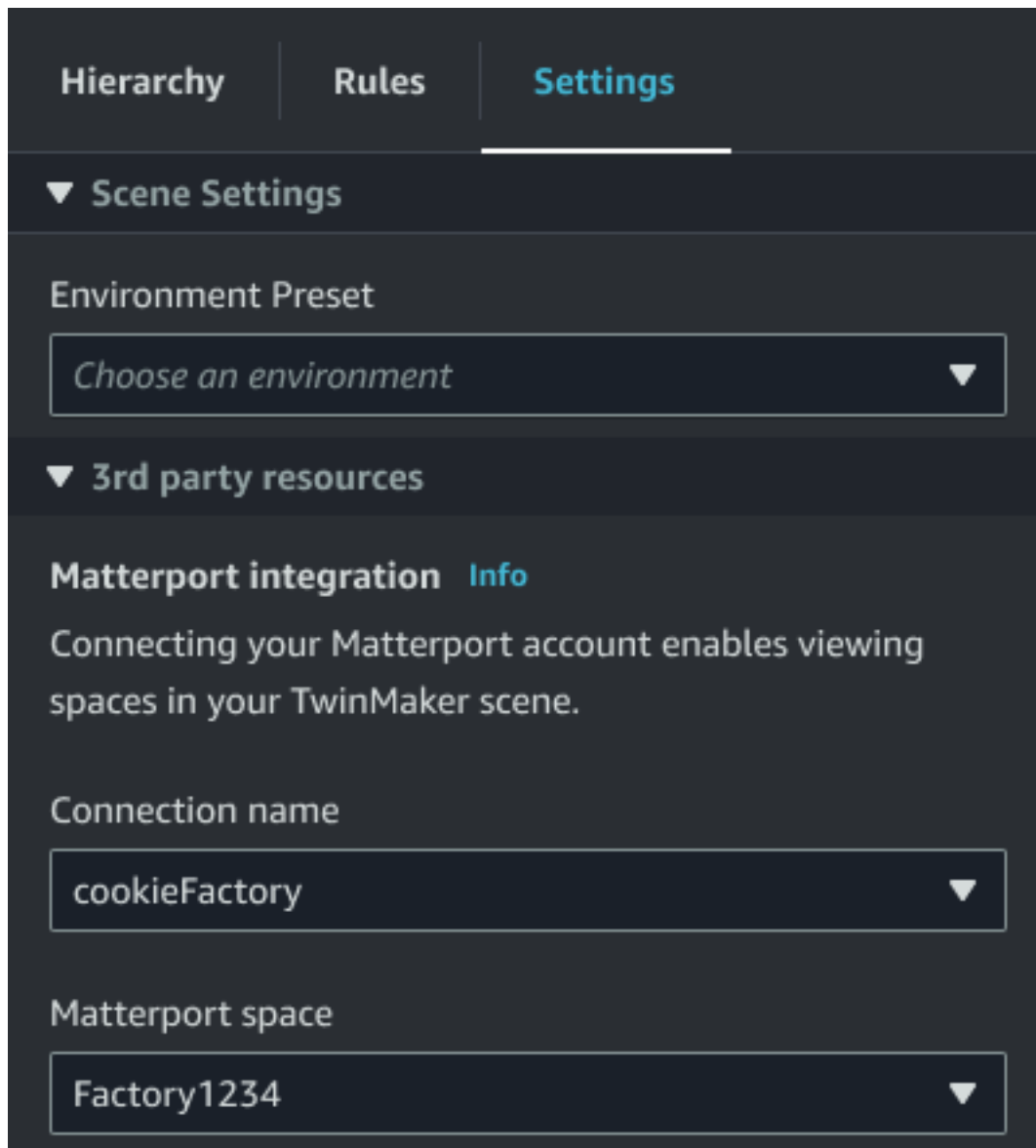


Note

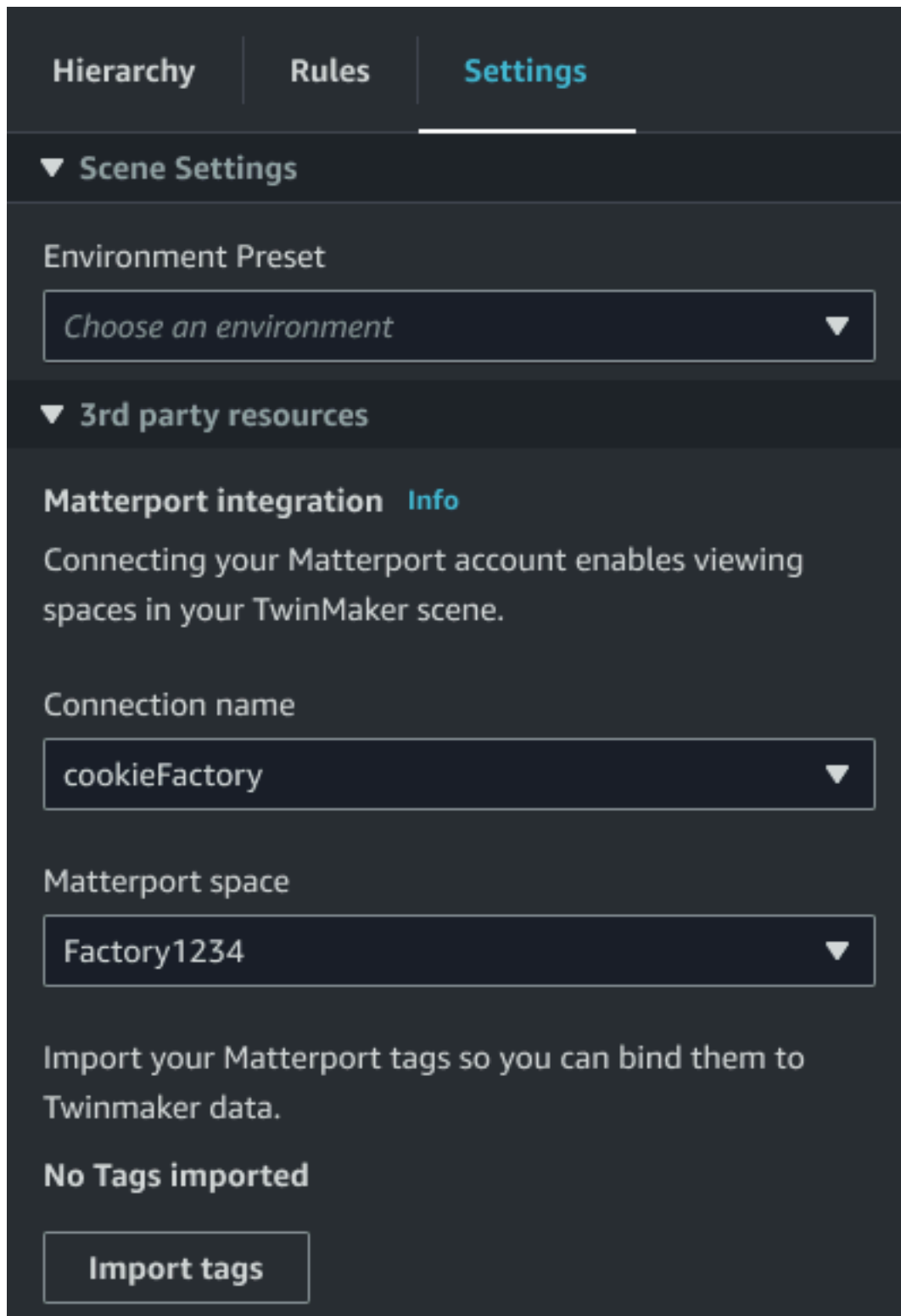
Jika Anda melihat pesan yang menyatakan Tidak ada koneksi, navigasikan ke halaman pengaturan [AWS IoT TwinMaker konsol](#) untuk memulai proses integrasi Matterport.



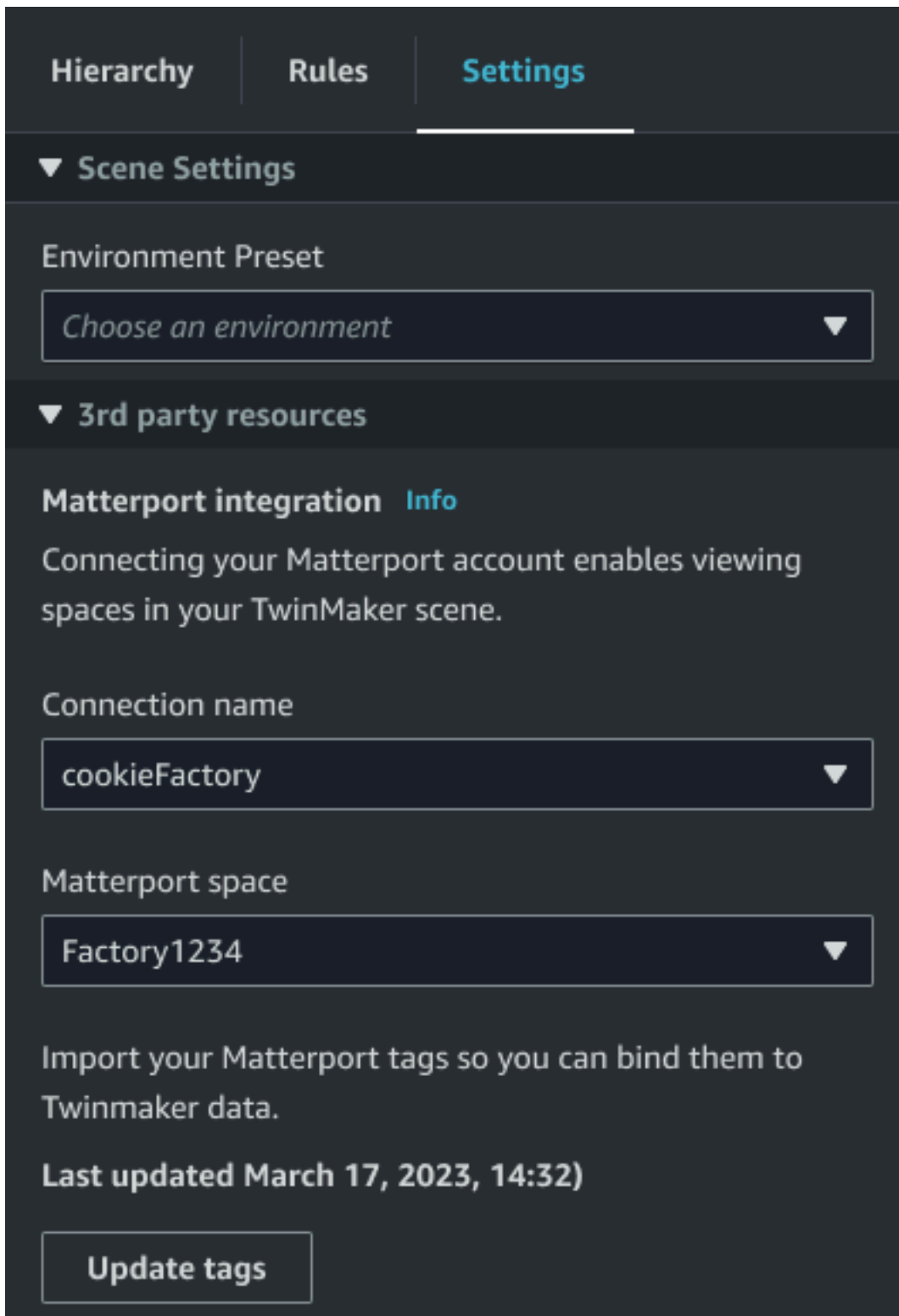
5. Selanjutnya, pilih ruang Matterport yang ingin Anda gunakan dalam adegan Anda dengan memilihnya di drop-down ruang Matterport.



6. Setelah memilih spasi, Anda dapat mengimpor tag Matterport Anda dan mengubahnya menjadi tag AWS IoT TwinMaker adegan dengan menekan tombol Impor tag.



Setelah Anda mengimpor tag Matterport, tombol diganti dengan tombol Perbarui tag. Anda dapat terus memperbarui tag Matterport Anda AWS IoT TwinMaker sehingga selalu mencerminkan perubahan terbaru di akun Matterport Anda.



7. Anda telah berhasil terintegrasi AWS IoT TwinMaker dengan Matterport, dan sekarang AWS IoT TwinMaker adegan Anda memiliki ruang dan tag Matterport yang diimpor. Anda dapat bekerja dalam adegan ini seperti yang Anda lakukan dengan AWS IoT TwinMaker adegan lainnya.

Untuk informasi selengkapnya tentang bekerja dengan AWS IoT TwinMaker adegan, lihat [Membuat dan mengedit AWS IoT TwinMaker adegan](#).

Gunakan spasi Matterport di dasbor AWS IoT TwinMaker Grafana Anda

Setelah Anda mengimpor ruang Matterport Anda ke dalam sebuah AWS IoT TwinMaker adegan, Anda dapat melihat pemandangan itu dengan ruang Matterport di dasbor Grafana Anda. Jika Anda telah mengonfigurasi Grafana dengan AWS IoT TwinMaker, maka Anda cukup membuka dasbor Grafana untuk melihat pemandangan Anda dengan ruang Matterport yang diimpor.

Jika Anda belum mengonfigurasi AWS IoT TwinMaker dengan Grafana, selesaikan proses integrasi Grafana terlebih dahulu. Anda memiliki dua pilihan saat berintegrasi AWS IoT TwinMaker dengan Grafana. Anda dapat menggunakan instans Grafana yang dikelola sendiri atau Anda dapat menggunakan Grafana Terkelola Amazon.

Lihat dokumentasi berikut untuk mempelajari lebih lanjut tentang opsi Grafana dan proses integrasi:

- [AWS IoT TwinMaker Integrasi dasbor Grafana](#).
- [Grafana yang Dikelola Amazon](#).
- [Grafana yang dikelola sendiri](#).

Gunakan spasi Matterport di aplikasi web Anda AWS IoT TwinMaker

Setelah mengimpor ruang Matterport ke dalam sebuah AWS IoT TwinMaker adegan, Anda dapat melihat adegan itu dengan ruang Matterport di aplikasi web kit AWS IoT aplikasi Anda.

Lihat dokumentasi berikut untuk mempelajari lebih lanjut tentang penggunaan kit AWS IoT aplikasi:

- Untuk mempelajari lebih lanjut tentang menggunakan AWS IoT TwinMaker kit AWS IoT aplikasi, lihat [Buat aplikasi web yang disesuaikan menggunakan Komponen AWS IoT TwinMaker UI](#).
- Untuk mempelajari lebih lanjut tentang menggunakan kit AWS IoT aplikasi, silakan kunjungi halaman [AWS IoT Application kit Github](#).
- Untuk petunjuk tentang cara memulai aplikasi web baru menggunakan kit AWS IoT aplikasi, silakan kunjungi halaman dokumentasi [Kit Aplikasi IoT](#) resmi.

AWS IoT TwinMaker Integrasi video

Kamera video menghadirkan peluang bagus untuk simulasi kembar digital. Anda dapat menggunakan AWS IoT TwinMaker untuk mensimulasikan lokasi kamera dan kondisi fisik Anda. Buat entitas AWS IoT TwinMaker untuk kamera di tempat Anda, dan gunakan komponen video untuk melakukan streaming video langsung dan metadata dari situs Anda ke AWS IoT TwinMaker adegan Anda atau ke dasbor Grafana.

AWS IoT TwinMaker dapat merekam video dari perangkat tepi dengan dua cara. Anda dapat melakukan streaming video dari perangkat tepi dengan konektor tepi untuk aliran video Kinesis, atau Anda dapat menyimpan video di perangkat tepi dan memulai pengunggahan video dengan pesan MQTT. Gunakan komponen ini untuk mengalirkan data video dari perangkat Anda untuk digunakan dengan AWS IoT layanan. Untuk menghasilkan sumber daya yang diperlukan dan menerapkan konektor tepi untuk Kinesis Video Streams, lihat [Memulai dengan konektor tepi untuk aliran video Kinesis](#) aktif. GitHub Untuk informasi selengkapnya tentang AWS IoT Greengrass komponen, lihat AWS IoT Greengrass dokumentasi pada [konektor tepi untuk Kinesis Video Streams](#).

Setelah Anda membuat AWS IoT SiteWise model yang diperlukan dan mengonfigurasi komponen Kinesis Video Streams Greengrass, Anda dapat melakukan streaming atau merekam video di tepi ke aplikasi kembar digital Anda di konsol. AWS IoT TwinMaker Anda juga dapat melihat streaming langsung dan metadata dari perangkat Anda di dasbor Grafana. Untuk informasi lebih lanjut tentang mengintegrasikan Grafana AWS IoT TwinMaker dan, lihat. [AWS IoT TwinMaker Integrasi dasbor Grafana](#)

Gunakan konektor tepi untuk aliran video Kinesis untuk mengalirkan video AWS IoT TwinMaker

Dengan konektor tepi untuk aliran video Kinesis, Anda dapat melakukan streaming video dan data ke entitas dalam adegan Anda AWS IoT TwinMaker. Anda menggunakan komponen video untuk melakukan ini. Untuk membuat komponen video untuk digunakan dalam adegan Anda, selesaikan prosedur berikut.

Prasyarat

Sebelum Anda membuat komponen video dalam AWS IoT TwinMaker adegan Anda, pastikan Anda telah menyelesaikan prasyarat berikut.

- Membuat AWS IoT SiteWise model dan aset yang diperlukan untuk konektor tepi untuk aliran video Kinesis. Untuk informasi selengkapnya tentang membuat AWS IoT SiteWise aset untuk konektor, lihat [Memulai konektor tepi untuk aliran video Kinesis](#).
- Menerapkan konektor tepi aliran video Kinesis di AWS IoT Greengrass perangkat Anda. [Untuk informasi selengkapnya tentang penerapan komponen konektor tepi aliran video Kinesis, lihat penerapan README](#).

Buat komponen video untuk AWS IoT TwinMaker adegan

Selesaikan langkah-langkah berikut untuk membuat konektor tepi untuk komponen aliran video Kinesis untuk adegan Anda.

1. Di AWS IoT TwinMaker konsol, buka adegan yang ingin Anda tambahkan komponen video.
2. Setelah adegan terbuka, pilih entitas yang ada atau buat entitas yang ingin Anda tambahkan komponennya, lalu pilih Tambah komponen.
3. Di panel Add component, masukkan nama untuk komponen, dan untuk Type, pilih `com.amazon.iotsitewise.connector.edgevideo`.
4. Pilih Model Aset dengan memilih nama model AWS IoT SiteWise kamera yang Anda buat. Nama ini harus memiliki format berikut: `EdgeConnectorForKVSCameraModel-0abc`, di mana string huruf dan angka di akhir cocok dengan nama aset Anda sendiri.
5. Untuk Aset, pilih aset AWS IoT SiteWise kamera yang ingin Anda streaming video. Sebuah jendela kecil muncul yang menunjukkan pratinjau aliran video saat ini.

Note

Untuk menguji streaming video Anda, pilih uji. Tes ini mengirimkan acara MQTT untuk memulai streaming langsung video. Tunggu beberapa saat untuk melihat video muncul di pemutar.

6. Untuk menambahkan komponen video ke entitas Anda, pilih Tambah komponen.

Tambahkan video dan metadata dari aliran video Kinesis ke dasbor Grafana

Setelah Anda membuat komponen video untuk entitas Anda di AWS IoT TwinMaker adegan Anda, Anda dapat mengonfigurasi panel video di Grafana untuk melihat streaming langsung. Pastikan Anda telah terintegrasi dengan AWS IoT TwinMaker Grafana dengan benar. Untuk informasi selengkapnya, lihat [AWS IoT TwinMakerIntegrasi dasbor Grafana](#).

Important

Untuk melihat video di dasbor Grafana Anda, Anda harus memastikan sumber data Grafana memiliki izin IAM yang tepat. Untuk membuat peran dan kebijakan yang diperlukan, lihat [Membuat peran IAM dasbor](#).

Selesaikan langkah-langkah berikut untuk melihat Kinesis Video Streams dan metadata di dasbor Grafana Anda.

1. Buka AWS IoT TwinMaker dasbor.
2. Pilih Tambah panel, lalu pilih Tambahkan panel kosong.
3. Dari daftar panel, pilih panel pemutar AWS IoT TwinMaker video.
4. Di panel pemutar AWS IoT TwinMaker video, masukkan nama streaming KinesisVideoStreamName, dengan nama aliran video Kinesis yang ingin Anda streaming video.

Note

Untuk melakukan streaming metadata ke panel video Grafana, Anda harus terlebih dahulu membuat entitas dengan komponen streaming video.

5. Opsional: Untuk mengalirkan metadata dari AWS IoT SiteWise aset ke pemutar video, untuk Entitas, pilih AWS IoT TwinMaker entitas yang Anda buat di adegan AndaAWS IoT TwinMaker. Untuk nama Komponen, pilih komponen video yang Anda buat untuk entitas dalam AWS IoT TwinMaker adegan Anda.

Menggunakan AWS IoT TwinMaker pustaka Flink

AWS IoT TwinMaker menyediakan perpustakaan Flink yang dapat Anda gunakan untuk membaca dan menulis data ke penyimpanan data eksternal yang digunakan dalam kembar digital Anda.

Anda menggunakan pustaka AWS IoT TwinMaker Flink dengan menginstalnya sebagai konektor kustom di Managed Service for Apache Flink dan melakukan query Flink SQL di notebook Zeppelin di Managed Service for Apache Flink. Notebook dapat dipromosikan ke aplikasi pemrosesan aliran yang terus berjalan. Pustaka memanfaatkan AWS IoT TwinMaker komponen untuk mengambil data dari ruang kerja Anda.

Pustaka AWS IoT TwinMaker Flink membutuhkan yang berikut ini.

Prasyarat

1. Ruang kerja yang terisi penuh dengan adegan dan komponen. Gunakan tipe komponen bawaan untuk data dari AWS layanan (AWS IoT SiteWise dan Kinesis Video Streams). Buat tipe komponen khusus untuk data dari sumber pihak ketiga. Untuk informasi selengkapnya, lihat [???](#).
2. Pemahaman notebook Studio dengan Managed Service untuk Apache Flink untuk Apache Flink. [Notebook ini didukung oleh Apache Zeppelin dan menggunakan kerangka Apache Flink.](#) Untuk informasi selengkapnya, lihat [Menggunakan notebook Studio dengan Managed Service for Apache Flink for Apache Flink.](#)

Untuk petunjuk tentang penggunaan pustaka, lihat [panduan pengguna pustaka AWS IoT TwinMaker Flink.](#)

Untuk petunjuk tentang pengaturan AWS IoT TwinMaker dengan mulai cepat dalam [AWS IoT TwinMaker sampel](#), lihat [file README untuk aplikasi wawasan sampel.](#)

Pencatatan dan pemantauan di AWS IoT TwinMaker

Pemantauan adalah bagian penting dari pemeliharaan keandalan, ketersediaan, dan performa AWS IoT TwinMaker dan AWS solusi Anda yang lainnya. AWS IoT TwinMaker mendukung alat pemantauan berikut untuk mengawasi layanan, melaporkan saat terjadi kesalahan, dan mengambil tindakan otomatis jika diperlukan:

- Amazon CloudWatch memantau secara waktu nyata AWS dan aplikasi yang Anda jalankan di AWS. Anda dapat mengumpulkan dan melacak metrik, membuat dasbor yang disesuaikan, dan mengatur alarm yang memberi tahu Anda atau mengambil tindakan saat metrik tertentu mencapai ambang batas yang ditentukan. Misalnya, Anda dapat CloudWatch melacak penggunaan CPU atau metrik lainnya untuk instans Amazon EC2 Anda dan secara otomatis meluncurkan instans baru ketika diperlukan. Untuk informasi selengkapnya, lihat [Panduan CloudWatch Pengguna Amazon](#).
- Amazon CloudWatch Logs memantau, menyimpan, dan menyediakan akses ke file log Anda dari AWS IoT TwinMaker gateway, CloudTrail, dan sumber lainnya. CloudWatch Log dapat memantau informasi dalam file log dan memberi tahu Anda ketika ambang tertentu terpenuhi. Anda juga dapat mengarsipkan data log Anda dalam penyimpanan yang sangat tahan lama. Untuk informasi selengkapnya, lihat [Panduan Pengguna Amazon CloudWatch Logs](#).
- AWS CloudTrail merekam panggilan API dan peristiwa terkait yang dilakukan oleh atau atas nama akun AWS Anda dan mengirimkan berkas log ke bucket Amazon S3 yang Anda tentukan. Anda dapat mengidentifikasi pengguna dan akun mana yang memanggil AWS, alamat IP sumber yang melakukan panggilan, dan kapan panggilan tersebut terjadi. Untuk mengetahui informasi selengkapnya, lihat [Panduan Pengguna AWS CloudTrail](#).

Topik

- [Pemantauan AWS IoT TwinMaker dengan CloudWatch metrik Amazon](#)
- [Mencatat panggilan API AWS IoT TwinMaker dengan AWS CloudTrail](#)

Pemantauan AWS IoT TwinMaker dengan CloudWatch metrik Amazon

Anda dapat memantau AWS IoT TwinMaker dengan menggunakan CloudWatch, yang mengumpulkan data mentah dan memprosesnya menjadi metrik yang dapat dibaca dan hampir waktu nyata. Statistik ini disimpan untuk jangka waktu 15 bulan, sehingga Anda dapat mengakses

informasi historis dan mendapatkan perspektif yang lebih baik tentang performa aplikasi atau layanan web Anda. Anda juga dapat mengatur alarm yang memperhatikan ambang batas tertentu dan mengirim notifikasi atau mengambil tindakan saat ambang batas tersebut terpenuhi. Untuk informasi selengkapnya, lihat [Panduan CloudWatch Pengguna Amazon](#).

AWS IoT TwinMaker menerbitkan metrik dan dimensi yang tercantum di bagian berikut ke `AWS/IoTTwinMaker` ruang nama.

Tip

AWS IoT TwinMaker menerbitkan metrik pada interval satu menit. Saat Anda melihat metrik ini dalam grafik di CloudWatch konsol, sebaiknya pilih Periode 1 menit untuk melihat resolusi tertinggi data metrik Anda yang tersedia.

Daftar Isi

- [Metrik](#)

Metrik

AWS IoT TwinMaker menerbitkan metrik berikut.

Metrik

Metrik	Deskripsi
<code>ComponentTypeCreationFailure</code>	<p>Metrik ini melaporkan apakah pembuatan jenis komponen berhasil.</p> <p>Metrik dipublikasikan saat jenis komponen dalam <code>CREATING</code> keadaan. Hal ini terjadi ketika jenis komponen dibuat dengan properti yang diperlukan dalam initializer skema dan properti ini dipakai dengan nilai default.</p> <p>Nilai metrik dapat baik <code>0</code> untuk sukses atau <code>1</code> untuk kegagalan.</p> <p>Dimensi: <code>ComponentTypeId</code>, <code>WorkspaceId</code>.</p>

Metrik	Deskripsi
<code>ComponentTypeUpdateFailure</code>	<p>Unit: Hitung</p> <p>Metrik ini melaporkan apakah pembaruan jenis komponen berhasil.</p> <p>Metrik dipublikasikan saat jenis komponen dalamUPDATING keadaan. Hal ini terjadi ketika jenis komponen diperbarui dengan properti yang diperlukan dalam initializer skema dan properti ini dipakai dengan nilai default.</p> <p>Nilai metrik dapat baik0 untuk sukses atau1 untuk kegagalan.</p> <p>Dimensi: ComponentTypeId, WorkspaceId.</p> <p>Unit: Hitung</p>
<code>EntityCreationFailure</code>	<p>Metrik ini melaporkan apakah pembuatan entitas berhasil. Metrik dipublikasikan saat entitas berada dalamCREATING keadaan. Hal ini terjadi ketika entitas dibuat dengan komponen.</p> <p>Nilai metrik dapat baik0 untuk sukses atau1 untuk kegagalan.</p> <p>Dimensi: EntityName, EntityId, WorkspaceId.</p> <p>Unit: Hitung</p>

Metrik	Deskripsi
EntityUpdateFailure	<p>Metrik ini melaporkan apakah pembaruan entitas berhasil. Metrik dipublikasikan saat entitas berada dalamUPDATING keadaan. Hal ini terjadi ketika entitas diperbarui.</p> <p>Nilai metrik dapat baik0 untuk sukses atau1 untuk kegagalan.</p> <p>Dimensi: EntityName, EntityId, WorkspaceId.</p> <p>Unit: Hitung</p>
EntityDeletionFailure	<p>Metrik ini melaporkan apakah penghapusan entitas berhasil. Metrik dipublikasikan saat entitas berada dalamDELETING keadaan. Hal ini terjadi ketika entitas dihapus.</p> <p>Nilai metrik dapat baik0 untuk sukses atau1 untuk kegagalan.</p> <p>Dimensi: EntityName, EntityId, WorkspaceId.</p> <p>Unit: Hitung</p>

 Tip

Semua metrik dipublikasikan keAWS/IoTTwinMaker namespace.

Mencatat panggilan API AWS IoT TwinMaker dengan AWS CloudTrail

AWS IoT TwinMakerterintegrasi denganAWS CloudTrail, layanan yang menyediakan catatan tindakan yang diambil oleh pengguna, peran, atauAWS layanan diAWS IoT TwinMaker. CloudTrail merekam panggilan API untukAWS IoT TwinMaker sebagai peristiwa. Panggilan yang direkam mencakup panggilan dari AWS IoT TwinMaker konsol dan panggilan kode ke operasi API AWS

IoT TwinMaker ini. Jika membuat jejak, Anda dapat mengaktifkan pengiriman CloudTrail peristiwa berkelanjutan ke bucket Amazon S3, termasuk kejadian untuk AWS IoT TwinMaker. Jika Anda tidak mengonfigurasi jejak, Anda masih dapat melihat peristiwa terbaru di dalam CloudTrail konsol di Riwayat kejadian. Menggunakan informasi yang dikumpulkan oleh CloudTrail, Anda dapat menentukan permintaan yang dibuat ke AWS IoT TwinMaker, alamat IP asal permintaan tersebut dibuat, siapa yang membuat permintaan, kapan permintaan dibuat, dan detail lainnya.

Untuk informasi selengkapnya CloudTrail, lihat [Panduan AWS CloudTrail Pengguna](#).

AWS IoT TwinMaker informasi dalam CloudTrail

Ketika Anda membuat AWS akun Anda, CloudTrail secara otomatis diaktifkan. CloudTrail catatan mendukung aktivitas peristiwa yang terjadi di AWS IoT TwinMaker, bersama dengan peristiwa AWS layanan lainnya dalam sejarah peristiwa. Anda dapat melihat, mencari, dan mengunduh peristiwa terbaru di akun AWS Anda. Untuk informasi selengkapnya, lihat [Menampilkan peristiwa dengan riwayat CloudTrail peristiwa](#).

Untuk catatan berkelanjutan tentang peristiwa di akun AWS Anda, termasuk peristiwa untuk AWS IoT TwinMaker, buat jejak. Jejak memungkinkan CloudTrail untuk mengirim berkas log ke bucket Amazon S3. Secara default, saat Anda membuat jejak di dalam konsol tersebut, jejak diterapkan ke semua AWS Wilayah. CloudTrail mencatat peristiwa dari semua Wilayah di AWS partisi dan mengirimkan berkas log ke bucket Amazon S3 yang Anda tentukan. Selain itu, Anda dapat mengonfigurasi AWS layanan lainnya untuk dianalisis lebih lanjut dan bertindak berdasarkan data peristiwa yang dikumpulkan di CloudTrail log. Untuk informasi selengkapnya, lihat yang berikut:

- [Gambaran umum untuk membuat jejak](#)
- [CloudTrail yang didukung dan integrasi](#)
- [Mengonfigurasi notifikasi Amazon SNS untuk CloudTrail](#)
- [Menerima file CloudTrail log dari beberapa Wilayah](#) dan [Menerima file CloudTrail log dari beberapa akun](#)

Sebagian besar AWS IoT TwinMaker operasi dicatat oleh CloudTrail dan didokumentasikan dalam [Referensi AWS IoT TwinMaker API](#).

Operasi data berikut tidak dicatat oleh CloudTrail:

- [GetPropertyValue](#)
- [GetPropertyValueHistory](#)

- [BatchPutPropertyValues](#)

Setiap entri peristiwa atau log berisi informasi tentang siapa yang membuat permintaan tersebut. Informasi identitas membantu Anda menentukan hal berikut:

- Baik permintaan tersebut dibuat dengan kredensial pengguna root atau .
- Baik permintaan tersebut dibuat dengan kredensial keamanan sementara untuk peran atau pengguna gabungan.
- Bahwa permintaan dibuat oleh layanan AWS lain.

Untuk informasi selengkapnya, lihat [Elemen userIdentity CloudTrail](#) .

Keamanan di AWS IoT TwinMaker

Keamanan cloud di AWS adalah prioritas tertinggi. Sebagai AWS pelanggan, Anda mendapat manfaat dari pusat data dan arsitektur jaringan yang dibangun untuk memenuhi persyaratan organisasi yang paling sensitif terhadap keamanan.

Keamanan adalah tanggung jawab bersama antara Anda AWS dan Anda. [Model tanggung jawab bersama](#) menjelaskan hal ini sebagai keamanan cloud dan keamanan dalam cloud:

- Keamanan cloud — AWS bertanggung jawab untuk melindungi infrastruktur yang menjalankan AWS layanan di AWS Cloud. AWS juga memberi Anda layanan yang dapat Anda gunakan dengan aman. Auditor pihak ketiga secara teratur menguji dan memverifikasi efektivitas keamanan kami sebagai bagian dari [Program AWS Kepatuhan Program AWS Kepatuhan](#) . Untuk mempelajari tentang program kepatuhan yang berlaku AWS IoT TwinMaker, lihat [AWS Layanan dalam Lingkup oleh AWS Layanan Program Kepatuhan](#) .
- Keamanan di cloud — Tanggung jawab Anda ditentukan oleh AWS layanan yang Anda gunakan. Anda juga bertanggung jawab atas faktor lain, yang mencakup sensitivitas data Anda, persyaratan perusahaan Anda, serta undang-undang dan peraturan yang berlaku.

Dokumentasi ini membantu Anda memahami cara menerapkan model tanggung jawab bersama saat menggunakan AWS IoT TwinMaker. Topik berikut menunjukkan cara mengonfigurasi AWS IoT TwinMaker untuk memenuhi tujuan keamanan dan kepatuhan Anda. Anda juga belajar cara menggunakan AWS layanan lain yang membantu Anda memantau dan mengamankan AWS IoT TwinMaker sumber daya Anda.

Topik

- [Perlindungan data di AWS IoT TwinMaker](#)
- [Identity and Access Management untuk AWS IoT TwinMaker](#)
- [AWS IoT TwinMaker dan VPC titik akhir antarmuka \(AWS PrivateLink\)](#)
- [Validasi Kepatuhan untuk AWS IoT TwinMaker](#)
- [Ketahanan di AWS IoT TwinMaker](#)
- [Keamanan Infrastruktur di AWS IoT TwinMaker](#)

Perlindungan data di AWS IoT TwinMaker

[Model tanggung jawab AWS bersama model](#) berlaku untuk perlindungan data di AWS IoT TwinMaker. Seperti yang dijelaskan dalam model AWS ini, bertanggung jawab untuk melindungi infrastruktur global yang menjalankan semua AWS Cloud. Anda bertanggung jawab untuk mempertahankan kendali atas konten yang di-host pada infrastruktur ini. Anda juga bertanggung jawab atas tugas-tugas konfigurasi dan manajemen keamanan untuk Layanan AWS yang Anda gunakan. Untuk informasi selengkapnya tentang privasi data, lihat [Privasi Data FAQ](#). Untuk informasi tentang perlindungan data di Eropa, lihat [Model Tanggung Jawab AWS Bersama dan](#) posting GDPR blog di Blog AWS Keamanan.

Untuk tujuan perlindungan data, kami menyarankan Anda melindungi Akun AWS kredensial dan mengatur pengguna individu dengan AWS IAM Identity Center atau AWS Identity and Access Management (IAM). Dengan cara itu, setiap pengguna hanya diberi izin yang diperlukan untuk memenuhi tanggung jawab tugasnya. Kami juga menyarankan supaya Anda mengamankan data dengan cara-cara berikut:

- Gunakan otentikasi multi-faktor (MFA) dengan setiap akun.
- Gunakan SSL/TLS untuk berkomunikasi dengan AWS sumber daya. Kami membutuhkan TLS 1.2 dan merekomendasikan TLS 1.3.
- Siapkan API dan pencatatan aktivitas pengguna dengan AWS CloudTrail.
- Gunakan solusi AWS enkripsi, bersama dengan semua kontrol keamanan default di dalamnya Layanan AWS.
- Gunakan layanan keamanan terkelola lanjut seperti Amazon Macie, yang membantu menemukan dan mengamankan data sensitif yang disimpan di Amazon S3.
- Jika Anda memerlukan FIPS 140-3 modul kriptografi yang divalidasi saat mengakses AWS melalui antarmuka baris perintah atau, gunakan titik akhir API FIPS Untuk informasi selengkapnya tentang FIPS titik akhir yang tersedia, lihat [Federal Information Processing Standard \(FIPS\) 140-3](#).

Kami sangat merekomendasikan agar Anda tidak pernah memasukkan informasi identifikasi yang sensitif, seperti nomor rekening pelanggan Anda, ke dalam tanda atau bidang isian bebas seperti bidang Nama. Ini termasuk ketika Anda bekerja dengan AWS IoT TwinMaker atau lainnya Layanan AWS menggunakan konsol, API, AWS CLI, atau AWS SDKs. Data apa pun yang Anda masukkan ke dalam tanda atau bidang isian bebas yang digunakan untuk nama dapat digunakan untuk log penagihan atau log diagnostik. Jika Anda memberikan URL ke server eksternal, kami sangat

menyarankan agar Anda tidak menyertakan informasi kredensial dalam URL untuk memvalidasi permintaan Anda ke server tersebut.

Enkripsi diam

AWS IoT TwinMaker menyimpan informasi ruang kerja Anda di bucket Amazon S3 yang dibuat layanan untuk Anda, jika Anda mau. Bucket yang dibuat layanan untuk Anda mengaktifkan enkripsi sisi server default. Jika Anda memilih untuk menggunakan bucket Amazon S3 sendiri saat membuat ruang kerja baru, sebaiknya aktifkan enkripsi sisi server default. Untuk informasi selengkapnya tentang enkripsi default di Amazon S3, lihat [Menyetel perilaku enkripsi sisi server default untuk bucket Amazon S3](#).

Enkripsi bergerak

Semua data yang AWS IoT TwinMaker dikirim ke dikirim melalui TLS koneksi menggunakan HTTPS protokol, sehingga aman secara default saat dalam perjalanan.

Note

Kami menyarankan Anda menggunakan alamat bucket HTTPS Amazon S3 sebagai kontrol untuk menerapkan enkripsi saat transit saat AWS IoT TwinMaker berinteraksi dengan bucket Amazon S3. Untuk informasi selengkapnya tentang bucket Amazon S3, lihat [Membuat, mengonfigurasi, dan bekerja dengan bucket Amazon S3](#).

Identity and Access Management untuk AWS IoT TwinMaker

AWS Identity and Access Management (IAM) adalah Layanan AWS yang membantu administrator mengontrol akses ke AWS sumber daya dengan aman. IAM administrator mengontrol siapa yang dapat diautentikasi (masuk) dan diberi wewenang (memiliki izin) untuk menggunakan sumber daya. AWS IoT TwinMaker IAM adalah Layanan AWS yang dapat Anda gunakan tanpa biaya tambahan.

Topik

- [Audiens](#)
- [Mengautentikasi dengan identitas](#)
- [Mengelola akses menggunakan kebijakan](#)

- [Bagaimana AWS IoT TwinMaker bekerja dengan IAM](#)
- [Contoh kebijakan berbasis identitas untuk AWS IoT TwinMaker](#)
- [Memecahkan masalah AWS IoT TwinMaker identitas dan akses](#)
- [Menggunakan peran terkait layanan untuk AWS IoT TwinMaker](#)
- [AWS kebijakan terkelola untuk AWS IoT TwinMaker](#)

Audiens

Cara Anda menggunakan AWS Identity and Access Management (IAM) berbeda, tergantung pada pekerjaan yang Anda lakukan AWS IoT TwinMaker.

Pengguna layanan — Jika Anda menggunakan AWS IoT TwinMaker layanan untuk melakukan pekerjaan Anda, administrator Anda memberi Anda kredensi dan izin yang Anda butuhkan. Saat Anda menggunakan lebih banyak AWS IoT TwinMaker fitur untuk melakukan pekerjaan Anda, Anda mungkin memerlukan izin tambahan. Memahami cara mengelola akses dapat membantu Anda meminta izin yang tepat dari administrator Anda. Jika Anda tidak dapat mengakses fitur di AWS IoT TwinMaker, lihat [Memecahkan masalah AWS IoT TwinMaker identitas dan akses](#).

Administrator layanan — Jika Anda bertanggung jawab atas AWS IoT TwinMaker sumber daya di perusahaan Anda, Anda mungkin memiliki akses penuh ke AWS IoT TwinMaker. Tugas Anda adalah menentukan AWS IoT TwinMaker fitur dan sumber daya mana yang harus diakses pengguna layanan Anda. Anda kemudian harus mengirimkan permintaan ke IAM administrator Anda untuk mengubah izin pengguna layanan Anda. Tinjau informasi di halaman ini untuk memahami konsep dasar IAM. Untuk mempelajari lebih lanjut tentang bagaimana perusahaan Anda dapat menggunakannya IAM AWS IoT TwinMaker, lihat [Bagaimana AWS IoT TwinMaker bekerja dengan IAM](#).

IAM administrator - Jika Anda seorang IAM administrator, Anda mungkin ingin mempelajari detail tentang cara menulis kebijakan untuk mengelola akses AWS IoT TwinMaker. Untuk melihat contoh kebijakan AWS IoT TwinMaker berbasis identitas yang dapat Anda gunakan, lihat. IAM [Contoh kebijakan berbasis identitas untuk AWS IoT TwinMaker](#)

Mengautentikasi dengan identitas

Otentikasi adalah cara Anda masuk AWS menggunakan kredensi identitas Anda. Anda harus diautentikasi (masuk ke AWS) sebagai Pengguna root akun AWS, sebagai IAM pengguna, atau dengan mengambil peran IAM.

Anda dapat masuk AWS sebagai identitas federasi dengan menggunakan kredensial yang disediakan melalui sumber identitas. AWS IAM Identity Center Pengguna (Pusat IAM Identitas), autentikasi masuk tunggal perusahaan Anda, dan kredensi Google atau Facebook Anda adalah contoh identitas federasi. Saat Anda masuk sebagai identitas federasi, administrator Anda sebelumnya menyiapkan federasi identitas menggunakan IAM peran. Ketika Anda mengakses AWS dengan menggunakan federasi, Anda secara tidak langsung mengambil peran.

Bergantung pada jenis pengguna Anda, Anda dapat masuk ke AWS Management Console atau portal AWS akses. Untuk informasi selengkapnya tentang masuk AWS, lihat [Cara masuk ke Panduan AWS Sign-In Pengguna Anda Akun AWS](#).

Jika Anda mengakses AWS secara terprogram, AWS sediakan kit pengembangan perangkat lunak (SDK) dan antarmuka baris perintah (CLI) untuk menandatangani permintaan Anda secara kriptografis dengan menggunakan kredensial Anda. Jika Anda tidak menggunakan AWS alat, Anda harus menandatangani permintaan sendiri. Untuk informasi selengkapnya tentang menggunakan metode yang disarankan untuk menandatangani permintaan sendiri, lihat [Menandatangani AWS API permintaan](#) di Panduan IAM Pengguna.

Apa pun metode autentikasi yang digunakan, Anda mungkin diminta untuk menyediakan informasi keamanan tambahan. Misalnya, AWS merekomendasikan agar Anda menggunakan otentikasi multi-faktor (MFA) untuk meningkatkan keamanan akun Anda. Untuk mempelajari lebih lanjut, lihat [Autentikasi multi-faktor](#) di Panduan AWS IAM Identity Center Pengguna dan [Menggunakan autentikasi multi-faktor \(MFA\) AWS di](#) Panduan Pengguna. IAM

Akun AWS pengguna root

Saat Anda membuat Akun AWS, Anda mulai dengan satu identitas masuk yang memiliki akses lengkap ke semua Layanan AWS dan sumber daya di akun. Identitas ini disebut pengguna Akun AWS root dan diakses dengan masuk dengan alamat email dan kata sandi yang Anda gunakan untuk membuat akun. Kami sangat menyarankan agar Anda tidak menggunakan pengguna root untuk tugas sehari-hari. Lindungi kredensial pengguna root Anda dan gunakan kredensial tersebut untuk melakukan tugas yang hanya dapat dilakukan pengguna root. Untuk daftar lengkap tugas yang mengharuskan Anda masuk sebagai pengguna root, lihat [Tugas yang memerlukan kredensi pengguna root](#) di IAMPanduan Pengguna.

Identitas gabungan

Sebagai praktik terbaik, mewajibkan pengguna manusia, termasuk pengguna yang memerlukan akses administrator, untuk menggunakan federasi dengan penyedia identitas untuk mengakses Layanan AWS dengan menggunakan kredensi sementara.

Identitas federasi adalah pengguna dari direktori pengguna perusahaan Anda, penyedia identitas web, direktori Pusat Identitas AWS Directory Service, atau pengguna mana pun yang mengakses Layanan AWS dengan menggunakan kredensial yang disediakan melalui sumber identitas. Ketika identitas federasi mengakses Akun AWS, mereka mengambil peran, dan peran memberikan kredensi sementara.

Untuk manajemen akses terpusat, kami sarankan Anda menggunakan AWS IAM Identity Center. Anda dapat membuat pengguna dan grup di Pusat IAM Identitas, atau Anda dapat menghubungkan dan menyinkronkan ke sekumpulan pengguna dan grup di sumber identitas Anda sendiri untuk digunakan di semua aplikasi Akun AWS dan aplikasi Anda. Untuk informasi tentang Pusat IAM Identitas, lihat [Apa itu Pusat IAM Identitas?](#) dalam AWS IAM Identity Center User Guide.

Pengguna dan grup IAM

[IAMPengguna](#) adalah identitas dalam diri Anda Akun AWS yang memiliki izin khusus untuk satu orang atau aplikasi. Jika memungkinkan, sebaiknya mengandalkan kredensi sementara daripada membuat IAM pengguna yang memiliki kredensi jangka panjang seperti kata sandi dan kunci akses. Namun, jika Anda memiliki kasus penggunaan khusus yang memerlukan kredensial jangka panjang dengan IAM pengguna, kami sarankan Anda memutar kunci akses. Untuk informasi selengkapnya, lihat [Memutar kunci akses secara teratur untuk kasus penggunaan yang memerlukan kredensi jangka panjang](#) di IAMPanduan Pengguna.

[IAMGrup](#) adalah identitas yang menentukan kumpulan IAM pengguna. Anda tidak dapat masuk sebagai grup. Anda dapat menggunakan grup untuk menentukan izin bagi beberapa pengguna sekaligus. Grup mempermudah manajemen izin untuk sejumlah besar pengguna sekaligus. Misalnya, Anda dapat memiliki grup bernama IAMAdmins dan memberikan izin grup tersebut untuk mengelola sumber daya IAM.

Pengguna berbeda dari peran. Pengguna secara unik terkait dengan satu orang atau aplikasi, tetapi peran dimaksudkan untuk dapat digunakan oleh siapa pun yang membutuhkannya. Pengguna memiliki kredensial jangka panjang permanen, tetapi peran memberikan kredensial sementara. Untuk mempelajari lebih lanjut, lihat [Kapan membuat IAM pengguna \(bukan peran\)](#) di Panduan IAM Pengguna.

IAMperan

[IAMPeran](#) adalah identitas dalam diri Anda Akun AWS yang memiliki izin khusus. Ini mirip dengan IAM pengguna, tetapi tidak terkait dengan orang tertentu. Anda dapat mengambil IAM peran sementara AWS Management Console dengan [beralih peran](#). Anda dapat mengambil peran dengan memanggil AWS CLI atau AWS API operasi atau dengan menggunakan kustomURL. Untuk informasi selengkapnya tentang metode penggunaan peran, lihat [Menggunakan IAM peran](#) di Panduan IAM Pengguna.

IAMperan dengan kredensi sementara berguna dalam situasi berikut:

- Akses pengguna terfederasi – Untuk menetapkan izin ke identitas terfederasi, Anda membuat peran dan menentukan izin untuk peran tersebut. Ketika identitas terfederasi mengautentikasi, identitas tersebut terhubung dengan peran dan diberi izin yang ditentukan oleh peran. Untuk informasi tentang peran untuk federasi, lihat [Membuat peran untuk Penyedia Identitas pihak ketiga](#) di Panduan IAM Pengguna. Jika Anda menggunakan Pusat IAM Identitas, Anda mengonfigurasi set izin. Untuk mengontrol apa yang dapat diakses identitas Anda setelah diautentikasi, Pusat IAM Identitas mengkorelasikan izin yang disetel ke peran. IAM Untuk informasi tentang set izin, lihat [Set izin](#) dalam Panduan Pengguna AWS IAM Identity Center .
- Izin IAM pengguna sementara — IAM Pengguna atau peran dapat mengambil IAM peran untuk sementara mengambil izin yang berbeda untuk tugas tertentu.
- Akses lintas akun — Anda dapat menggunakan IAM peran untuk memungkinkan seseorang (prinsipal tepercaya) di akun lain mengakses sumber daya di akun Anda. Peran adalah cara utama untuk memberikan akses lintas akun. Namun, dengan beberapa Layanan AWS, Anda dapat melampirkan kebijakan secara langsung ke sumber daya (alih-alih menggunakan peran sebagai proxy). Untuk mempelajari perbedaan antara peran dan kebijakan berbasis sumber daya untuk akses lintas akun, lihat [Akses sumber daya lintas akun di IAM](#) Panduan Pengguna. IAM
- Akses lintas layanan — Beberapa Layanan AWS menggunakan fitur lain Layanan AWS. Misalnya, saat Anda melakukan panggilan dalam suatu layanan, biasanya layanan tersebut menjalankan aplikasi di Amazon EC2 atau menyimpan objek di Amazon S3. Sebuah layanan mungkin melakukannya menggunakan izin prinsipal yang memanggil, menggunakan peran layanan, atau peran terkait layanan.
- Sesi akses teruskan (FAS) — Saat Anda menggunakan IAM pengguna atau peran untuk melakukan tindakan AWS, Anda dianggap sebagai prinsipal. Ketika Anda menggunakan beberapa layanan, Anda mungkin melakukan sebuah tindakan yang kemudian menginisiasi tindakan lain di layanan yang berbeda. FAS menggunakan izin dari pemanggilan utama Layanan AWS, dikombinasikan dengan permintaan Layanan AWS untuk membuat permintaan ke layanan

hilir. FASPermintaan hanya dibuat ketika layanan menerima permintaan yang memerlukan interaksi dengan orang lain Layanan AWS atau sumber daya untuk menyelesaikannya. Dalam hal ini, Anda harus memiliki izin untuk melakukan kedua tindakan tersebut. Untuk detail kebijakan saat membuat FAS permintaan, lihat [Meneruskan sesi akses](#).

- Peran layanan — Peran layanan adalah [IAMperan](#) yang diasumsikan layanan untuk melakukan tindakan atas nama Anda. IAMAdministrator dapat membuat, memodifikasi, dan menghapus peran layanan dari dalamIAM. Untuk informasi selengkapnya, lihat [Membuat peran untuk mendelegasikan izin ke Layanan AWS](#) dalam IAMPanduan Pengguna.
- Peran terkait layanan — Peran terkait layanan adalah jenis peran layanan yang ditautkan ke peran layanan. Layanan AWS Layanan tersebut dapat menjalankan peran untuk melakukan tindakan atas nama Anda. Peran terkait layanan muncul di Anda Akun AWS dan dimiliki oleh layanan. IAMAdministrator dapat melihat, tetapi tidak mengedit izin untuk peran terkait layanan.
- Aplikasi yang berjalan di Amazon EC2 — Anda dapat menggunakan IAM peran untuk mengelola kredensial sementara untuk aplikasi yang berjalan pada EC2 instance dan membuat AWS CLI atau AWS API meminta. Ini lebih baik untuk menyimpan kunci akses dalam EC2 instance. Untuk menetapkan AWS peran ke EC2 instance dan membuatnya tersedia untuk semua aplikasinya, Anda membuat profil instance yang dilampirkan ke instance. Profil instance berisi peran dan memungkinkan program yang berjalan pada EC2 instance untuk mendapatkan kredensial sementara. Untuk informasi selengkapnya, lihat [Menggunakan IAM peran untuk memberikan izin ke aplikasi yang berjalan di EC2 instans Amazon](#) di IAMPanduan Pengguna.

Untuk mempelajari apakah akan menggunakan IAM peran atau IAM pengguna, lihat [Kapan membuat IAM peran \(bukan pengguna\)](#) di Panduan IAM Pengguna.

Mengelola akses menggunakan kebijakan

Anda mengontrol akses AWS dengan membuat kebijakan dan melampirkannya ke AWS identitas atau sumber daya. Kebijakan adalah objek AWS yang, ketika dikaitkan dengan identitas atau sumber daya, menentukan izinnya. AWS mengevaluasi kebijakan ini ketika prinsipal (pengguna, pengguna root, atau sesi peran) membuat permintaan. Izin dalam kebijakan menentukan apakah permintaan diizinkan atau ditolak. Sebagian besar kebijakan disimpan AWS sebagai JSON dokumen. Untuk informasi selengkapnya tentang struktur dan isi dokumen JSON kebijakan, lihat [Ringkasan JSON kebijakan](#) di Panduan IAM Pengguna.

Administrator dapat menggunakan AWS JSON kebijakan untuk menentukan siapa yang memiliki akses ke apa. Yaitu, principal dapat melakukan tindakan pada suatu sumber daya, dan dalam suatu syarat.

Secara default, pengguna dan peran tidak memiliki izin. Untuk memberikan izin kepada pengguna untuk melakukan tindakan pada sumber daya yang mereka butuhkan, IAM administrator dapat membuat IAM kebijakan. Administrator kemudian dapat menambahkan IAM kebijakan ke peran, dan pengguna dapat mengambil peran.

IAMkebijakan menentukan izin untuk tindakan terlepas dari metode yang Anda gunakan untuk melakukan operasi. Misalnya, anggaplah Anda memiliki kebijakan yang mengizinkan tindakan `iam:GetRole`. Pengguna dengan kebijakan itu bisa mendapatkan informasi peran dari AWS Management Console, AWS CLI, atau AWS API.

Kebijakan berbasis identitas

Kebijakan berbasis identitas adalah dokumen kebijakan JSON izin yang dapat Anda lampirkan ke identitas, seperti pengguna, grup IAM pengguna, atau peran. Kebijakan ini mengontrol jenis tindakan yang dapat dilakukan oleh pengguna dan peran, di sumber daya mana, dan berdasarkan kondisi seperti apa. Untuk mempelajari cara membuat kebijakan berbasis identitas, lihat [Membuat IAM kebijakan di Panduan Pengguna](#). IAM

Kebijakan berbasis identitas dapat dikategorikan lebih lanjut sebagai kebijakan inline atau kebijakan yang dikelola. Kebijakan inline disematkan langsung ke satu pengguna, grup, atau peran. Kebijakan terkelola adalah kebijakan mandiri yang dapat Anda lampirkan ke beberapa pengguna, grup, dan peran dalam. Akun AWS Kebijakan AWS terkelola mencakup kebijakan terkelola dan kebijakan yang dikelola pelanggan. Untuk mempelajari cara memilih antara kebijakan terkelola atau kebijakan sebaris, lihat [Memilih antara kebijakan terkelola dan kebijakan sebaris](#) di IAMPanduan Pengguna.

Kebijakan berbasis sumber daya

Kebijakan berbasis sumber daya adalah dokumen JSON kebijakan yang Anda lampirkan ke sumber daya. Contoh kebijakan berbasis sumber daya adalah kebijakan kepercayaan IAM peran dan kebijakan bucket Amazon S3. Dalam layanan yang mendukung kebijakan berbasis sumber daya, administrator layanan dapat menggunakannya untuk mengontrol akses ke sumber daya tertentu. Untuk sumber daya tempat kebijakan dilampirkan, kebijakan menentukan tindakan apa yang dapat dilakukan oleh prinsipal tertentu pada sumber daya tersebut dan dalam kondisi apa. Anda harus [menentukan prinsipal](#) dalam kebijakan berbasis sumber daya. Prinsipal dapat mencakup akun, pengguna, peran, pengguna federasi, atau. Layanan AWS

Kebijakan berbasis sumber daya merupakan kebijakan inline yang terletak di layanan tersebut. Anda tidak dapat menggunakan kebijakan AWS terkelola IAM dalam kebijakan berbasis sumber daya.

Daftar kontrol akses (ACLs)

Access control lists (ACLs) mengontrol prinsipal mana (anggota akun, pengguna, atau peran) yang memiliki izin untuk mengakses sumber daya. ACLs mirip dengan kebijakan berbasis sumber daya, meskipun mereka tidak menggunakan format dokumen kebijakan. JSON

Amazon S3, AWS WAF, dan Amazon VPC adalah contoh layanan yang mendukung ACLs. Untuk mempelajari selengkapnya ACLs, lihat [Ikhtisar daftar kontrol akses \(ACL\)](#) di Panduan Pengembang Layanan Penyimpanan Sederhana Amazon.

Jenis-jenis kebijakan lain

AWS mendukung jenis kebijakan tambahan yang kurang umum. Jenis-jenis kebijakan ini dapat mengatur izin maksimum yang diberikan kepada Anda oleh jenis kebijakan yang lebih umum.

- **Batas izin** — Batas izin adalah fitur lanjutan tempat Anda menetapkan izin maksimum yang dapat diberikan oleh kebijakan berbasis identitas kepada entitas (pengguna atau peran). IAM IAM Anda dapat menetapkan batasan izin untuk suatu entitas. Izin yang dihasilkan adalah perpotongan antara kebijakan berbasis identitas milik entitas dan batasan izinnya. Kebijakan berbasis sumber daya yang menentukan pengguna atau peran dalam bidang `Principal` tidak dibatasi oleh batasan izin. Penolakan eksplisit dalam salah satu kebijakan ini akan menggantikan pemberian izin. Untuk informasi selengkapnya tentang batas izin, lihat [Batas izin untuk IAM entitas](#) di IAMPanduan Pengguna.
- **Kebijakan kontrol layanan (SCPs)** — SCPs adalah JSON kebijakan yang menentukan izin maksimum untuk organisasi atau unit organisasi (OU) di AWS Organizations. AWS Organizations adalah layanan untuk mengelompokkan dan mengelola secara terpusat beberapa Akun AWS yang dimiliki bisnis Anda. Jika Anda mengaktifkan semua fitur dalam suatu organisasi, maka Anda dapat menerapkan kebijakan kontrol layanan (SCPs) ke salah satu atau semua akun Anda. SCPMembatasi izin untuk entitas di akun anggota, termasuk masing-masing Pengguna root akun AWS. Untuk informasi selengkapnya tentang Organizations dan SCPs, lihat [Kebijakan kontrol layanan](#) di Panduan AWS Organizations Pengguna.
- **Kebijakan sesi** – Kebijakan sesi adalah kebijakan lanjutan yang Anda berikan sebagai parameter ketika Anda membuat sesi sementara secara programatis untuk peran atau pengguna terfederasi. Izin sesi yang dihasilkan adalah perpotongan antara kebijakan berbasis identitas pengguna atau peran dan kebijakan sesi. Izin juga bisa datang dari kebijakan berbasis sumber daya. Penolakan secara tegas dalam salah satu kebijakan ini membatalkan izin. Untuk informasi selengkapnya, lihat [Kebijakan sesi](#) di Panduan IAM Pengguna.

Berbagai jenis kebijakan

Ketika beberapa jenis kebijakan berlaku pada suatu permintaan, izin yang dihasilkan lebih rumit untuk dipahami. Untuk mempelajari cara AWS menentukan apakah akan mengizinkan permintaan saat beberapa jenis kebijakan terlibat, lihat [Logika evaluasi kebijakan](#) di Panduan IAM Pengguna.

Bagaimana AWS IoT TwinMaker bekerja dengan IAM

Sebelum Anda menggunakan IAM untuk mengelola akses AWS IoT TwinMaker, pelajari IAM fitur apa yang tersedia untuk digunakan AWS IoT TwinMaker.

IAM fitur yang dapat Anda gunakan dengan AWS IoT TwinMaker

IAM fitur	AWS IoT TwinMaker dukungan
Kebijakan berbasis identitas	Ya
Kebijakan berbasis sumber daya	Tidak
Tindakan kebijakan	Ya
Sumber daya kebijakan	Ya
Kunci kondisi kebijakan	Ya
ACLs	Tidak
ABAC(tag dalam kebijakan)	Parsial
Kredensial sementara	Ya
Izin prinsipal	Ya
Peran layanan	Ya
Peran terkait layanan	Tidak

Untuk mendapatkan tampilan tingkat tinggi tentang cara AWS IoT TwinMaker dan AWS layanan lain bekerja dengan sebagian besar IAM fitur, lihat [AWS layanan yang berfungsi IAM](#) di Panduan AWS IAM Identity Center Pengguna.

Kebijakan berbasis identitas untuk AWS IoT TwinMaker

Mendukung kebijakan berbasis identitas: Ya

Kebijakan berbasis identitas adalah dokumen kebijakan JSON izin yang dapat Anda lampirkan ke identitas, seperti pengguna, grup IAM pengguna, atau peran. Kebijakan ini mengontrol jenis tindakan yang dapat dilakukan oleh pengguna dan peran, di sumber daya mana, dan berdasarkan kondisi seperti apa. Untuk mempelajari cara membuat kebijakan berbasis identitas, lihat [Membuat IAM kebijakan di Panduan Pengguna](#). IAM

Dengan kebijakan IAM berbasis identitas, Anda dapat menentukan tindakan dan sumber daya yang diizinkan atau ditolak serta kondisi di mana tindakan diizinkan atau ditolak. Anda tidak dapat menentukan secara spesifik prinsipal dalam sebuah kebijakan berbasis identitas karena prinsipal berlaku bagi pengguna atau peran yang melekat kepadanya. Untuk mempelajari semua elemen yang dapat Anda gunakan dalam JSON kebijakan, lihat [referensi elemen IAM JSON kebijakan](#) di Panduan IAM Pengguna.

Contoh kebijakan berbasis identitas untuk AWS IoT TwinMaker

Untuk melihat contoh kebijakan AWS IoT TwinMaker berbasis identitas, lihat. [Contoh kebijakan berbasis identitas untuk AWS IoT TwinMaker](#)

Kebijakan berbasis sumber daya dalam AWS IoT TwinMaker

Mendukung kebijakan berbasis sumber daya: Tidak

Kebijakan berbasis sumber daya adalah dokumen JSON kebijakan yang Anda lampirkan ke sumber daya. Contoh kebijakan berbasis sumber daya adalah kebijakan kepercayaan IAM peran dan kebijakan bucket Amazon S3. Dalam layanan yang mendukung kebijakan berbasis sumber daya, administrator layanan dapat menggunakannya untuk mengontrol akses ke sumber daya tertentu. Untuk sumber daya tempat kebijakan dilampirkan, kebijakan menentukan tindakan apa yang dapat dilakukan oleh prinsipal tertentu pada sumber daya tersebut dan dalam kondisi apa. Anda harus [menentukan prinsipal](#) dalam kebijakan berbasis sumber daya. Prinsipal dapat mencakup akun, pengguna, peran, pengguna federasi, atau Layanan AWS

Untuk mengaktifkan akses lintas akun, Anda dapat menentukan seluruh akun atau IAM entitas di akun lain sebagai prinsipal dalam kebijakan berbasis sumber daya. Menambahkan prinsipal akun silang ke kebijakan berbasis sumber daya hanya setengah dari membangun hubungan kepercayaan.

Ketika prinsipal dan sumber daya berbeda Akun AWS, IAM administrator di akun tepercaya juga harus memberikan izin entitas utama (pengguna atau peran) untuk mengakses sumber daya. Mereka memberikan izin dengan melampirkan kebijakan berbasis identitas kepada entitas. Namun, jika kebijakan berbasis sumber daya memberikan akses ke prinsipal dalam akun yang sama, tidak diperlukan kebijakan berbasis identitas tambahan. Untuk informasi selengkapnya, lihat [Akses sumber daya lintas akun IAM di](#) Panduan IAM Pengguna.

Tindakan kebijakan untuk AWS IoT TwinMaker

Mendukung tindakan kebijakan: Ya

Administrator dapat menggunakan AWS JSON kebijakan untuk menentukan siapa yang memiliki akses ke apa. Yaitu, principal dapat melakukan tindakan pada suatu sumber daya, dan dalam suatu syarat.

ActionElemen JSON kebijakan menjelaskan tindakan yang dapat Anda gunakan untuk mengizinkan atau menolak akses dalam kebijakan. Tindakan kebijakan biasanya memiliki nama yang sama dengan AWS API operasi terkait. Ada beberapa pengecualian, seperti tindakan khusus izin yang tidak memiliki operasi yang cocok. API Ada juga beberapa operasi yang memerlukan beberapa tindakan dalam suatu kebijakan. Tindakan tambahan ini disebut tindakan dependen.

Menyertakan tindakan dalam kebijakan untuk memberikan izin untuk melakukan operasi terkait.

Untuk melihat daftar AWS IoT TwinMaker tindakan, lihat [Tindakan yang ditentukan oleh AWS IoT TwinMaker](#) dalam Referensi Otorisasi Layanan.

Tindakan kebijakan AWS IoT TwinMaker menggunakan awalan berikut sebelum tindakan:

```
iottwinmaker
```

Untuk menetapkan secara spesifik beberapa tindakan dalam satu pernyataan, pisahkan tindakan tersebut dengan koma.

```
"Action": [  
  "iottwinmaker:action1",  
  "iottwinmaker:action2"  
]
```

Untuk melihat contoh kebijakan AWS IoT TwinMaker berbasis identitas, lihat [Contoh kebijakan berbasis identitas untuk AWS IoT TwinMaker](#)

Sumber daya kebijakan untuk AWS IoT TwinMaker

Mendukung sumber daya kebijakan: Ya

Administrator dapat menggunakan AWS JSON kebijakan untuk menentukan siapa yang memiliki akses ke apa. Yaitu, principal dapat melakukan tindakan pada suatu sumber daya, dan dalam suatu syarat.

Elemen `Resource` JSON kebijakan menentukan objek atau objek yang tindakan tersebut berlaku. Pernyataan harus menyertakan elemen `Resource` atau `NotResource`. Sebagai praktik terbaik, tentukan sumber daya menggunakan [Amazon Resource Name \(ARN\)](#). Anda dapat melakukan ini untuk tindakan yang mendukung jenis sumber daya tertentu, yang dikenal sebagai izin tingkat sumber daya.

Untuk tindakan yang tidak mendukung izin di tingkat sumber daya, misalnya operasi pencantuman, gunakan wildcard (*) untuk menunjukkan bahwa pernyataan tersebut berlaku untuk semua sumber daya.

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

Untuk melihat daftar jenis sumber daya dan jenis AWS IoT TwinMaker sumber daya ARNs, lihat [Sumber daya yang ditentukan oleh AWS IoT TwinMaker](#) dalam Referensi Otorisasi Layanan. Untuk mempelajari tindakan mana yang dapat Anda tentukan ARN dari setiap sumber daya, lihat [Tindakan yang ditentukan oleh AWS IoT TwinMaker](#).

Untuk melihat contoh kebijakan AWS IoT TwinMaker berbasis identitas, lihat [Contoh kebijakan berbasis identitas untuk AWS IoT TwinMaker](#)

Kunci kondisi kebijakan untuk AWS IoT TwinMaker

Mendukung kunci kondisi kebijakan khusus layanan: Ya

Administrator dapat menggunakan AWS JSON kebijakan untuk menentukan siapa yang memiliki akses ke apa. Yaitu, di mana utama dapat melakukan tindakan pada sumber daya, dan dalam kondisi apa.

Elemen Condition (atau blok Condition) akan memungkinkan Anda menentukan kondisi yang menjadi dasar suatu pernyataan berlaku. Elemen Condition bersifat opsional. Anda dapat membuat ekspresi bersyarat yang menggunakan [operator kondisi](#), misalnya sama dengan atau kurang dari, untuk mencocokkan kondisi dalam kebijakan dengan nilai-nilai yang diminta.

Jika Anda menentukan beberapa elemen Condition dalam sebuah pernyataan, atau beberapa kunci dalam elemen Condition tunggal, maka AWS akan mengevaluasinya menggunakan operasi AND logis. Jika Anda menentukan beberapa nilai untuk satu kunci kondisi, AWS mengevaluasi kondisi menggunakan OR operasi logis. Semua kondisi harus dipenuhi sebelum izin pernyataan diberikan.

Anda juga dapat menggunakan variabel placeholder saat menentukan kondisi. Misalnya, Anda dapat memberikan izin IAM pengguna untuk mengakses sumber daya hanya jika ditandai dengan nama IAM pengguna mereka. Untuk informasi selengkapnya, lihat [elemen IAM kebijakan: variabel dan tag](#) di Panduan IAM Pengguna.

AWS mendukung kunci kondisi global dan kunci kondisi khusus layanan. Untuk melihat semua kunci kondisi AWS global, lihat [kunci konteks kondisi AWS global](#) di Panduan IAM Pengguna.

Untuk melihat daftar kunci AWS IoT TwinMaker kondisi, lihat [Kunci kondisi untuk AWS IoT TwinMaker](#) dalam Referensi Otorisasi Layanan. Untuk mempelajari tindakan dan sumber daya yang dapat Anda gunakan kunci kondisi, lihat [Tindakan yang ditentukan oleh AWS IoT TwinMaker](#).

Untuk melihat contoh kebijakan AWS IoT TwinMaker berbasis identitas, lihat [Contoh kebijakan berbasis identitas untuk AWS IoT TwinMaker](#)

Daftar kontrol akses (ACLs) di AWS IoT TwinMaker

Mendukung ACLs: Tidak

Access control lists (ACLs) mengontrol prinsipal mana (anggota akun, pengguna, atau peran) yang memiliki izin untuk mengakses sumber daya. ACLs mirip dengan kebijakan berbasis sumber daya, meskipun mereka tidak menggunakan format dokumen kebijakan. JSON

Kontrol akses berbasis atribut () ABAC dengan AWS IoT TwinMaker

Mendukung ABAC (tag dalam kebijakan): Sebagian

Attribute-based access control (ABAC) adalah strategi otorisasi yang mendefinisikan izin berdasarkan atribut. Dalam AWS, atribut ini disebut tag. Anda dapat melampirkan tag ke IAM entitas (pengguna

atau peran) dan ke banyak AWS sumber daya. Menandai entitas dan sumber daya adalah langkah pertama dari ABAC. Kemudian Anda merancang ABAC kebijakan untuk mengizinkan operasi ketika tag prinsipal cocok dengan tag pada sumber daya yang mereka coba akses.

ABAC membantu dalam lingkungan yang berkembang pesat dan membantu dengan situasi di mana manajemen kebijakan menjadi rumit.

Untuk mengendalikan akses berdasarkan tag, berikan informasi tentang tag di [elemen kondisi](#) dari kebijakan menggunakan kunci kondisi `aws:ResourceTag/key-name`, `aws:RequestTag/key-name`, atau `aws:TagKeys`.

Jika sebuah layanan mendukung ketiga kunci kondisi untuk setiap jenis sumber daya, nilainya adalah Ya untuk layanan tersebut. Jika suatu layanan mendukung ketiga kunci kondisi untuk hanya beberapa jenis sumber daya, nilainya adalah Parsial.

Untuk informasi lebih lanjut tentang ABAC, lihat [Apa itu ABAC?](#) dalam IAM User Guide. Untuk melihat tutorial dengan langkah-langkah persiapan ABAC, lihat [Menggunakan kontrol akses berbasis atribut \(ABAC\)](#) di IAM Panduan Pengguna.

Menggunakan kredensi sementara dengan AWS IoT TwinMaker

Mendukung kredensi sementara: Ya

Beberapa Layanan AWS tidak berfungsi saat Anda masuk menggunakan kredensi sementara. Untuk informasi tambahan, termasuk yang Layanan AWS bekerja dengan kredensi sementara, lihat [Layanan AWS yang berfungsi IAM](#) di IAM Panduan Pengguna.

Anda menggunakan kredensi sementara jika Anda masuk AWS Management Console menggunakan metode apa pun kecuali nama pengguna dan kata sandi. Misalnya, ketika Anda mengakses AWS menggunakan link sign-on (SSO) tunggal perusahaan Anda, proses tersebut secara otomatis membuat kredensi sementara. Anda juga akan secara otomatis membuat kredensial sementara ketika Anda masuk ke konsol sebagai seorang pengguna lalu beralih peran. Untuk informasi selengkapnya tentang beralih peran, lihat [Beralih ke peran \(konsol\)](#) di Panduan IAM Pengguna.

Anda dapat secara manual membuat kredensi sementara menggunakan atau. AWS CLI AWS API Anda kemudian dapat menggunakan kredensial sementara tersebut untuk mengakses. AWS AWS merekomendasikan agar Anda secara dinamis menghasilkan kredensi sementara alih-alih menggunakan kunci akses jangka panjang. Untuk informasi selengkapnya, lihat [Kredensi keamanan sementara](#) di IAM

Izin utama lintas layanan untuk AWS IoT TwinMaker

Mendukung sesi akses maju (FAS): Ya

Saat Anda menggunakan IAM pengguna atau peran untuk melakukan tindakan AWS, Anda dianggap sebagai prinsipal. Ketika Anda menggunakan beberapa layanan, Anda mungkin melakukan sebuah tindakan yang kemudian menginisiasi tindakan lain di layanan yang berbeda. FAS menggunakan izin dari pemanggilan utama Layanan AWS, dikombinasikan dengan permintaan Layanan AWS untuk membuat permintaan ke layanan hilir. FAS permintaan hanya dibuat ketika layanan menerima permintaan yang memerlukan interaksi dengan orang lain Layanan AWS atau sumber daya untuk menyelesaikannya. Dalam hal ini, Anda harus memiliki izin untuk melakukan kedua tindakan tersebut. Untuk detail kebijakan saat membuat FAS permintaan, lihat [Meneruskan sesi akses](#).

Peran layanan untuk AWS IoT TwinMaker

Mendukung peran layanan: Ya

Peran layanan adalah [IAM peran](#) yang diasumsikan layanan untuk melakukan tindakan atas nama Anda. IAM Administrator dapat membuat, memodifikasi, dan menghapus peran layanan dari dalam IAM. Untuk informasi selengkapnya, lihat [Membuat peran untuk mendelegasikan izin ke Layanan AWS](#) dalam [IAM Panduan Pengguna](#).

Warning

Mengubah izin untuk peran layanan dapat merusak AWS IoT TwinMaker fungsionalitas. Edit peran layanan hanya jika AWS IoT TwinMaker memberikan panduan untuk melakukannya.

Peran terkait layanan untuk AWS IoT TwinMaker

Mendukung peran terkait layanan: Tidak

Peran terkait layanan adalah jenis peran layanan yang ditautkan ke. Layanan AWS Layanan tersebut dapat menjalankan peran untuk melakukan tindakan atas nama Anda. Peran terkait layanan muncul di Anda Akun AWS dan dimiliki oleh layanan. IAM Administrator dapat melihat, tetapi tidak mengedit izin untuk peran terkait layanan.

Untuk detail tentang membuat atau mengelola peran terkait layanan, lihat [AWS layanan yang berfungsi](#) dengannya. IAM Cari layanan dalam tabel yang memiliki Yes di kolom Peran terkait layanan. Pilih tautan Ya untuk melihat dokumentasi peran terkait layanan untuk layanan tersebut.

Contoh kebijakan berbasis identitas untuk AWS IoT TwinMaker

Secara default, pengguna dan peran tidak memiliki izin untuk membuat atau memodifikasi AWS IoT TwinMaker sumber daya. Mereka juga tidak dapat melakukan tugas dengan menggunakan AWS Management Console, AWS Command Line Interface (AWS CLI), atau AWS API. Untuk memberikan izin kepada pengguna untuk melakukan tindakan pada sumber daya yang mereka butuhkan, IAM administrator dapat membuat IAM kebijakan. Administrator kemudian dapat menambahkan IAM kebijakan ke peran, dan pengguna dapat mengambil peran.

Untuk mempelajari cara membuat kebijakan IAM berbasis identitas menggunakan contoh dokumen kebijakan ini, lihat [Membuat JSON IAM kebijakan di Panduan Pengguna](#). IAM

Untuk detail tentang tindakan dan jenis sumber daya yang ditentukan oleh AWS IoT TwinMaker, termasuk format ARNs untuk setiap jenis sumber daya, lihat [Kunci tindakan, sumber daya, dan kondisi untuk AWS IoT TwinMaker](#) dalam Referensi Otorisasi Layanan.

Topik

- [Praktik terbaik kebijakan](#)
- [Menggunakan konsol AWS IoT TwinMaker](#)
- [Mengizinkan pengguna melihat izin mereka sendiri](#)

Praktik terbaik kebijakan

Kebijakan berbasis identitas menentukan apakah seseorang dapat membuat, mengakses, atau menghapus AWS IoT TwinMaker sumber daya di akun Anda. Tindakan ini membuat Akun AWS Anda dikenai biaya. Ketika Anda membuat atau mengedit kebijakan berbasis identitas, ikuti panduan dan rekomendasi ini:

- Mulailah dengan kebijakan AWS terkelola dan beralih ke izin hak istimewa paling sedikit — Untuk mulai memberikan izin kepada pengguna dan beban kerja Anda, gunakan kebijakan AWS terkelola yang memberikan izin untuk banyak kasus penggunaan umum. Mereka tersedia di Anda Akun AWS. Kami menyarankan Anda mengurangi izin lebih lanjut dengan menentukan kebijakan yang dikelola AWS pelanggan yang khusus untuk kasus penggunaan Anda. Untuk informasi selengkapnya, lihat [kebijakan AWS terkelola](#) atau [kebijakan terkelola untuk fungsi pekerjaan](#) di Panduan IAM Pengguna.
- Menerapkan izin hak istimewa paling sedikit — Saat Anda menetapkan izin dengan IAM kebijakan, berikan hanya izin yang diperlukan untuk melakukan tugas. Anda melakukannya dengan

mendefinisikan tindakan yang dapat diambil pada sumber daya tertentu dalam kondisi tertentu, yang juga dikenal sebagai izin dengan hak akses paling rendah. Untuk informasi selengkapnya tentang penggunaan IAM untuk menerapkan izin, lihat [Kebijakan dan izin IAM di IAM](#) Panduan Pengguna.

- Gunakan ketentuan dalam IAM kebijakan untuk membatasi akses lebih lanjut — Anda dapat menambahkan kondisi ke kebijakan Anda untuk membatasi akses ke tindakan dan sumber daya. Misalnya, Anda dapat menulis kondisi kebijakan untuk menentukan bahwa semua permintaan harus dikirim menggunakan SSL. Anda juga dapat menggunakan ketentuan untuk memberikan akses ke tindakan layanan jika digunakan melalui yang spesifik Layanan AWS, seperti AWS CloudFormation. Untuk informasi selengkapnya, lihat [elemen IAM JSON kebijakan: Kondisi](#) dalam Panduan IAM Pengguna.
- Gunakan IAM Access Analyzer untuk memvalidasi IAM kebijakan Anda guna memastikan izin yang aman dan fungsional — IAM Access Analyzer memvalidasi kebijakan baru dan yang sudah ada sehingga kebijakan tersebut mematuhi bahasa IAM kebijakan () JSON dan praktik terbaik. IAM Access Analyzer menyediakan lebih dari 100 pemeriksaan kebijakan dan rekomendasi yang dapat ditindaklanjuti untuk membantu Anda membuat kebijakan yang aman dan fungsional. Untuk informasi selengkapnya, lihat [Validasi kebijakan IAM Access Analyzer](#) di IAMPanduan Pengguna.
- Memerlukan otentikasi multi-faktor (MFA) - Jika Anda memiliki skenario yang mengharuskan IAM pengguna atau pengguna root di dalam Anda Akun AWS, aktifkan MFA untuk keamanan tambahan. Untuk meminta MFA kapan API operasi dipanggil, tambahkan MFA kondisi ke kebijakan Anda. Untuk informasi selengkapnya, lihat [Mengonfigurasi API akses MFA yang dilindungi](#) di IAMPanduan Pengguna.

Untuk informasi selengkapnya tentang praktik terbaik di IAM, lihat [Praktik terbaik keamanan IAM di](#) Panduan IAM Pengguna.

Menggunakan konsol AWS IoT TwinMaker

Untuk mengakses AWS IoT TwinMaker konsol, Anda harus memiliki set izin minimum. Izin ini harus memungkinkan Anda untuk membuat daftar dan melihat detail tentang AWS IoT TwinMaker sumber daya di Anda Akun AWS. Jika Anda membuat kebijakan berbasis identitas yang lebih ketat daripada izin minimum yang diperlukan, konsol tidak akan berfungsi sebagaimana mestinya untuk entitas (pengguna atau peran) dengan kebijakan tersebut.

Anda tidak perlu mengizinkan izin konsol minimum untuk pengguna yang melakukan panggilan hanya ke AWS CLI atau AWS API. Sebagai gantinya, izinkan akses hanya ke tindakan yang cocok dengan API operasi yang mereka coba lakukan.

Untuk memastikan bahwa pengguna dan peran masih dapat menggunakan AWS IoT TwinMaker konsol, lampirkan juga kebijakan AWS IoT TwinMaker ConsoleAccess atau ReadOnly AWS terkelola ke entitas. Untuk informasi selengkapnya, lihat [Menambahkan izin ke pengguna](#) di Panduan AWS IAM Identity Center Pengguna.

Mengizinkan pengguna melihat izin mereka sendiri

Contoh ini menunjukkan cara Anda membuat kebijakan yang memungkinkan IAM pengguna melihat kebijakan sebaris dan terkelola yang dilampirkan pada identitas pengguna mereka. Kebijakan ini mencakup izin untuk menyelesaikan tindakan ini di konsol atau secara terprogram menggunakan atau. AWS CLI AWS API

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ViewOwnUserInfo",
      "Effect": "Allow",
      "Action": [
        "iam:GetUserPolicy",
        "iam:ListGroupForUser",
        "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": "*"
    }
  ]
}
```

```
]
}
```

Memecahkan masalah AWS IoT TwinMaker identitas dan akses

Gunakan informasi berikut untuk membantu Anda mendiagnosis dan memperbaiki masalah umum yang mungkin Anda temui saat bekerja dengan AWS IoT TwinMaker dan IAM.

Topik

- [Saya tidak berwenang untuk melakukan tindakan di AWS IoT TwinMaker](#)
- [Saya tidak berwenang untuk melakukan iam: PassRole](#)
- [Saya ingin mengizinkan orang di luar saya Akun AWS untuk mengakses AWS IoT TwinMaker sumber daya saya](#)

Saya tidak berwenang untuk melakukan tindakan di AWS IoT TwinMaker

Jika Anda menerima pesan kesalahan bahwa Anda tidak memiliki otorisasi untuk melakukan tindakan, kebijakan Anda harus diperbarui agar Anda dapat melakukan tindakan tersebut.

Contoh kesalahan berikut terjadi ketika mateojackson IAM pengguna mencoba menggunakan konsol untuk melihat detail tentang *my-example-widget* sumber daya fiksi tetapi tidak memiliki izin `iottwinmaker:GetWidget` fiksi.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
iottwinmaker:GetWidget on resource: my-example-widget
```

Dalam hal ini, kebijakan untuk pengguna mateojackson harus diperbarui untuk mengizinkan akses ke sumber daya *my-example-widget* dengan menggunakan tindakan `iottwinmaker:GetWidget`.

Jika Anda memerlukan bantuan, hubungi AWS administrator Anda. Administrator Anda adalah orang yang memberi Anda kredensial masuk.

Saya tidak berwenang untuk melakukan iam: PassRole

Jika Anda menerima kesalahan yang tidak diizinkan untuk melakukan `iam:PassRole` tindakan, kebijakan Anda harus diperbarui agar Anda dapat meneruskan peran AWS IoT TwinMaker.

Beberapa Layanan AWS memungkinkan Anda untuk meneruskan peran yang ada ke layanan tersebut alih-alih membuat peran layanan baru atau peran terkait layanan. Untuk melakukannya, Anda harus memiliki izin untuk meneruskan peran ke layanan.

Contoh kesalahan berikut terjadi ketika IAM pengguna bernama `marymajor` mencoba menggunakan konsol untuk melakukan tindakan di AWS IoT TwinMaker. Namun, tindakan tersebut memerlukan layanan untuk mendapatkan izin yang diberikan oleh peran layanan. Mary tidak memiliki izin untuk meneruskan peran tersebut pada layanan.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
iam:PassRole
```

Dalam kasus ini, kebijakan Mary harus diperbarui agar dia mendapatkan izin untuk melakukan tindakan `iam:PassRole` tersebut.

Jika Anda memerlukan bantuan, hubungi AWS administrator Anda. Administrator Anda adalah orang yang memberi Anda kredensial masuk.

Saya ingin mengizinkan orang di luar saya Akun AWS untuk mengakses AWS IoT TwinMaker sumber daya saya

Anda dapat membuat peran yang dapat digunakan pengguna di akun lain atau orang-orang di luar organisasi Anda untuk mengakses sumber daya Anda. Anda dapat menentukan siapa saja yang dipercaya untuk mengambil peran tersebut. Untuk layanan yang mendukung kebijakan berbasis sumber daya atau daftar kontrol akses (ACLs), Anda dapat menggunakan kebijakan tersebut untuk memberi orang akses ke sumber daya Anda.

Untuk mempelajari selengkapnya, periksa referensi berikut:

- Untuk mempelajari apakah AWS IoT TwinMaker mendukung fitur ini, lihat [Bagaimana AWS IoT TwinMaker bekerja dengan IAM](#).
- Untuk mempelajari cara menyediakan akses ke sumber daya Anda di seluruh sumber daya Akun AWS yang Anda miliki, lihat [Menyediakan akses ke IAM pengguna lain Akun AWS yang Anda miliki](#) di Panduan IAM Pengguna.
- Untuk mempelajari cara menyediakan akses ke sumber daya Anda kepada pihak ketiga Akun AWS, lihat [Menyediakan akses yang Akun AWS dimiliki oleh pihak ketiga](#) dalam Panduan IAM Pengguna.
- Untuk mempelajari cara menyediakan akses melalui federasi identitas, lihat [Menyediakan akses ke pengguna yang diautentikasi secara eksternal \(federasi identitas\) di Panduan Pengguna](#). IAM

- Untuk mempelajari perbedaan antara menggunakan peran dan kebijakan berbasis sumber daya untuk akses lintas akun, lihat [Akses sumber daya lintas akun di IAM](#) Panduan Pengguna. IAM

Menggunakan peran terkait layanan untuk AWS IoT TwinMaker

AWS IoT TwinMaker menggunakan AWS Identity and Access Management (IAM) peran [terkait layanan](#). Peran terkait layanan adalah jenis peran unik yang ditautkan langsung ke IAM AWS IoT TwinMaker. Peran terkait layanan telah ditentukan sebelumnya oleh AWS IoT TwinMaker dan mencakup semua izin yang diperlukan layanan untuk memanggil AWS layanan lain atas nama Anda.

Peran terkait layanan membuat pengaturan AWS IoT TwinMaker lebih mudah karena Anda tidak perlu menambahkan izin yang diperlukan secara manual. AWS IoT TwinMaker mendefinisikan izin peran terkait layanan, dan kecuali ditentukan lain, hanya AWS IoT TwinMaker dapat mengambil perannya. Izin yang ditetapkan mencakup kebijakan kepercayaan dan kebijakan izin, dan kebijakan izin tersebut tidak dapat dilampirkan ke entitas lain mana pun. IAM

Anda dapat menghapus peran tertaut layanan hanya setelah menghapus sumber daya terkait terlebih dahulu. Ini melindungi AWS IoT TwinMaker sumber daya Anda karena Anda tidak dapat secara tidak sengaja menghapus izin untuk mengakses sumber daya.

Untuk informasi tentang layanan lain yang mendukung peran terkait layanan, lihat [AWS layanan yang bekerja dengan IAM](#) dan cari layanan yang memiliki Ya di kolom Peran terkait layanan. Pilih Ya bersama tautan untuk melihat dokumentasi peran tertaut layanan untuk layanan tersebut.

Izin peran terkait layanan untuk AWS IoT TwinMaker

AWS IoT TwinMaker menggunakan peran terkait layanan bernama `AWSServiceRoleForIoT`— Memungkinkan AWS IoT TwinMaker untuk memanggil AWS layanan lain dan menyinkronkan sumber daya mereka atas nama Anda.

Peran `AWSServiceRoleForIoT` terkait layanan mempercayai layanan berikut untuk mengambil peran:

- `iottwinmaker.amazonaws.com`

Kebijakan izin peran bernama `AWSIoTServiceRolePolicy` memungkinkan AWS IoT TwinMaker untuk menyelesaikan tindakan berikut pada sumber daya yang ditentukan:

- Tindakan: `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` pada all your `iotsitewise` asset and `asset-model` resources

Anda harus mengonfigurasi izin agar pengguna, grup, atau peran Anda membuat, mengedit, atau menghapus peran terkait layanan. Untuk informasi selengkapnya, lihat [Izin peran terkait layanan di Panduan Pengguna IAM](#).

Membuat peran terkait layanan untuk AWS IoT TwinMaker

Anda tidak perlu membuat peran terkait layanan secara manual. Saat Anda menyinkronkan AWS IoT SiteWise aset dan model aset (sinkronisasi aset) di AWS Management Console, AWS CLI, atau AWS API, akan AWS IoT TwinMaker membuat peran terkait layanan untuk Anda.

Jika Anda menghapus peran terkait layanan ini, dan ingin membuatnya lagi, Anda dapat mengulangi proses yang sama untuk membuat kembali peran tersebut di akun Anda. Saat Anda menyinkronkan AWS IoT SiteWise aset dan model aset (sinkronisasi aset), AWS IoT TwinMaker buat kembali peran terkait layanan untuk Anda.

Anda juga dapat menggunakan IAM konsol untuk membuat peran terkait layanan dengan kasus penggunaan “IoT TwinMaker - Managed Role”. Di AWS CLI atau AWS API, buat peran terkait layanan dengan nama `iottwinmaker.amazonaws.com` layanan. Untuk informasi selengkapnya, lihat [Membuat peran terkait layanan](#) di IAMPanduan Pengguna. Jika Anda menghapus peran terkait layanan ini, Anda dapat mengulang proses yang sama untuk membuat peran tersebut lagi.

Mengedit peran terkait layanan untuk AWS IoT TwinMaker

AWS IoT TwinMaker tidak memungkinkan Anda untuk mengedit peran `AWSServiceRoleForIoT` TwinMaker terkait layanan. Setelah Anda membuat peran terkait layanan, Anda tidak dapat mengubah nama peran karena berbagai entitas mungkin mereferensikan peran tersebut. Namun, Anda dapat mengedit deskripsi peran menggunakan IAM. Untuk informasi selengkapnya, lihat [Mengedit peran terkait layanan](#) di IAMPanduan Pengguna.

Menghapus peran terkait layanan untuk AWS IoT TwinMaker

Jika Anda tidak perlu lagi menggunakan fitur atau layanan yang memerlukan peran terkait layanan, kami merekomendasikan Anda menghapus peran tersebut. Dengan begitu, Anda tidak memiliki entitas yang tidak digunakan yang tidak dipantau atau dipelihara secara aktif. Namun, Anda harus membersihkan serviceLinked ruang kerja apa pun yang masih menggunakan peran terkait layanan sebelum Anda dapat menghapus peran secara manual.

Note

Jika AWS IoT TwinMaker layanan menggunakan peran saat Anda mencoba menghapus sumber daya, maka penghapusan mungkin gagal. Jika hal itu terjadi, tunggu beberapa menit dan coba mengoperasikannya lagi.

Untuk menghapus peran terkait layanan secara manual menggunakan IAM

Gunakan IAM konsol, AWS CLI, atau AWS API untuk menghapus peran AWSServiceRoleForIoTTwinMaker terkait layanan. Untuk informasi selengkapnya, lihat [Menghapus peran terkait layanan di Panduan Pengguna](#). IAM

Wilayah yang Didukung untuk AWS IoT TwinMaker peran terkait layanan

AWS IoT TwinMaker mendukung penggunaan peran terkait layanan di semua Wilayah tempat layanan tersedia. Untuk informasi selengkapnya, lihat [AWS Wilayah dan titik akhir](#).

AWS kebijakan terkelola untuk AWS IoT TwinMaker

Untuk menambahkan izin ke pengguna, grup, dan peran, lebih mudah menggunakan kebijakan AWS terkelola daripada menulis kebijakan sendiri. Butuh waktu dan keahlian untuk [membuat kebijakan terkelola IAM pelanggan](#) yang hanya memberi tim Anda izin yang mereka butuhkan. Untuk memulai dengan cepat, Anda dapat menggunakan kebijakan AWS terkelola kami. Kebijakan ini mencakup kasus penggunaan umum dan tersedia di Akun AWS Anda. Untuk informasi selengkapnya tentang kebijakan AWS [AWS terkelola](#), lihat [kebijakan terkelola](#) di Panduan IAM Pengguna.

AWS layanan memelihara dan memperbarui kebijakan AWS terkelola. Anda tidak dapat mengubah izin dalam kebijakan AWS terkelola. Layanan terkadang menambahkan izin tambahan ke kebijakan

yang dikelola AWS untuk mendukung fitur-fitur baru. Jenis pembaruan ini akan memengaruhi semua identitas (pengguna, grup, dan peran) di mana kebijakan tersebut dilampirkan. Layanan kemungkinan besar akan memperbarui kebijakan yang dikelola AWS saat ada fitur baru yang diluncurkan atau saat ada operasi baru yang tersedia. Layanan tidak menghapus izin dari kebijakan AWS terkelola, sehingga pembaruan kebijakan tidak akan merusak izin yang ada.

Selain itu, AWS mendukung kebijakan terkelola untuk fungsi pekerjaan yang mencakup beberapa layanan. Misalnya, kebijakan `ReadOnlyAccess` AWS terkelola menyediakan akses hanya-baca ke semua AWS layanan dan sumber daya. Saat layanan meluncurkan fitur baru, AWS menambahkan izin hanya-baca untuk operasi dan sumber daya baru. Untuk daftar dan deskripsi kebijakan fungsi pekerjaan, lihat [kebijakan AWS terkelola untuk fungsi pekerjaan](#) di Panduan IAM Pengguna.

AWS kebijakan terkelola: `AWSIoTtwinMakerServiceRolePolicy`

Anda tidak dapat melampirkan `AWSIoTtwinMakerServiceRolePolicy` ke IAM entitas Anda. Kebijakan ini dilampirkan ke peran terkait layanan yang mengizinkan untuk melakukan tindakan atas nama Anda. Untuk informasi selengkapnya, lihat [Izin peran terkait layanan untuk AWS IoT TwinMaker](#).

Kebijakan izin peran bernama `AWSIoTtwinMakerServiceRolePolicy` memungkinkan AWS IoT TwinMaker untuk menyelesaikan tindakan berikut pada sumber daya yang ditentukan:

- Tindakan: `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` pada 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 pembaruan kebijakan AWS terkelola

Lihat detail tentang pembaruan kebijakan AWS terkelola sejak layanan ini mulai melacak perubahan ini. Untuk peringatan otomatis tentang perubahan pada halaman ini, berlangganan RSS feed di halaman Riwayat dokumen.

Perubahan	Deskripsi	Tanggal
AWSIoT TwinMakerServiceRolePolicy — Ditambahkan kebijakan	AWS IoT TwinMaker menambahkan kebijakan izin peran bernama <code>AWSIoT TwinMakerServiceRolePolicy</code> yang memungkinkan AWS IoT TwinMaker untuk menyelesaikan tindakan berikut pada sumber daya yang ditentukan: <ul style="list-style-type: none"> Tindakan: <code>iotsitwise:DescribeAsset,</code> 	17 November 2023

Perubahan	Deskripsi	Tanggal
	<p>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 pada all your iotsitewise asset and asset-model resources</p> <p>Untuk informasi selengkapnya, lihat Izin peran terkait layanan untuk AWS IoT TwinMaker.</p>	
mulai melacak perubahan	mulai melacak perubahan untuk kebijakan yang AWS dikelola.	Mei 11, 2022

AWS IoT TwinMaker dan VPC titik akhir antarmuka ()AWS PrivateLink

Anda dapat membuat koneksi pribadi antara virtual private cloud (VPC) dan AWS IoT TwinMaker dengan membuat VPC titik akhir antarmuka. Endpoint antarmuka didukung oleh [AWS PrivateLink](#), yang dapat Anda gunakan untuk mengakses secara pribadi AWS IoT TwinMaker APIs tanpa gateway internet, perangkat terjemahan alamat jaringan (NAT), VPN koneksi, atau koneksi Direct AWS Connect. Contoh di Anda VPC tidak memerlukan alamat IP publik untuk berkomunikasi AWS IoT TwinMaker APIs. Lalu lintas antara Anda VPC dan AWS IoT TwinMaker tidak meninggalkan jaringan Amazon.

Setiap titik akhir antarmuka diwakili oleh satu atau beberapa [Antarmuka Jaringan Elastis](#) di subnet Anda.

Untuk informasi selengkapnya, lihat [VPC Titik akhir antarmuka \(AWS PrivateLink\)](#) di Panduan VPC Pengguna Amazon.

Pertimbangan untuk titik akhir AWS IoT TwinMaker VPC

Sebelum menyiapkan VPC titik akhir antarmuka AWS IoT TwinMaker, tinjau [properti dan batasan titik akhir Antarmuka](#) di VPC Panduan Pengguna Amazon.

AWS IoT TwinMaker mendukung membuat panggilan ke semua API tindakannya dari Anda VPC.

- Untuk API operasi pesawat data, gunakan titik akhir berikut:

```
data.iottwinmaker.region.amazonaws.com
```

API Operasi pesawat data meliputi:

- [GetPropertyValue](#)
 - [GetPropertyValueHistory](#)
 - [BatchPutPropertyValues](#)
- Untuk API operasi bidang kontrol, gunakan titik akhir berikut:

```
api.iottwinmaker.region.amazonaws.com
```

API Operasi pesawat kontrol yang didukung meliputi:

- [CreateComponentType](#)
- [CreateEntity](#)
- [CreateScene](#)
- [CreateWorkspace](#)
- [DeleteComponentType](#)
- [DeleteEntity](#)
- [DeleteScene](#)
- [DeleteWorkspace](#)
- [GetComponentType](#)
- [GetEntity](#)
- [GetScene](#)
- [GetWorkspace](#)
- [ListComponentTypes](#)
- [ListComponentTypes](#)
- [ListEntities](#)
- [ListScenes](#)
- [ListTagsForResource](#)
- [ListWorkspaces](#)
- [TagResource](#)
- [UntagResource](#)
- [UpdateComponentType](#)
- [UpdateEntity](#)
- [UpdateScene](#)
- [UpdateWorkspace](#)

Membuat VPC titik akhir antarmuka untuk AWS IoT TwinMaker

Anda dapat membuat VPC titik akhir untuk AWS IoT TwinMaker layanan dengan menggunakan VPC konsol Amazon atau AWS Command Line Interface (AWS CLI). Untuk informasi selengkapnya, lihat [Membuat titik akhir antarmuka](#) di Panduan VPC Pengguna Amazon.

Buat VPC titik akhir untuk AWS IoT TwinMaker yang menggunakan nama layanan berikut.

- Untuk API operasi pesawat data, gunakan nama layanan berikut:

```
com.amazonaws.region.iottwinmaker.data
```

- Untuk API operasi pesawat kontrol, gunakan nama layanan berikut:

```
com.amazonaws.region.iottwinmaker.api
```

Jika Anda mengaktifkan privat DNS untuk titik akhir, Anda dapat membuat API permintaan AWS IoT TwinMaker dengan menggunakan DNS nama defaultnya untuk Wilayah, misalnya, `iottwinmaker.us-east-1.amazonaws.com`.

Untuk informasi selengkapnya, lihat [Mengakses layanan melalui titik akhir antarmuka](#) di VPC Panduan Pengguna Amazon.

AWS IoT TwinMaker PrivateLink didukung di wilayah berikut:

- us-east-1

ControlPlane Layanan ini didukung di zona ketersediaan berikut: `use1-az1`, `use1-az2`, dan `use1-az6`.

DataPlane Layanan ini didukung di zona ketersediaan berikut: `use1-az1`, `use1-az2`, dan `use1-az4`.

- us-west-2

DataPlane Layanan ControlPlane dan didukung di zona ketersediaan berikut: `usw2-az1`, `usw2-az2`, dan `usw2-az3`.

- eu-west-1
- eu-central-1
- ap-southeast-1
- ap-southeast-2

Untuk informasi selengkapnya tentang zona ketersediaan, lihat [Availability Zone IDs untuk AWS sumber daya Anda - AWS Resource Access Manager](#).

Mengakses AWS IoT TwinMaker melalui titik akhir antarmuka VPC

Saat Anda membuat titik akhir antarmuka, AWS IoT TwinMaker buat DNS nama host khusus titik akhir yang dapat Anda gunakan untuk berkomunikasi. AWS IoT TwinMaker DNSOps pribadi diaktifkan secara default. Untuk informasi selengkapnya, lihat [Menggunakan zona yang dihosting pribadi](#) di Panduan VPC Pengguna Amazon.

Jika Anda mengaktifkan private DNS untuk endpoint, Anda dapat membuat API permintaan AWS IoT TwinMaker melalui salah satu titik VPC akhir berikut.

- Untuk API operasi pesawat data, gunakan titik akhir berikut. Ganti *region* dengan AWS wilayah Anda.

```
data.iottwinmaker.region.amazonaws.com
```

- Untuk API operasi pesawat kontrol, gunakan titik akhir berikut. Ganti *region* dengan AWS wilayah Anda.

```
api.iottwinmaker.region.amazonaws.com
```

Jika Anda menonaktifkan pribadi DNS untuk titik akhir, Anda harus melakukan hal berikut untuk mengakses AWS IoT TwinMaker melalui titik akhir:

- Tentukan VPC titik akhir URL dalam API permintaan.
 - Untuk API operasi pesawat data, gunakan titik akhir URL berikut. Ganti *vpc-endpoint-id* dan *region* dengan ID VPC titik akhir dan Wilayah Anda.

```
vpc-endpoint-id.data.iottwinmaker.region.vpce.amazonaws.com
```

- Untuk API operasi pesawat kontrol, gunakan titik akhir URL berikut. Ganti *vpc-endpoint-id* dan *region* dengan ID VPC titik akhir dan Wilayah Anda.

```
vpc-endpoint-id.api.iottwinmaker.region.vpce.amazonaws.com
```

- Nonaktifkan injeksi awalan host. AWS CLI Dan tambahkan AWS SDKs titik akhir layanan dengan berbagai awalan host saat Anda memanggil setiap operasi. API Hal ini AWS SDKs menyebabkan AWS CLI dan menghasilkan tidak valid URLs AWS IoT TwinMaker ketika Anda menentukan titik akhir. VPC

⚠ Important

Anda tidak dapat menonaktifkan injeksi awalan host di AWS CLI atau AWS Tools for PowerShell. Ini berarti bahwa jika Anda telah menonaktifkan privatDNS, Anda tidak akan dapat menggunakan AWS CLI atau AWS Tools for PowerShell mengakses AWS IoT TwinMaker melalui VPC titik akhir. Jika Anda ingin menggunakan alat ini untuk mengakses AWS IoT TwinMaker melalui titik akhir, aktifkan pribadiDNS.

Untuk informasi selengkapnya tentang cara menonaktifkan injeksi awalan host di AWS SDKs, lihat bagian dokumentasi berikut untuk masing-masing 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](#)

Untuk informasi selengkapnya, lihat [Mengakses layanan melalui titik akhir antarmuka](#) di VPC Panduan Pengguna Amazon.

Membuat kebijakan VPC endpoint untuk AWS IoT TwinMaker

Anda dapat melampirkan kebijakan titik akhir ke VPC titik akhir yang mengontrol akses ke AWS IoT TwinMaker. Kebijakan titik akhir menentukan informasi berikut:

- Prinsipal yang dapat melakukan tindakan.
- Tindakan yang dapat dilakukan.
- Sumber daya yang menjadi target tindakan.

Untuk informasi selengkapnya, lihat [Mengontrol akses ke layanan dengan VPC titik akhir](#) di Panduan VPC Pengguna Amazon.

Contoh: kebijakan VPC endpoint untuk tindakan AWS IoT TwinMaker

Berikut ini adalah contoh kebijakan endpoint untuk AWS IoT TwinMaker. Saat dilampirkan ke titik akhir, kebijakan ini memberikan akses ke AWS IoT TwinMaker tindakan yang tercantum untuk IAM pengguna di AWS akun `iottwinmakeradmin 123456789012` di semua sumber daya.

```
{
  "Statement": [
    {
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:user/role"
      },
      "Resource": "*",
      "Effect": "Allow",
      "Action": [
        "iottwinmaker:CreateEntity",
        "iottwinmaker:GetScene",
        "iottwinmaker:ListEntities"
      ]
    }
  ]
}
```


Validasi Kepatuhan untuk AWS IoT TwinMaker

Untuk mempelajari apakah an Layanan AWS berada dalam lingkup program kepatuhan tertentu, lihat [Layanan AWS di Lingkup oleh Program Kepatuhan Layanan AWS](#) dan pilih program kepatuhan yang Anda minati. Untuk informasi umum, lihat [Program AWS Kepatuhan Program AWS](#) .

Anda dapat mengunduh laporan audit pihak ketiga menggunakan AWS Artifact. Untuk informasi selengkapnya, lihat [Mengunduh Laporan di AWS Artifact](#) .

Tanggung jawab kepatuhan Anda saat menggunakan Layanan AWS ditentukan oleh sensitivitas data Anda, tujuan kepatuhan perusahaan Anda, dan hukum dan peraturan yang berlaku. AWS menyediakan sumber daya berikut untuk membantu kepatuhan:

- [Panduan Memulai Cepat Keamanan dan Kepatuhan — Panduan](#) penerapan ini membahas pertimbangan arsitektur dan memberikan langkah-langkah untuk menerapkan lingkungan dasar AWS yang berfokus pada keamanan dan kepatuhan.
- [Arsitektur untuk HIPAA Keamanan dan Kepatuhan di Amazon Web Services](#) — Whitepaper ini menjelaskan bagaimana perusahaan dapat menggunakan AWS untuk membuat HIPAA aplikasi yang memenuhi syarat.

 Note

Tidak semua Layanan AWS HIPAA memenuhi syarat. Untuk informasi selengkapnya, lihat [Referensi Layanan yang HIPAA Memenuhi Syarat](#).

- [AWS Sumber Daya AWS](#) — Kumpulan buku kerja dan panduan ini mungkin berlaku untuk industri dan lokasi Anda.
- [AWS Panduan Kepatuhan Pelanggan](#) - Memahami model tanggung jawab bersama melalui lensa kepatuhan. Panduan ini merangkum praktik terbaik untuk mengamankan Layanan AWS dan memetakan panduan untuk kontrol keamanan di berbagai kerangka kerja (termasuk Institut Standar dan Teknologi Nasional (NIST), Dewan Standar Keamanan Industri Kartu Pembayaran (PCI), dan Organisasi Internasional untuk Standardisasi ()). ISO
- [Mengevaluasi Sumber Daya dengan Aturan](#) dalam Panduan AWS Config Pengembang — AWS Config Layanan menilai seberapa baik konfigurasi sumber daya Anda mematuhi praktik internal, pedoman industri, dan peraturan.
- [AWS Security Hub](#)— Ini Layanan AWS memberikan pandangan komprehensif tentang keadaan keamanan Anda di dalamnya AWS. Security Hub menggunakan kontrol keamanan untuk sumber daya AWS Anda serta untuk memeriksa kepatuhan Anda terhadap standar industri keamanan dan praktik terbaik. Untuk daftar layanan dan kontrol yang didukung, lihat [Referensi kontrol Security Hub](#).
- [Amazon GuardDuty](#) — Ini Layanan AWS mendeteksi potensi ancaman terhadap beban kerja Akun AWS, kontainer, dan data Anda dengan memantau lingkungan Anda untuk aktivitas yang mencurigakan dan berbahaya. GuardDuty dapat membantu Anda mengatasi berbagai persyaratan kepatuhan, seperti PCIDSS, dengan memenuhi persyaratan deteksi intrusi yang diamanatkan oleh kerangka kerja kepatuhan tertentu.
- [AWS Audit Manager](#)Ini Layanan AWS membantu Anda terus mengaudit AWS penggunaan Anda untuk menyederhanakan cara Anda mengelola risiko dan kepatuhan terhadap peraturan dan standar industri.

Ketahanan di AWS IoT TwinMaker

Infrastruktur AWS global dibangun di sekitar Wilayah AWS dan Availability Zones. Wilayah AWS menyediakan beberapa Availability Zone yang terpisah secara fisik dan terisolasi, yang terhubung dengan latensi rendah, throughput tinggi, dan jaringan yang sangat redundan. Dengan Zona Ketersediaan, Anda dapat merancang serta mengoperasikan aplikasi dan basis data yang secara otomatis melakukan fail over di antara zona tanpa gangguan. Zona Ketersediaan memiliki ketersediaan dan toleransi kesalahan yang lebih baik, dan dapat diskalakan dibandingkan infrastruktur pusat data tunggal atau multi tradisional.

Untuk informasi selengkapnya tentang Wilayah AWS dan Availability Zone, lihat [Infrastruktur AWS Global](#).

Selain infrastruktur AWS global, AWS IoT TwinMaker menawarkan beberapa fitur untuk membantu mendukung ketahanan data dan kebutuhan cadangan Anda.

Keamanan Infrastruktur di AWS IoT TwinMaker

Sebagai layanan terkelola, AWS IoT TwinMaker dilindungi oleh prosedur keamanan jaringan AWS global yang dijelaskan dalam whitepaper [Amazon Web Services: Tinjauan Proses Keamanan](#).

Anda menggunakan API panggilan yang AWS dipublikasikan untuk mengakses AWS IoT TwinMaker melalui jaringan. Klien harus mendukung Transport Layer Security (TLS) 1.2 atau yang lebih baru. Kami merekomendasikan TLS 1.3 atau yang lebih baru. Klien juga harus mendukung cipher suite dengan perfect forward secrecy (PFS) seperti Ephemeral Diffie-Hellman () atau Elliptic Curve Ephemeral Diffie-Hellman (). DHE ECDHE Sebagian besar sistem modern seperti Java 7 dan versi lebih baru mendukung mode-mode ini.

Selain itu, permintaan harus ditandatangani dengan menggunakan ID kunci akses dan kunci akses rahasia yang terkait dengan IAM prinsipal. Atau Anda bisa menggunakan [AWS Security Token Service](#) (AWS STS) untuk membuat kredensial keamanan sementara guna menandatangani permintaan.

Titik akhir dan kuota

AWS IoT TwinMaker titik akhir dan kuota

Anda dapat menemukan informasi tentang AWS IoT TwinMaker titik akhir dan kuota di Referensi [AWS Umum](#).

- Untuk informasi tentang titik akhir layanan, lihat titik [akhir AWS IoT TwinMaker layanan](#).
- Untuk informasi tentang kuota, lihat [kuota AWS IoT TwinMaker layanan](#).
- Untuk informasi tentang batas API pelambatan, lihat batas [AWS IoT TwinMaker API pelambatan](#).

Informasi tambahan tentang titik AWS IoT TwinMaker akhir

Untuk terhubung secara terprogram ke AWS IoT TwinMaker, gunakan titik akhir. Jika Anda menggunakan HTTP klien, Anda perlu awalan bidang kontrol dan bidang data APIs sebagai berikut. Namun, tidak perlu menambahkan awalan AWS SDK dan AWS Command Line Interface perintah karena mereka secara otomatis menambahkan awalan yang diperlukan.

- Gunakan api awalan untuk bidang APIs kontrol. Misalnya, `api.iottwinmaker.us-west-1.amazonaws.com`.
- Gunakan data awalan untuk bidang APIs data. Misalnya, `data.iottwinmaker.us-west-1.amazonaws.com`.

Riwayat dokumen untuk Panduan Pengguna AWS IoT TwinMaker

Tabel berikut menjelaskan rilis dokumentasi untuk AWS IoT TwinMaker.

Perubahan	Deskripsi	Tanggal
Peran terkait layanan baru dan kebijakan IAM baru	AWS IoT TwinMaker menambahkan peran terkait layanan baru, yang disebut AWSServiceRoleForIoT TwinMaker menambahkan peran terkait layanan baru ini untuk memungkinkan AWS IoT TwinMaker panggilan AWS layanan lain dan menyinkronkan sumber daya mereka atas nama Anda. Kebijakan <code>AWSIoT TwinMakerServiceRolePolicy</code> IAM baru dilampirkan pada peran ini, dan kebijakan tersebut memberikan izin untuk AWS IoT TwinMaker menelepon AWS layanan lain dan menyinkronkan sumber daya mereka atas nama Anda.	17 November 2023
Rilis awal	Rilis awal Panduan AWS IoT TwinMaker Pengguna	30 November 2021

Terjemahan disediakan oleh mesin penerjemah. Jika konten terjemahan yang diberikan bertentangan dengan versi bahasa Inggris aslinya, utamakan versi bahasa Inggris.