aws

User Guide

AWS IoT TwinMaker

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

AWS IoT TwinMaker: User Guide

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

Die Handelsmarken und Handelsaufmachung von Amazon dürfen nicht in einer Weise in Verbindung mit nicht von Amazon stammenden Produkten oder Services verwendet werden, durch die Kunden irregeführt werden könnten oder Amazon in schlechtem Licht dargestellt oder diskreditiert werden könnte. Alle anderen Handelsmarken, die nicht Eigentum von Amazon sind, gehören den jeweiligen Besitzern, die möglicherweise zu Amazon gehören oder nicht, mit Amazon verbunden sind oder von Amazon gesponsert werden.

Table of Contents

Was ist AWS IoT TwinMaker? 1
Funktionsweise 1
Wichtige Konzepte und Komponenten 2
Workspace 3
Modell aus Entität und Komponente 4
Visualisierung 6
Erste Schritte mit AWS IoT TwinMaker 8
Erstellen und verwalten Sie eine Servicerolle für AWS IoT TwinMaker 9
Vertrauen zuweisen 9
Amazon-S3-Berechtigungen 10
Weisen Sie einem bestimmten Amazon S3 S3-Bucket Berechtigungen zu 11
Berechtigungen für integrierte Konnektoren 12
Berechtigungen für einen Connector zu einer externen Datenquelle 16
Ändern Sie Ihre Workspace-IAM-Rolle, um den Athena-Datenkonnektor zu verwenden 17
Einen Workspace erstellen 19
Erstellen Sie Ihre erste Entität 21
Ein AWS Konto einrichten 25
Melde dich an für ein AWS-Konto 25
Erstellen eines Benutzers mit Administratorzugriff 26
Komponententypen verwenden und erstellen 28
Integrierte Komponententypen 28
Kernmerkmale der AWS IoT TwinMaker Komponententypen 29
Eigenschaftsdefinitionen erstellen 31
Erstellung von Funktionen 32
Beispiele für Komponententypen 32
Alarm (Zusammenfassung) 33
Timestream-Telemetrie 34
Alarm (erbt vom abstrakten Alarm) 35
Beispiele für Geräte 36
Massenoperationen 39
Wichtige Konzepte und Terminologie 39
AWS IoT TwinMaker metadataTransferJob Funktionalität 40
Durchführung von Massenimport- und Exportvorgängen 42
metadataTransferJob Voraussetzungen 42

IAM-Berechtigungen 42
Führen Sie einen Massenvorgang aus 46
Fehlerbehandlung 50
Metadaten-Vorlagen importieren 51
AWS IoT TwinMaker metadataTransferJob Beispiele 54
AWS IoT TwinMaker Auftragsschema für die Übertragung von Metadaten 56
Datenanschlüsse 73
Datenanschlüsse 73
Konnektor für den Schema-Initialisierer 74
DataReaderByEntity 75
DataReaderByComponentType 76
DataReader 78
AttributePropertyValueReaderByEntity 79
DataWriter 80
Beispiele 81
Athena tabellarischer Datenkonnektor 90
AWS IoT TwinMaker Voraussetzungen für den Athena-Datenkonnektor 90
Verwenden des Athena-Datenkonnektors 91
Verwenden der JSON-Referenz für den Tabellendatenkonnektor von Athena 95
Verwenden des Athena-Datenkonnektors 96
Visualisieren Sie Athena-Tabellendaten in Grafana 97
AWS IoT TwinMaker Konnektor für Zeitreihen-Daten 99
AWS IoT TwinMaker Voraussetzungen für Zeitreihen-Datenkonnektoren 100
Hintergrund des Datenkonnektors für Zeitreihen 100
Entwicklung eines Datenkonnektors für Zeitreihen 102
Verbesserung Ihres Datenkonnektors 111
Testen Sie Ihren Connector 112
Sicherheit 112
AWS IoT TwinMaker Ressourcen erstellen 112
Was kommt als Nächstes 114
AWS IoT TwinMaker Cookie Factory-Datenanschluss 114
AWS IoT TwinMaker Szenen erstellen 120
Bevor Sie Szenen erstellen 120
Optimieren Sie Ihre Ressourcen, bevor Sie sie importieren AWS IoT TwinMaker 120
Bewährte Methoden für die Leistung in AWS IoT TwinMaker 121
Weitere Informationen 121

Ressourcen hochladen in AWS IoT TwinMaker 122
Laden Sie Dateien mithilfe der Konsole in die Ressourcenbibliothek hoch 122
Erstelle deine Szenen 123
Verwenden Sie die 3D-Navigation AWS IoT TwinMaker in Ihren In-Szenen 124
Fügen Sie feste Kameras hinzu 126
Verbesserte Bearbeitung 127
Gezielte Platzierung von Szenenobjekten 127
Auswahl eines Untermodells 127
Bearbeiten Sie Entitäten in der Szenenhierarchie 128
Fügen Sie Anmerkungen zu Entitäten hinzu 129
Fügen Sie Überlagerungen zu Tags hinzu 134
Bearbeiten Sie Ihre Szenen 142
Fügen Sie Modelle hinzu 142
Fügen Sie Widgets hinzu 143
Hinzufügen von Tags 147
Optimieren Sie Ihr 3D-Modell 147
Verwenden von 3D-Kacheln in Ihrer Szene 147
Dynamische Szenen 150
Statische und dynamische Szenen 150
Typen und Entitäten von Szenenkomponenten 151
Dynamische Szenenkonzepte 152
AWS IoT TwinMaker Integration des App-Kits 153
AWS IoT TwinMaker Preismodus wechseln 154
Wissensgraph 156
AWS IoT TwinMaker Kernkonzepte des Wissensgraphen 156
Verwenden von Knowledge Graph 157
Generieren Sie ein Szenengraph 160
AWS IoT TwinMaker Voraussetzungen für Szenengraphen 161
Binden Sie 3D-Knoten in Ihrer Szene 161
Erstellen Sie eine Webanwendung 164
Wissensgraph Grafana-Panel 166
AWS IoT TwinMaker Voraussetzungen für den Abfrage-Editor 166
Wissensgraph Grafana-Berechtigungen 166
Knowledge Graph zusätzliche Ressourcen 171
Asset-Synchronisierung mit AWS IoT SiteWise 184
Verwenden Sie die Asset-Synchronisierung mit AWS IoT SiteWise 184

Verwenden eines benutzerdefinierten Arbeitsbereichs ... 185
Verwenden des Io TSite WiseDefaultWorkspace ... 190
Unterschiede zwischen benutzerdefinierten und Standardarbeitsbereichen ... 191
Ressourcen, die synchronisiert wurden von AWS IoT SiteWise ... 192
Benutzerdefinierte und Standardarbeitsbereiche ... 192
Nur Standard-Workspace ... 193
Ressourcen wurden nicht synchronisiert ... 194
Verwenden Sie synchronisierte Entitäten und Komponententypen in AWS IoT TwinMaker ... 194
Analysieren Sie den Synchronisierungsstatus und die Fehler ... 195
Synchronisieren Sie den Jobstatus ... 195
Löschen Sie einen Synchronisierungsauftrag ... 197
Beschränkungen der Asset-Synchronisierung ... 199
Grafana-Dashboards einrichten ... 201
CORS-Konfiguration ... 202
Einrichtung Ihrer Grafana-Umgebung ... 203
Amazon Managed Grafana ... 203
Selbstverwaltetes Grafana ... 204
Eine Dashboard-Rolle erstellen ... 205
Eine IAM-Richtlinie erstellen ... 205
Laden Sie ein Video vom Edge hoch ... 209
Fügen Sie weitere Berechtigungen hinzu ... 209
Die Grafana-Dashboard-IAM-Rolle erstellen ... 211
Eine AWS IoT TwinMaker Video-Player-Richtlinie erstellen ... 212
Beschränken Sie den Zugriff auf Ihre Ressourcen ... 213
Schränke GET-Berechtigungen nach unten ... 214
Erlaubnisbereich nach unten AWS IoT SiteWise BatchPutAssetPropertyValue ... 215
Alarme mit Grafana-Dashboards verbinden ... 218
AWS IoT SiteWise Voraussetzungen für die Alarm-Konfiguration ... 218
Definieren Sie die AWS IoT SiteWise IAM-Rolle der Alarmkomponente ... 219
Abfragen und Aktualisierungen über die AWS IoT TwinMaker API ... 220
Konfigurieren Sie Ihr Grafana-Dashboard für Alarme ... 222
Verwenden Sie das Grafana-Dashboard zur Alarmvisualisierung ... 224
Matterport-Integration ... 227
Überblick über die Integration ... 228
Voraussetzungen für die Matterport-Integration ... 229
Matterport SDK-Anmeldeinformationen ... 231

Speichern Sie die Matterport-Anmeldeinformationen in AWS Secrets Manager 232
Matterport-Scans in Szenen AWS IoT TwinMaker 235
Matterport in deinem AWS IoT TwinMaker Grafana-Dashboard 241
Matterport-Integration mit dem App-Kit AWS IoT 241
Video streamen zu AWS IoT TwinMaker 243
Verwenden Sie den Edge-Anschluss für Kinesis-Videostream, um Videos einzuspielen AWS IoT TwinMaker 243
Voraussetzungen 243
Erstellen Sie Videokomponenten für Szenen AWS IoT TwinMaker 244
Fügen Sie Video und Metadaten aus dem Kinesis-Videostream zu einem Grafana-Dashboard hinzu 245
Verwendung der AWS IoT TwinMaker Flink-Bibliothek 247
Protokollierung und Überwachung 248
Überwachung mit CloudWatch Amazon-Metriken 249
Metriken 249
Protokollierung von API-Aufrufen mit AWS CloudTrail 252
AWS IoT TwinMaker Informationen in CloudTrail 253
Sicherheit 255
Datenschutz 256
Verschlüsselung im Ruhezustand 257
Verschlüsselung während der Übertragung 257
Identitäts- und Zugriffsverwaltung 257
Zielgruppe 258
Authentifizierung mit Identitäten 259
Verwalten des Zugriffs mit Richtlinien 263
Wie AWS IoT TwinMaker funktioniert mit IAM 266
Beispiele für identitätsbasierte Richtlinien 273
Fehlerbehebung 276
Verwenden von serviceverknüpften Rollen 278
AWS verwaltete Richtlinien 281
VPC-Endpunkte (AWS PrivateLink) 285
Überlegungen zu AWS IoT TwinMaker VPC-Endpunkten 286
Erstellen eines Schnittstellen-VPC-Endpunkts für AWS IoT TwinMaker 287
Zugriff AWS IoT TwinMaker über eine Schnittstelle (VPC-Endpunkt) 288
Erstellen einer VPC-Endpunktrichtlinie für AWS IoT TwinMaker 290
Compliance-Validierung 291

Ausfallsicherheit 292
Sicherheit der Infrastruktur 293
Endpunkte und Kontingente 294
AWS IoT TwinMaker Endpunkte und Kontingente 294
Zusätzliche Endpunktinformationen 294
Dokumentverlauf 295
..... ccxcvi

Was ist AWS IoT TwinMaker?

AWS IoT TwinMaker ist ein AWS IoT Service, mit dem Sie betriebsbereite digitale Zwillinge aus physischen und digitalen Systemen erstellen können. AWS IoT TwinMaker erstellt digitale Visualisierungen mithilfe von Messungen und Analysen aus einer Vielzahl von realen Sensoren, Kameras und Unternehmensanwendungen, damit Sie den Überblick über Ihre physische Fabrik, Ihr Gebäude oder Ihre Industrieanlage behalten. Sie können diese realen Daten verwenden, um den Betrieb zu überwachen, Fehler zu diagnostizieren und zu korrigieren und den Betrieb zu optimieren.

Ein digitaler Zwilling ist eine digitale Live-Repräsentation eines Systems und all seiner physischen und digitalen Komponenten. Er wird dynamisch mit Daten aktualisiert, um die tatsächliche Struktur, den Zustand und das Verhalten des Systems nachzuahmen. Sie können es verwenden, um Geschäftsergebnisse zu erzielen.

Endbenutzer interagieren mithilfe einer Benutzeroberflächenanwendung mit Daten aus Ihrem digitalen Zwilling.

Funktionsweise

Um die Mindestanforderungen für die Erstellung eines digitalen Zwillings zu erfüllen, müssen Sie wie folgt vorgehen.

- Modellieren Sie Geräte, Geräte, Räume und Prozesse an einem physischen Ort.
- Connect diese Modelle mit Datenquellen, die wichtige Kontextinformationen speichern, wie z. B. Kamera-Feeds mit Sensordaten.
- Erstellen Sie Visualisierungen, die Benutzern helfen, die Daten und Erkenntnisse zu verstehen, damit sie Geschäftsentscheidungen effizienter treffen können.
- Stellen Sie Endbenutzern digitale Zwillinge zur Verfügung, um Geschäftsergebnisse zu erzielen.

AWS IoT TwinMaker begegnet diesen Herausforderungen durch die Bereitstellung der folgenden Funktionen.

- Wissensgraph für Entitätskomponentensysteme: AWS IoT TwinMaker bietet Tools zur Modellierung von Geräten, Geräten, Räumen und Prozessen in einem Wissensgraphen.

  Dieser Wissensgraph enthält Metadaten über das System und kann eine Verbindung zu Daten an verschiedenen Orten herstellen. AWS IoT TwinMaker bietet integrierte Anschlüsse für in AWS

IoT SiteWise Kinesis Video Streams gespeicherte Daten. Sie können auch benutzerdefinierte Konnektoren für Daten erstellen, die an anderen Speicherorten gespeichert sind.

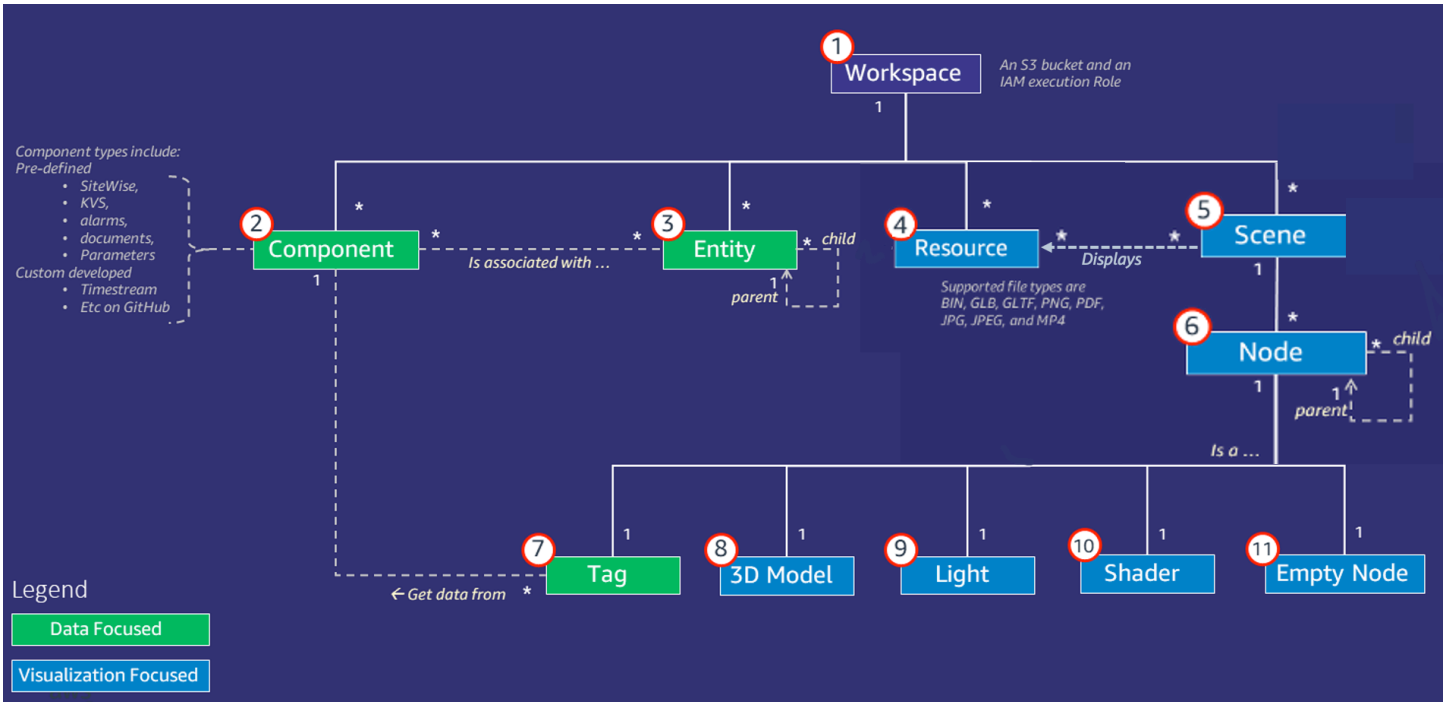
Der Knowledge Graph und die Konnektoren bilden zusammen eine einzige Schnittstelle für die Abfrage von Daten an unterschiedlichen Orten.

- Scene Composer: Die AWS IoT TwinMaker Konsole bietet ein Tool zur Szenenkomposition zum Erstellen von Szenen in 3D. Sie laden Ihre zuvor erstellten 3D-/CAD-Modelle hoch, die für die Webanzeige optimiert und in das .gltf- oder .glb-Format konvertiert wurden. Anschließend verwenden Sie den Scene Composer, um mehrere Modelle in einer einzigen Szene zu platzieren und so visuelle Darstellungen ihrer Funktionen zu erstellen.

  Sie können auch Daten in der Szene überlagern. Sie können beispielsweise ein Tag an einer Szenenposition erstellen, das eine Verbindung zu Temperaturdaten eines Sensors herstellt. Dadurch werden die Daten dem Standort zugeordnet.
- Anwendungen: AWS IoT TwinMaker bietet ein Plug-in für Grafana und Amazon Managed Grafana, mit dem Sie Dashboard-Anwendungen für Endbenutzer erstellen können.
- Tools von Drittanbietern: Mendix arbeitet mit ihnen zusammen AWS IoT TwinMaker , um Komplettlösungen für das industrielle IoT anzubieten. Sehen Sie sich den Workshop Lean Daily Management Application with Mendix AWS IoT TwinMaker an und erfahren Sie, wie Sie die Mendix Low Code Application Development Platform (LCAP) mit AWS Diensten wie AWS IoT TwinMaker Kinesis Video Streams und nutzen können. AWS IoT SiteWise

Wichtige Konzepte und Komponenten

Das folgende Diagramm zeigt, wie die Schlüsselkonzepte AWS IoT TwinMaker zusammenpassen.



> **Note**
>
> Sternchen (*) im Diagramm weisen auf Beziehungen hin. one-to-many Die Kontingente für jede dieser Beziehungen finden Sie unter AWS IoT TwinMaker Endpunkte und Kontingente.

In den folgenden Abschnitten werden die im Diagramm dargestellten Konzepte beschrieben.

Workspace

Ein Workspace ist ein Container auf oberster Ebene für Ihre digitale Zwillingsanwendung. In diesem Arbeitsbereich erstellen Sie einen logischen Satz von Entitäten, Komponenten, Szenenobjekten und anderen Ressourcen für Ihren digitalen Zwilling. Es dient auch als Sicherheitsgrenze für die Verwaltung des Zugriffs auf die digitale Zwillingsanwendung und die darin enthaltenen Ressourcen. Jeder Workspace ist mit dem Amazon S3 S3-Bucket verknüpft, in dem Ihre Workspace-Daten gespeichert sind. Sie verwenden IAM-Rollen, um den Zugriff auf Ihren Workspace einzuschränken.

Ein Workspace kann mehrere Komponenten, Entitäten, Szenen und Ressourcen enthalten. Ein Komponententyp, eine Entität, eine Szene oder eine Ressource existiert nur innerhalb eines Workspace.

Modell aus Entität und Komponente

AWS IoT TwinMaker stellt Tools bereit, mit denen Sie Ihr System mithilfe eines entity-component-based Wissensgraphen modellieren können. Sie können die Entity-Component-Architektur verwenden, um eine Darstellung Ihres physischen Systems zu erstellen. Dieses Entitätskomponentenmodell besteht aus Entitäten, Komponenten und Beziehungen. Weitere Informationen zu Entitätskomponentensystemen finden Sie unter Entitätskomponentensystem.

Entität

Entitäten sind digitale Repräsentationen der Elemente in einem digitalen Zwilling, die die Fähigkeiten dieses Elements erfassen. Dieses Element kann ein physisches Gerät, ein Konzept oder ein Prozess sein. Entitäten sind Komponenten zugeordnet. Diese Komponenten stellen Daten und Kontext für die zugehörige Entität bereit.

Mit AWS IoT TwinMaker können Sie Entitäten für eine effizientere Verwaltung in benutzerdefinierten Hierarchien organisieren. Die Standardansicht des Entitäts- und Komponentensystems ist hierarchisch.

Komponente

Komponenten stellen Kontext und Daten für Entitäten in einer Szene bereit. Sie fügen Komponenten zu Entitäten hinzu. Die Lebensdauer einer Komponente ist an die Lebensdauer einer Entität gebunden.

Komponenten können statische Daten hinzufügen, z. B. eine Liste von Dokumenten oder die Koordinaten eines geografischen Standorts. Sie können auch Funktionen haben, die eine Verbindung zu anderen Systemen herstellen, einschließlich Systemen, die Zeitreihendaten enthalten, wie AWS IoT SiteWise z. B. Cloud-Historikern für Zeitreihen.

Komponenten werden durch JSON-Dokumente definiert, die die Verbindung zwischen einer Datenquelle und AWS IoT TwinMaker beschreiben. Komponenten können externe Datenquellen oder Datenquellen beschreiben, in die integriert sind AWS IoT TwinMaker. Eine Komponente greift mithilfe einer Lambda-Funktion, die im JSON-Dokument angegeben ist, auf eine externe Datenquelle zu. Ein Workspace kann viele Komponenten enthalten. Komponenten stellen Daten über zugehörige Entitäten für Tags bereit.

AWS IoT TwinMaker stellt mehrere integrierte Komponenten bereit, die Sie über die Konsole hinzufügen können. Sie können auch Ihre eigenen benutzerdefinierten Komponenten erstellen, um

eine Verbindung zu Datenquellen wie Timestream-Telemetrie und Geokoordinaten herzustellen. Beispiele hierfür sind TimeStream Telemetrie, Geodatenkomponenten und Konnektoren zu Datenquellen von Drittanbietern wie Snowflake.

AWS IoT TwinMaker bietet die folgenden Typen von integrierten Komponenten für allgemeine Anwendungsfälle:

- Dokument, wie z. B. Benutzerhandbücher oder Bilder, befinden sich unter der angegebenen Adresse URLs.
- Zeitreihen, z. B. Sensordaten von AWS IoT SiteWise.
- Alarme, z. B. Zeitreihenalarme aus externen Datenquellen.
- Video von IP-Kameras, die mit Kinesis Video Streams verbunden sind.
- Benutzerdefinierte Komponenten für die Verbindung mit zusätzlichen Datenquellen. Sie können beispielsweise einen benutzerdefinierten Konnektor erstellen, um Ihre AWS IoT TwinMaker Entitäten mit extern gespeicherten Zeitreihendaten zu verbinden.

Datenquellen

Eine Datenquelle ist der Speicherort der Quelldaten Ihres digitalen Zwillings. AWS IoT TwinMaker unterstützt zwei Arten von Datenquellen:

- Hierarchiekonnektoren, mit denen Sie ein externes Modell kontinuierlich synchronisieren können AWS IoT TwinMaker.
- Zeitreihen-Konnektoren, mit denen Sie eine Verbindung zu Zeitreihendatenbanken herstellen können, wie z. AWS IoT SiteWise

Eigenschaft

Eigenschaften sind statische und zeitreihengestützte Werte, die in Komponenten enthalten sind. Wenn Sie Komponenten zu Entitäten hinzufügen, beschreiben die Eigenschaften in der Komponente Details zum aktuellen Status der Entität.

AWS IoT TwinMaker unterstützt drei Arten von Eigenschaften:

- Einzelwert, non-time-series Eigenschaften — AWS IoT TwinMaker Bei diesen Eigenschaften handelt es sich in der Regel um statische Schlüssel-Wert-Paare, die direkt in den Metadaten der zugehörigen Entität gespeichert werden.

- Zeitreiheneigenschaften — AWS IoT TwinMaker speichert einen Verweis auf den Zeitreihenspeicher für diese Eigenschaften. Dies ist standardmäßig der neueste Wert.
- Beziehungseigenschaften — Diese Eigenschaften speichern einen Verweis auf eine andere Entität oder Komponente. `seen_by`Ist beispielsweise eine Beziehungskomponente, die eine Kamera-Entität mit einer anderen Entität in Beziehung setzen kann, die direkt von dieser Kamera visualisiert wird.

Mithilfe der einheitlichen Datenabfrageschnittstelle können Sie Eigenschaftswerte aus heterogenen Datenquellen abfragen.

Visualisierung

Sie erweitern AWS IoT TwinMaker damit eine dreidimensionale Darstellung Ihres digitalen Zwillings und betrachten ihn dann in Grafana. Verwenden Sie zum Erstellen von Szenen vorhandene CAD- oder andere 3D-Dateitypen. Anschließend verwenden Sie Datenüberlagerungen, um relevante Daten für Ihren digitalen Zwilling hinzuzufügen.

Szenen

Szenen sind dreidimensionale Darstellungen, die den visuellen Kontext für die damit verbundenen Daten bieten. AWS IoT TwinMaker Szenen können mit einem einzigen GLTF- (GL Transmission Format) - oder GLB-3D-Modell für die gesamte Umgebung oder mithilfe einer Komposition aus mehreren Modellen erstellt werden. Szenen enthalten auch Tags, um interessante Punkte in der Szene zu kennzeichnen.

Szenen sind die Container der obersten Ebene für Visualisierungen. Eine Szene besteht aus einem oder mehreren Knoten.

Ein Workspace kann mehrere Szenen enthalten. Ein Workspace kann beispielsweise eine Szene für jede Etage einer Einrichtung enthalten.

Ressourcen

In Szenen werden Ressourcen angezeigt, die in der AWS IoT TwinMaker Konsole als Knoten angezeigt werden. Eine Szene kann viele Ressourcen enthalten.

Ressourcen sind Bilder und `glTF` basierte, dreidimensionale Modelle, die zur Erstellung einer Szene verwendet werden. Eine Ressource kann ein einzelnes Gerät oder eine komplette Site darstellen.

Sie platzieren Ressourcen in einer Szene, indem Sie eine .gltf- oder .glb-Datei in Ihre Workspace-Ressourcenbibliothek hochladen und sie dann zu Ihrer Szene hinzufügen.

Erweiterte Benutzeroberfläche

Mit können AWS IoT TwinMaker Sie Ihre Szenen mit Daten-Overlays erweitern, die wichtigen Kontext und Informationen wie Sensordaten zu Orten in der Szene hinzufügen.

Knoten: Knoten sind Instanzen von Tags, Lichtern und dreidimensionalen Modellen. Sie können auch leer sein, um Ihrer Szenenhierarchie Struktur zu verleihen. Sie können beispielsweise mehrere Knoten unter einem einzigen leeren Knoten gruppieren.

Tags: Ein Tag ist ein Knotentyp, der Daten aus einer Komponente (über eine Entität) darstellt. Ein Tag kann nur einer Komponente zugeordnet werden. Ein Tag ist eine Anmerkung, die zu einer bestimmten `x,y,z` Koordinatenposition einer Szene hinzugefügt wird. Das Tag verbindet diesen Szenenteil mithilfe einer Entitätseigenschaft mit dem Knowledge Graph. Sie können ein Tag verwenden, um das Verhalten oder das visuelle Erscheinungsbild eines Elements in der Szene zu konfigurieren, z. B. eines Alarms.

Lichter: Sie können einer Szene Lichter hinzufügen, um bestimmte Objekte in den Fokus zu rücken, oder Schatten auf Objekte werfen, um deren physische Position anzuzeigen.

Dreidimensionale Modelle: Ein dreidimensionales Modell ist eine visuelle Darstellung einer als Ressource importierten .gltf- oder .glb-Datei.

> **Note**
>
> AWS IoT TwinMaker ist nicht für den Einsatz in oder in Verbindung mit dem Betrieb gefährlicher Umgebungen oder kritischer Systeme vorgesehen, die zu schweren Körperverletzungen oder zum Tod führen oder Umwelt- oder Sachschäden verursachen können.
> Die durch Ihre Nutzung von gesammelten Daten AWS IoT TwinMaker sollten entsprechend Ihrem Anwendungsfall auf ihre Richtigkeit hin überprüft werden. AWS IoT TwinMaker sollte nicht als Ersatz für die menschliche Überwachung physischer Systeme verwendet werden, um zu beurteilen, ob diese Systeme sicher funktionieren.

Erste Schritte mit AWS IoT TwinMaker

In den Themen in diesem Abschnitt wird beschrieben, wie Sie die folgenden Schritte ausführen können.

- Erstellen und richten Sie einen neuen Workspace ein.
- Erstellen Sie eine Entität und fügen Sie ihr eine Komponente hinzu.

Voraussetzungen:

Um Ihren ersten Workspace und Ihre erste Szene zu erstellen, benötigen Sie die folgenden AWS Ressourcen.

- Ein AWS -Konto.
- Eine IAM-Servicerolle für AWS IoT TwinMaker. Diese Rolle wird standardmäßig automatisch generiert, wenn Sie einen neuen AWS IoT TwinMaker Workspace in der AWS IoT TwinMaker Konsole erstellen.

  Wenn Sie sich nicht dafür entscheiden, AWS IoT TwinMaker automatisch eine neue IAM-Servicerolle erstellen zu lassen, müssen Sie eine angeben, die Sie bereits erstellt haben.

  Anweisungen zum Erstellen und Verwalten dieser Servicerolle finden Sie unter???.

  Weitere Informationen zu IAM-Dienstrollen finden Sie unter Eine Rolle erstellen, um Berechtigungen an eine zu delegieren. AWS-Service

  > **Important**
  >
  > Dieser Servicerolle muss eine Richtlinie beigefügt sein, die dem Service Lese- und Schreibberechtigungen für einen Amazon S3 S3-Bucket erteilt. AWS IoT TwinMaker verwendet diese Rolle, um in Ihrem Namen auf andere Dienste zuzugreifen. Außerdem müssen Sie dieser Rolle eine Vertrauensstellung zuweisen, AWS IoT TwinMaker damit der Dienst die Rolle übernehmen kann. Wenn Ihr Zwilling mit anderen AWS Diensten interagiert, fügen Sie auch die erforderlichen Berechtigungen für diese Dienste hinzu.

Themen

- Erstellen und verwalten Sie eine Servicerolle für AWS IoT TwinMaker
- Einen Workspace erstellen
- Erstellen Sie Ihre erste Entität
- Ein AWS Konto einrichten

Erstellen und verwalten Sie eine Servicerolle für AWS IoT TwinMaker

AWS IoT TwinMaker erfordert, dass Sie eine Servicerolle verwenden, damit sie in Ihrem Namen auf Ressourcen in anderen Diensten zugreifen kann. Diese Rolle muss ein Vertrauensverhältnis mit haben AWS IoT TwinMaker. Wenn Sie einen Workspace erstellen, müssen Sie dem Workspace diese Rolle zuweisen. Dieses Thema enthält Beispielrichtlinien, die Ihnen zeigen, wie Sie Berechtigungen für allgemeine Szenarien konfigurieren.

Vertrauen zuweisen

Die folgende Richtlinie stellt eine Vertrauensbeziehung zwischen Ihrer Rolle und her AWS IoT TwinMaker. Ordnen Sie diese Vertrauensbeziehung der Rolle zu, die Sie für Ihren Workspace verwenden.

JSON

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

Amazon-S3-Berechtigungen

Die folgende Richtlinie ermöglicht es Ihrer Rolle, aus einem Amazon S3-Bucket zu lesen, zu löschen und in einen Amazon S3 S3-Bucket zu schreiben. Workspaces speichern Ressourcen in Amazon S3, sodass die Amazon S3 S3-Berechtigungen für alle Workspaces erforderlich sind.

JSON

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

> **Note**
>
> Wenn Sie einen Workspace erstellen, AWS IoT TwinMaker erstellt eine Datei in Ihrem Amazon S3 S3-Bucket, die angibt, dass er von einem Workspace verwendet wird. Diese Richtlinie gibt Ihnen die AWS IoT TwinMaker Erlaubnis, diese Datei zu löschen, wenn Sie den Workspace löschen.
> AWS IoT TwinMaker platziert andere Objekte, die sich auf Ihren Workspace beziehen. Es liegt in Ihrer Verantwortung, diese Objekte zu löschen, wenn Sie einen Workspace löschen.

Weisen Sie einem bestimmten Amazon S3 S3-Bucket Berechtigungen zu

Wenn Sie in der AWS IoT TwinMaker Konsole einen Workspace erstellen, können Sie wählen, ob Sie einen Amazon S3 S3-Bucket für Sie AWS IoT TwinMaker erstellen möchten. Sie können Informationen zu diesem Bucket finden, indem Sie den folgenden AWS CLI Befehl verwenden.

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

Das folgende Beispiel zeigt das Format der Ausgabe dieses Befehls.

```
{
    "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"
}
```

Um Ihre Richtlinie so zu aktualisieren, dass sie Berechtigungen für einen bestimmten Amazon S3 S3-Bucket zuweist, verwenden Sie den Wert von*bucket name*.

Die folgende Richtlinie ermöglicht es Ihrer Rolle, aus einem bestimmten Amazon S3-Bucket zu lesen, zu löschen und in einen bestimmten Amazon S3 S3-Bucket zu schreiben.

JSON

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

Berechtigungen für integrierte Konnektoren

Wenn dein Workspace mithilfe integrierter Konnektoren mit anderen AWS Diensten interagiert, musst du Berechtigungen für diese Dienste in diese Richtlinie aufnehmen. Wenn du den Komponententyp com.amazon.iotsitewise.connector verwendest, musst du Berechtigungen für angeben. AWS IoT SiteWise Weitere Informationen zu Komponententypen finden Sie unter. ???

> **Note**
>
> Wenn Sie mithilfe eines benutzerdefinierten Komponententyps mit anderen AWS Diensten interagieren, müssen Sie der Rolle die Berechtigung zum Ausführen der Lambda-Funktion erteilen, die die Funktion in Ihrem Komponententyp implementiert. Weitere Informationen finden Sie unter ???.

Das folgende Beispiel zeigt, wie Sie das AWS IoT SiteWise in Ihre Richtlinie aufnehmen können.

JSON

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

```
        "Resource": "asset model ARN"
        },
    {
      "Effect": "Allow",
      "Action": [
        "s3:DeleteObject"
      ],
      "Resource": [
        "arn:aws:s3:::*/DO_NOT_DELETE_WORKSPACE_*"
      ]
    }
  ]
}
```

Wenn Sie den Komponententyp com.amazon.iotsitewise.connector verwenden und Immobiliendaten aus lesen müssen AWS IoT SiteWise, müssen Sie die folgende Berechtigung in Ihre Richtlinie aufnehmen.

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

Wenn Sie den Komponententyp com.amazon.iotsitewise.connector verwenden und dafür Eigenschaftsdaten schreiben müssen, müssen Sie die folgende Berechtigung in Ihre Richtlinie aufnehmen. AWS IoT SiteWise

```
...
{
    "Action": [
```

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

Wenn Sie den Komponententyp com.amazon.iotsitewise.connector.edgevideo verwenden, müssen Sie Berechtigungen für und Kinesis Video Streams angeben. AWS IoT SiteWise Die folgende Beispielrichtlinie zeigt, wie Sie Kinesis Video Streams Streams-Berechtigungen in Ihre Richtlinie aufnehmen AWS IoT SiteWise können.

```
...
{
    "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"
},
...
```

Berechtigungen für einen Connector zu einer externen Datenquelle

Wenn Sie einen Komponententyp erstellen, der eine Funktion verwendet, die eine Verbindung zu einer externen Datenquelle herstellt, müssen Sie Ihrer Servicerolle die Erlaubnis erteilen, die Lambda-Funktion zu verwenden, die die Funktion implementiert. Weitere Informationen zum Erstellen von Komponententypen und Funktionen finden Sie unter???.

Das folgende Beispiel erteilt Ihrer Servicerolle die Erlaubnis, eine Lambda-Funktion zu verwenden.

JSON

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

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

Weitere Informationen zum Erstellen von Rollen und zum Zuweisen von Richtlinien und Vertrauensbeziehungen zu diesen mithilfe der IAM-Konsole, der und der AWS CLI IAM-API finden Sie unter Eine Rolle erstellen, um Berechtigungen an eine zu delegieren. AWS-Service

Ändern Sie Ihre Workspace-IAM-Rolle, um den Athena-Datenkonnektor zu verwenden

Um den AWS IoT TwinMaker Athena Table Data Connector zu verwenden, müssen Sie Ihre AWS IoT TwinMaker Workspace-IAM-Rolle aktualisieren. Fügen Sie Ihrer Workspace-IAM-Rolle die folgenden Berechtigungen hinzu:

> **Note**
>
> Diese IAM-Änderung funktioniert nur für Athena-Tabellendaten, die in Amazon S3 gespeichert sind. AWS Glue Um Athena mit anderen Datenquellen zu verwenden, müssen Sie eine IAM-Rolle für Athena konfigurieren, siehe Identitäts- und Zugriffsmanagement in Athena.

```
{
    "Effect": "Allow",
    "Action": [
        "athena:GetQueryExecution",
        "athena:GetQueryResults",
```

```
        "athena:GetTableMetadata",
        "athena:GetWorkGroup",
        "athena:StartQueryExecution",
        "athena:StopQueryExecution"
    ],
    "Resource": [
        "athena resouces arn"
    ]
},// Athena permission
{
    "Effect": "Allow",
    "Action": [
        "glue:GetTable",
        "glue:GetTables",
        "glue:GetDatabase",
        "glue:GetDatabases"
    ],
    "Resource": [
        "glue resouces arn"
    ]
},// This is an example for accessing aws glue
{
    "Effect": "Allow",
    "Action": [
        "s3:ListBucket",
        "s3:GetObject"
    ],
    "Resource": [
        "Amazon S3 data source bucket resources arn"
    ]
}, // S3 bucket for storing the tabular data.
{
    "Effect": "Allow",
    "Action": [
        "s3:GetBucketLocation",
        "s3:GetObject",
        "s3:ListBucket",
        "s3:ListBucketMultipartUploads",
        "s3:ListMultipartUploadParts",
        "s3:AbortMultipartUpload",
        "s3:CreateBucket",
        "s3:PutObject",
        "s3:PutBucketPublicAccessBlock"
    ],
```

```
    "Resource": [
        "S3 query result bucket resources arn"
    ]
} // Storing the query results
```

Weitere Informationen zur Athena IAM-Konfiguration finden Sie im Identitäts- und Zugriffsmanagement in Athena.

Einen Workspace erstellen

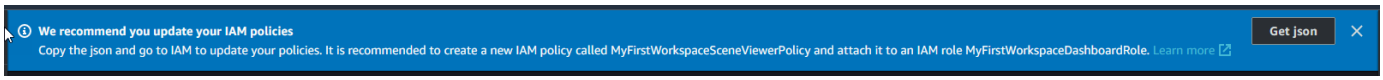
Gehen Sie wie folgt vor, um Ihren ersten Workspace zu erstellen und zu konfigurieren.

> **Note**
>
> In diesem Thema erfahren Sie, wie Sie einen einfachen Workspace mit einer einzigen Ressource erstellen. Für einen Workspace mit vollem Funktionsumfang und mehreren Ressourcen probieren Sie das Beispiel-Setup im Github-Repository für AWS IoT TwinMaker Beispiele aus.

1. Wählen Sie auf der Startseite der AWS IoT TwinMaker Konsole im linken Navigationsbereich Workspaces aus.
2. Wählen Sie auf der Seite Workspaces die Option Workspace erstellen aus.
3. Gib auf der Seite „Einen Workspace erstellen“ einen Namen für deinen Workspace ein.
4. (Optional) Füge eine Beschreibung für deinen Workspace hinzu.
5. Wählen Sie unter S3-Ressource die Option S3-Bucket erstellen aus. Diese Option erstellt einen Amazon S3 S3-Bucket, in dem Informationen und Ressourcen zum Workspace AWS IoT TwinMaker gespeichert werden. Jeder Workspace benötigt seinen eigenen Bucket.
6. Wählen Sie unter Ausführungsrolle entweder eine neue Rolle automatisch generieren oder die benutzerdefinierte IAM-Rolle aus, die Sie für diesen Workspace erstellt haben.

   Wenn Sie „Neue Rolle automatisch generieren“ wählen, wird AWS IoT TwinMaker der Rolle eine Richtlinie hinzugefügt, die der neuen Servicerolle Zugriff auf andere AWS Services gewährt, einschließlich der Lese- und Schreibberechtigung für den Amazon S3 S3-Bucket, den Sie im vorherigen Schritt angegeben haben. Informationen zur Zuweisung von Berechtigungen für diese Rolle finden Sie unter. ???

7. Wählen Sie „Arbeitsbereich erstellen“. Das folgende Banner wird oben auf der Workspaces-Seite angezeigt.



8. Wählen Sie Get json aus. Wir empfehlen Ihnen, die IAM-Richtlinie, die Sie sehen, zu der IAM-Rolle hinzuzufügen, die für Benutzer und Konten AWS IoT TwinMaker erstellt wurde, die das Grafana-Dashboard aufrufen. Der Name dieser Rolle folgt diesem Muster: *workspace-name*DashboardRole, Anweisungen zum Erstellen einer Richtlinie und zum Anhängen dieser an eine Rolle finden Sie unter Ändern einer Rollenberechtigungsrichtlinie (Konsole).

   Das folgende Beispiel enthält die Richtlinie, die der Dashboard-Rolle hinzugefügt werden soll.

   JSON

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

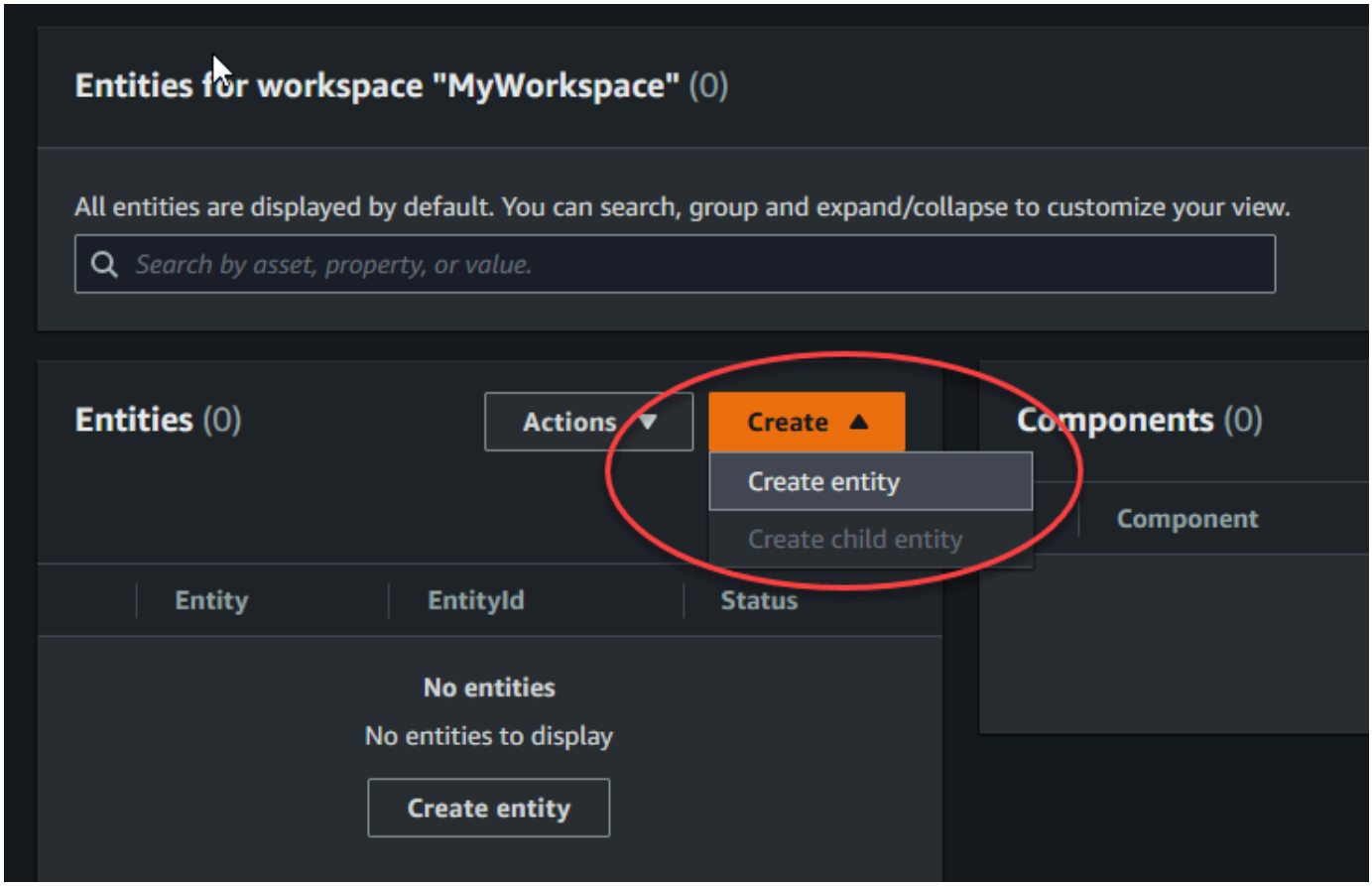
```
    },
    {
      "Effect": "Allow",
      "Action": "iottwinmaker:ListWorkspaces",
      "Resource": "*"
    }
  ]
}
```

Sie sind jetzt bereit, mit Ihrer ersten Entität ein Datenmodell für Ihren Workspace zu erstellen. Anweisungen dazu finden Sie unter Erstellen Sie Ihre erste Entität.

Erstellen Sie Ihre erste Entität

Gehen Sie wie folgt vor, um Ihre erste Entität zu erstellen.

1. Wählen Sie auf der Seite Workspaces Ihren Workspace und dann im linken Bereich Entitäten aus.
2. Wählen Sie auf der Seite Entitäten die Option Erstellen und dann Entität erstellen aus.



3. Geben Sie im Fenster „Entität erstellen“ einen Namen für Ihre Entität ein. In diesem Beispiel wird eine **`CookieMixer`** Entität verwendet.
4. (Optional) Geben Sie eine Beschreibung für Ihre Entität ein.
5. Wählen Sie Entität erstellen,

Entitäten enthalten Daten zu jedem Element in Ihrem Workspace. Sie fügen Daten in Entitäten ein, indem Sie Komponenten hinzufügen. AWS IoT TwinMaker stellt die folgenden integrierten Komponententypen bereit.

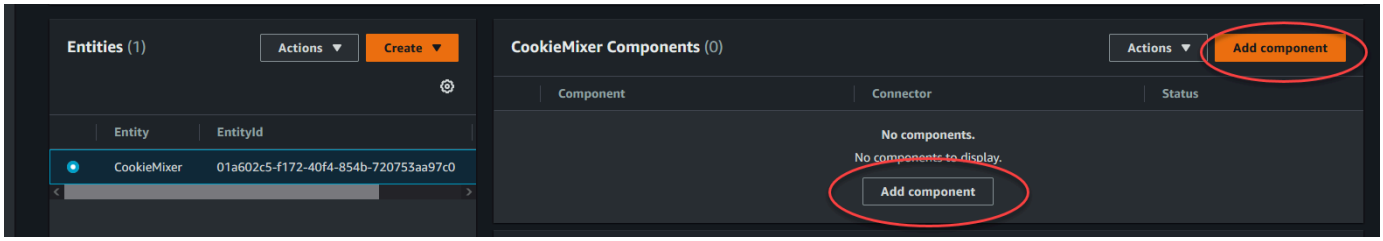
- Parameter: Fügt eine Reihe von Schlüssel-Wert-Eigenschaften hinzu.
- Dokument: Fügt einen Namen und eine URL für ein Dokument hinzu, das Informationen über die Entität enthält.
- Alarme: Stellt eine Verbindung zu einer Alarm-Zeitreihen-Datenquelle her.
- SiteWise Konnektor: Ruft Zeitreiheneigenschaften ab, die in einem Asset definiert sind. AWS IoT SiteWise

- Edge Connector für Kinesis Video Streams AWS IoT Greengrass: Ruft Videodaten vom Edge Connector für KVS ab. AWS IoT Greengrass Weitere Informationen finden Sie unter AWS IoT TwinMaker Videointegration.

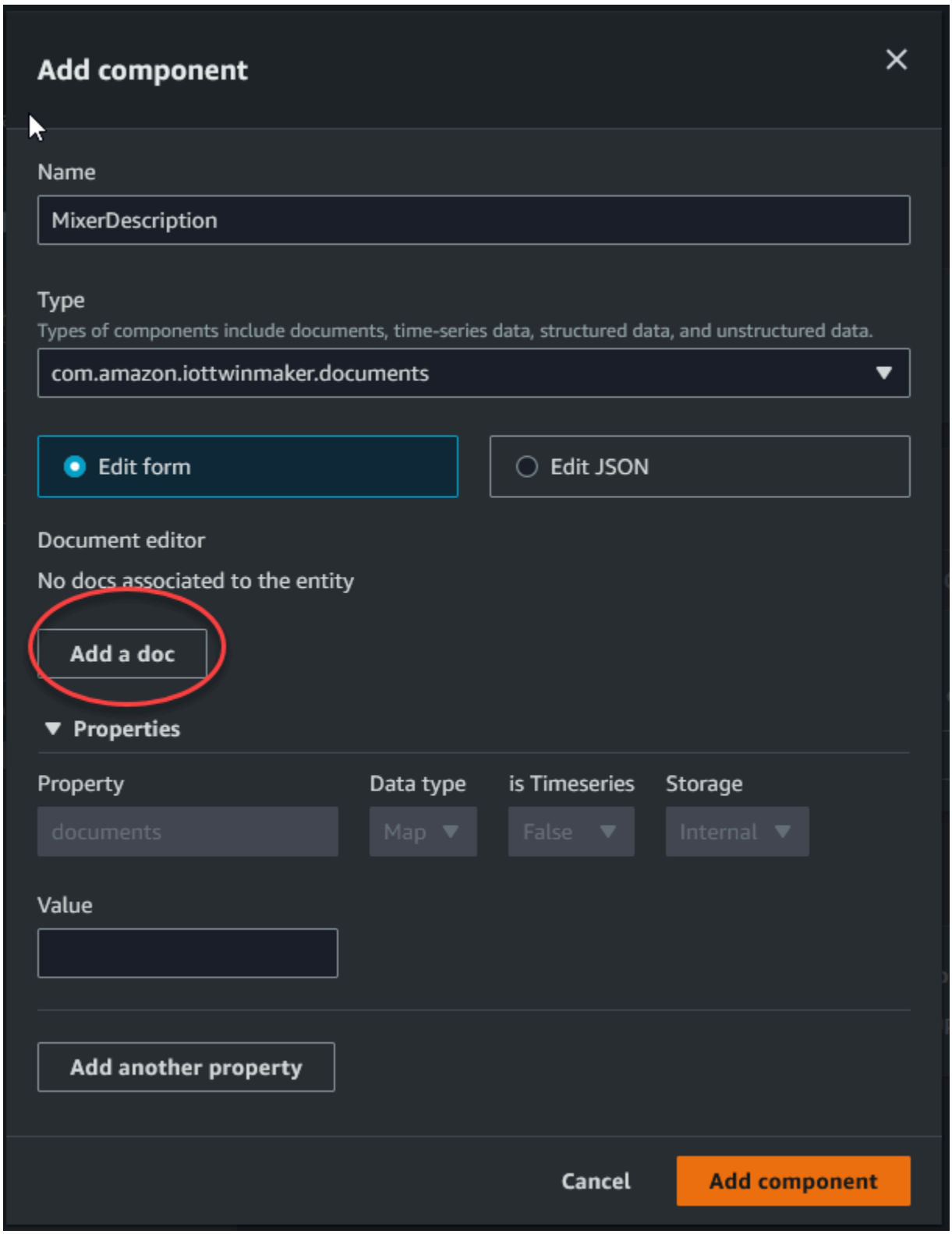
Sie können sich diese Komponententypen und ihre Definitionen ansehen, indem Sie im linken Bereich Komponententypen auswählen. Sie können auch einen neuen Komponententyp auf der Seite Komponententypen erstellen. Weitere Informationen zum Erstellen von Komponententypen finden Sie unterKomponententypen verwenden und erstellen.

In diesem Beispiel erstellen wir eine einfache Dokumentkomponente, die beschreibende Informationen zu Ihrer Entität hinzufügt.

1. Wählen Sie auf der Seite Entitäten die Entität aus und klicken Sie dann auf Komponente hinzufügen.



2. Geben Sie im Fenster Komponente hinzufügen einen Namen für Ihre Komponente ein. Da in diesem Beispiel eine Cookie-Mixer-Entität verwendet wird, geben wir diesen **`MixerDescription`** in das Feld Name ein.



3. Wählen Sie Dokument hinzufügen und geben Sie dann Werte für den Dokumentnamen und die externe URL ein. Mit der Komponente Dokumente können Sie eine Liste mit externen Dokumenten speichern URLs , die wichtige Informationen über die Entität enthalten.

4. Wählen Sie Komponente hinzufügen.

Sie sind jetzt bereit, Ihre erste Szene zu erstellen. Anweisungen dazu finden Sie unter AWS IoT TwinMaker Szenen erstellen und bearbeiten.

Ein AWS Konto einrichten

Wenn Sie noch kein Konto haben AWS-Konto, führen Sie die folgenden Schritte aus, um eines zu erstellen.

Um sich für eine anzumelden AWS-Konto

1. Öffnen Sie https://portal.aws.amazon.com/billing/die Anmeldung.
2. Folgen Sie den Online-Anweisungen.

   Während der Anmeldung erhalten Sie einen Telefonanruf oder eine Textnachricht und müssen einen Verifizierungscode über die Telefontasten eingeben.

   Wenn Sie sich für eine anmelden AWS-Konto, Root-Benutzer des AWS-Kontoswird eine erstellt. Der Root-Benutzer hat Zugriff auf alle AWS-Services und Ressourcen des Kontos. Als bewährte Sicherheitsmethode weisen Sie einem Administratorbenutzer Administratorzugriff zu und verwenden Sie nur den Root-Benutzer, um Aufgaben auszuführen, die Root-Benutzerzugriff erfordern.

Melde dich an für ein AWS-Konto

Wenn Sie noch keine haben AWS-Konto, führen Sie die folgenden Schritte aus, um eine zu erstellen.

Um sich für eine anzumelden AWS-Konto

1. Öffnen Sie https://portal.aws.amazon.com/billing/die Anmeldung.
2. Folgen Sie den Online-Anweisungen.

   Während der Anmeldung erhalten Sie einen Telefonanruf oder eine Textnachricht und müssen einen Verifizierungscode über die Telefontasten eingeben.

   Wenn Sie sich für eine anmelden AWS-Konto, Root-Benutzer des AWS-Kontoswird eine erstellt. Der Root-Benutzer hat Zugriff auf alle AWS-Services und Ressourcen des Kontos. Als bewährte

Sicherheitsmethode weisen Sie einem Administratorbenutzer Administratorzugriff zu und verwenden Sie nur den Root-Benutzer, um Aufgaben auszuführen, die Root-Benutzerzugriff erfordern.

AWS sendet Ihnen nach Abschluss des Anmeldevorgangs eine Bestätigungs-E-Mail. Du kannst jederzeit deine aktuellen Kontoaktivitäten einsehen und dein Konto verwalten, indem du zu https://aws.amazon.com/gehst und Mein Konto auswählst.

Erstellen eines Benutzers mit Administratorzugriff

Nachdem Sie sich für einen angemeldet haben AWS-Konto, sichern Sie Ihren Root-Benutzer des AWS-Kontos AWS IAM Identity Center, aktivieren und erstellen Sie einen Administratorbenutzer, sodass Sie den Root-Benutzer nicht für alltägliche Aufgaben verwenden.

Sichern Sie Ihre Root-Benutzer des AWS-Kontos

1. Melden Sie sich AWS Management Consoleals Kontoinhaber an, indem Sie Root-Benutzer auswählen und Ihre AWS-Konto E-Mail-Adresse eingeben. Geben Sie auf der nächsten Seite Ihr Passwort ein.

   Hilfe bei der Anmeldung mit dem Root-Benutzer finden Sie unter Anmelden als Root-Benutzer im AWS-Anmeldung Benutzerhandbuch zu.
2. Aktivieren Sie die Multi-Faktor-Authentifizierung (MFA) für den Root-Benutzer.

   Anweisungen finden Sie unter Aktivieren eines virtuellen MFA-Geräts für Ihren AWS-Konto Root-Benutzer (Konsole) im IAM-Benutzerhandbuch.

Erstellen eines Benutzers mit Administratorzugriff

1. Aktivieren Sie das IAM Identity Center.

   Anweisungen finden Sie unter Aktivieren AWS IAM Identity Center im AWS IAM Identity Center Benutzerhandbuch.
2. Gewähren Sie einem Administratorbenutzer im IAM Identity Center Benutzerzugriff.

   Ein Tutorial zur Verwendung von IAM-Identity-Center-Verzeichnis als Identitätsquelle finden Sie IAM-Identity-Center-Verzeichnis im Benutzerhandbuch unter Benutzerzugriff mit der Standardeinstellung konfigurieren.AWS IAM Identity Center

Anmelden als Administratorbenutzer

- Um sich mit Ihrem IAM-Identity-Center-Benutzer anzumelden, verwenden Sie die Anmelde-URL, die an Ihre E-Mail-Adresse gesendet wurde, als Sie den IAM-Identity-Center-Benutzer erstellt haben.

  Hilfe bei der Anmeldung mit einem IAM Identity Center-Benutzer finden Sie im AWS-Anmeldung Benutzerhandbuch unter Anmeldung beim AWS Access-Portal.

Weiteren Benutzern Zugriff zuweisen

1. Erstellen Sie im IAM-Identity-Center einen Berechtigungssatz, der den bewährten Vorgehensweisen für die Anwendung von geringsten Berechtigungen folgt.

   Anweisungen hierzu finden Sie unter Berechtigungssatz erstellen im AWS IAM Identity Center Benutzerhandbuch.
2. Weisen Sie Benutzer einer Gruppe zu und weisen Sie der Gruppe dann Single Sign-On-Zugriff zu.

   Eine genaue Anleitung finden Sie unter Gruppen hinzufügen im AWS IAM Identity Center Benutzerhandbuch.

Komponententypen verwenden und erstellen

Dieses Thema führt Sie durch die Werte und Strukturen, die Sie zum Erstellen eines AWS IoT TwinMaker Komponententyps verwenden. Es zeigt Ihnen, wie Sie ein Anforderungsobjekt erstellen, das Sie an die CreateComponentTypeAPI oder mithilfe des Komponententyp-Editors in der AWS IoT TwinMaker Konsole übergeben können.

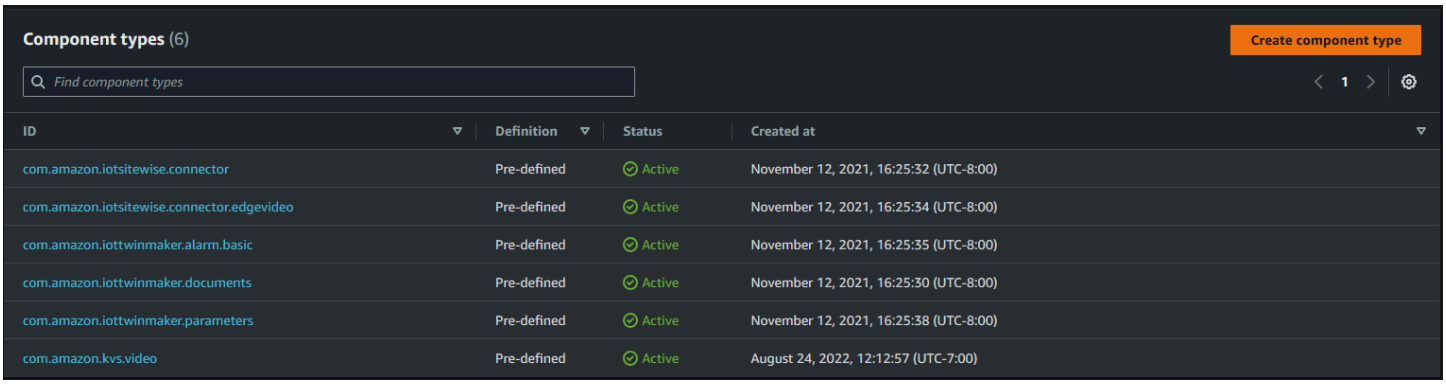
Komponenten bieten Kontext für Eigenschaften und Daten für die zugehörigen Entitäten.

Integrierte Komponententypen

Wenn Sie in der AWS IoT TwinMaker Konsole einen Workspace auswählen und dann im linken Bereich Komponententypen auswählen, werden Ihnen die folgenden Komponententypen angezeigt.

- com.amazon.iotsitewise.resourcesync: Ein Komponententyp, der Ihre AWS IoT SiteWise Ressourcen und Objektmodelle automatisch synchronisiert und sie in Entitätskomponenten und Komponententypen konvertiert. AWS IoT TwinMaker Weitere Informationen zur Verwendung der Asset-Synchronisierung finden Sie unter Asset-Synchronisierung mit. AWS IoT SiteWiseAWS IoT SiteWise
- com.amazon.iottwinmaker.alarm.basic: Eine grundlegende Alarmkomponente, die Alarmdaten aus einer externen Quelle an eine Entität überträgt. Diese Komponente enthält keine Funktion, die eine Verbindung zu einer bestimmten Datenquelle herstellt. Das bedeutet, dass die Alarmkomponente abstrakt ist und von einem anderen Komponententyp vererbt werden kann, der eine Datenquelle und eine Funktion angibt, die aus dieser Quelle liest.
- com.amazon.iottwinmaker.documents: Eine einfache Zuordnung von Titeln zu URLs Dokumenten, die Informationen über eine Entität enthalten.
- com.amazon.iotsitewise.connector.edgevideo: Eine Komponente, die mithilfe der Komponente Edge Connector for Kinesis Video Streams Video Videos von einem IoT-Gerät in eine Entität überträgt. AWS IoT Greengrass Die AWS IoT Greengrass Komponente Edge Connector for Kinesis Video Streams ist keine AWS IoT TwinMaker Komponente, sondern eine vorgefertigte AWS IoT Greengrass Komponente, die lokal auf Ihrem IoT-Gerät bereitgestellt wird.
- com.amazon.iotsitewise.connector: Eine Komponente, die Daten in eine Entität überträgt. AWS IoT SiteWise
- com.amazon.iottwinmaker.parameters: Eine Komponente, die einer Entität statische Schlüssel-Wert-Paare hinzufügt.

- com.amazon.kvs.video: Eine Komponente, die Videos aus Kinesis Video Streams in eine Entität überträgt. AWS IoT TwinMaker



Kernmerkmale der AWS IoT TwinMaker Komponententypen

In der folgenden Liste werden die Kernfunktionen der Komponententypen beschrieben.

- Eigenschaftsdefinitionen: Das PropertyDefinitionRequestObjekt definiert eine Eigenschaft, die Sie im Scene Composer auffüllen können, oder sie kann mit Daten aus externen Datenquellen gefüllt werden. Statische Eigenschaften, die Sie festlegen, werden in AWS IoT TwinMaker gespeichert. Zeitreiheneigenschaften und andere Eigenschaften, die aus Datenquellen abgerufen werden, werden extern gespeichert.

  Sie geben Eigenschaftsdefinitionen innerhalb einer Zeichenfolge für die `PropertyDefinitionRequest` Map an. Jede Zeichenfolge muss für die Map eindeutig sein.
- Funktionen: Das FunctionRequestObjekt spezifiziert eine Lambda-Funktion, die aus einer externen Datenquelle liest und möglicherweise in eine externe Datenquelle schreibt.

  Ein Komponententyp, der eine Eigenschaft mit einem Wert enthält, der extern gespeichert ist, für den jedoch keine entsprechende Funktion zum Abrufen der Werte vorhanden ist, ist ein abstrakter Komponententyp. Sie können konkrete Komponententypen von einem abstrakten Komponententyp aus erweitern. Sie können einer Entität keine abstrakten Komponententypen hinzufügen. Sie erscheinen nicht im Scene Composer.

  Sie geben Funktionen innerhalb einer Zeichenfolge an, die zugeordnet werden `FunctionRequest` sollen. Die Zeichenfolge muss einen der folgenden vordefinierten Funktionstypen angeben.
  - `dataReader`: Eine Funktion, die Daten aus einer externen Quelle bezieht.

- `dataReaderByEntity`: Eine Funktion, die Daten aus einer externen Quelle bezieht.

  Wenn Sie diese Art von Datenlesegerät verwenden, unterstützt der GetPropertyValueHistoryAPI-Vorgang nur entitätsspezifische Abfragen für Eigenschaften in diesem Komponententyp. (Sie können den Verlauf der Eigenschaftswerte nur für `componentName` + `entityId` anfordern.)

- `dataReaderByComponentType`: Eine Funktion, die Daten aus einer externen Quelle bezieht.

  Wenn Sie diese Art von Datenlesegerät verwenden, unterstützt der GetPropertyValueHistoryAPI-Vorgang nur entitätsübergreifende Abfragen für Eigenschaften in diesem Komponententyp. (Sie können nur die Historie der Eigenschaftswerte für anfordern`componentTypeId`.)

- `dataWriter`: Eine Funktion, die Daten in eine externe Quelle schreibt.
- `schemaInitializer`: Eine Funktion, die Eigenschaftswerte automatisch initialisiert, wenn Sie eine Entität erstellen, die den Komponententyp enthält.

Bei einem nicht abstrakten Komponententyp ist einer der drei Typen von Datenlesefunktionen erforderlich.

Ein Beispiel für eine Lambda-Funktion, die Time-Stream-Telemetriekomponenten, einschließlich Alarme, implementiert, finden Sie im Datenlesegerät unter Beispiele.AWS IoT TwinMaker

> **Note**
>
> Da der Alarmanschluss vom abstrakten Alarmkomponententyp erbt, muss die Lambda-Funktion den `alarm_key` Wert zurückgeben. Wenn Sie diesen Wert nicht zurückgeben, erkennt Grafana ihn nicht als Alarm. Dies ist für alle Komponenten erforderlich, die Alarme zurückgeben.

- Vererbung: Komponententypen fördern die Wiederverwendbarkeit von Code durch Vererbung. Ein Komponententyp kann bis zu 10 übergeordnete Komponententypen erben.

  Verwenden Sie den `extendsFrom` Parameter, um die Komponententypen anzugeben, von denen Ihr Komponententyp Eigenschaften und Funktionen erbt.

- isSingleton: Einige Komponenten enthalten Eigenschaften, wie z. B. Ortskoordinaten, die einer Entität nicht mehr als einmal hinzugefügt werden können. Setzen Sie den Wert des `isSingleton` Parameters auf, `true` um anzugeben, dass Ihr Komponententyp nur einmal in einer Entität enthalten sein kann.

Eigenschaftsdefinitionen erstellen

In der folgenden Tabelle werden die Parameter von a beschriebenPropertyDefinitionRequest.

Parameter	Beschreibung
isExternalId	Ein boolescher Wert, der angibt, ob es sich bei der Eigenschaft um einen eindeutigen Bezeichner (z. B. eine AWS IoT SiteWise Asset-ID) eines extern gespeicherten Eigenschaftswerts handelt. Standardmäßig wird der Wert dieser Eigenscha ft auf false gesetzt.
isStoredExternally	Ein boolescher Wert, der angibt, ob der Eigenschaftswert extern gespeichert wird. Standardmäßig wird der Wert dieser Eigenscha ft auf false gesetzt.
isTimeSeries	Ein boolescher Wert, der angibt, ob die Eigenschaft Zeitreihendaten speichert. Der Standardwert dieser Eigenschaft ist false
isRequiredInEntity	Ein boolescher Wert, der angibt, ob die Eigenschaft einen Wert in einer Entität haben muss, die den Komponententyp verwendet.
dataType	Ein DataTypeObjekt, das den Datentyp (z. B. Zeichenfolge, Map, Liste und Maßeinheit) der Eigenschaft angibt.
defaultValue	Ein DataValueObjekt, das den Standardwert der Eigenschaft angibt.

Parameter	Beschreibung
`configuration`	Eine string-to-string Map, die zusätzliche Informationen angibt, die Sie für die Verbindung mit einer externen Datenquelle benötigen.

Erstellung von Funktionen

In der folgenden Tabelle werden die Parameter von a beschrieben`FunctionRequest`.

Parameter	Beschreibung
`implementedBy`	Ein DataConnectorObjekt, das die Lambda-Funktion spezifiziert, die eine Verbindung zur externen Datenquelle herstellt.
`requiredProperties`	Eine Liste von Eigenschaften, die die Funktion benötigt, um aus einer externen Datenquelle zu lesen und in diese zu schreiben.
`scope`	Der Geltungsbereich der Funktion. Wird `Workspace` für Funktionen verwendet , deren Geltungsbereich sich über einen gesamten Workspace erstreckt. Wird `Entity` für Funktionen verwendet, deren Gültigkeitsbereich auf die Entität beschränkt ist, die die Komponente enthält.

Beispiele, die zeigen, wie Komponententypen erstellt und erweitert werden, finden Sie unter???.

Beispiele für Komponententypen

Dieses Thema enthält Beispiele, die zeigen, wie wichtige Konzepte von Komponententypen implementiert werden.

Alarm (Zusammenfassung)

Das folgende Beispiel ist der abstrakte Alarmkomponententyp, der in der AWS IoT TwinMaker Konsole angezeigt wird. Er enthält eine `functions` Liste, die aus einer bestehtdataReader, die keinen `implementedBy` Wert hat.

```
{
  "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
    }
  }
}
```

Hinweise:

Werte für `componentTypeId` und `workspaceID` sind erforderlich. Der Wert von `componentTypeId` muss für Ihren Workspace eindeutig sein. Der Wert von `alarm_key` ist eine eindeutige Kennung, die eine Funktion verwenden kann, um Alarmdaten von einer externen Quelle abzurufen. Der Wert des Schlüssels ist erforderlich und wird in gespeichert AWS IoT TwinMaker. Die `alarm_status` Zeitreihenwerte werden in der externen Quelle gespeichert.

Weitere Beispiele sind unter AWS IoT TwinMaker Beispiele verfügbar.

Timestream-Telemetrie

Das folgende Beispiel ist ein einfacher Komponententyp, der Telemetriedaten über einen bestimmten Komponententyp (z. B. einen Alarm oder einen Cookie-Mixer) aus einer externen Quelle abruft. Es spezifiziert eine Lambda-Funktion, die Komponententypen erben.

```
{
    "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 (erbt vom abstrakten Alarm)

Das folgende Beispiel erbt sowohl vom abstrakten Alarm- als auch vom Timestream-Telemetrie-Komponententyp. Es spezifiziert eine eigene Lambda-Funktion, die Alarmdaten abruft.

```
{
    "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**
>
> Da der Alarmanschluss vom abstrakten Alarmkomponententyp erbt, muss die Lambda-Funktion den `alarm_key` Wert zurückgeben. Wenn Sie diesen Wert nicht zurückgeben, erkennt Grafana ihn nicht als Alarm. Dies ist für alle Komponenten erforderlich, die Alarme zurückgeben.

Beispiele für Geräte

Die Beispiele in diesem Abschnitt zeigen, wie potenzielle Geräte modelliert werden. Sie können diese Beispiele verwenden, um Anregungen zu erhalten, wie Sie Geräte in Ihren eigenen Prozessen modellieren können.

Keksmischer

Das folgende Beispiel erbt vom Komponententyp Timestream-Telemetrie. Es spezifiziert zusätzliche Zeitreiheneigenschaften für die Rotationsgeschwindigkeit und Temperatur eines Cookie-Mixers.

```
{
    "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
        }
    }
}
```

Wassertank

Das folgende Beispiel erbt vom Komponententyp Timestream-Telemetrie. Es gibt zusätzliche Zeitreiheneigenschaften für das Volumen und die Durchflussrate eines Wassertanks an.

```
{
    "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
        }
    }
```

```
}
```

Standort des Weltraums

Das folgende Beispiel enthält Eigenschaften, deren Werte in gespeichert sind AWS IoT TwinMaker. Da die Werte von Benutzern angegeben und intern gespeichert werden, ist keine Funktion erforderlich, um sie abzurufen. In dem Beispiel wird der `RELATIONSHIP` Datentyp auch verwendet, um eine Beziehung zu einem anderen Komponententyp anzugeben.

Diese Komponente bietet einen einfachen Mechanismus, um einem digitalen Zwilling Kontext hinzuzufügen. Sie können damit Metadaten hinzufügen, die angeben, wo sich etwas befindet. Sie können diese Informationen auch in der Logik verwenden, um zu bestimmen, welche Kameras ein Gerät oder einen Raum sehen können, oder um zu wissen, wie eine Person an einen Ort geschickt werden kann.

```
{
    "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 Massenoperationen

Verwenden Sie a metadataTransferJob , um Ihre AWS IoT TwinMaker Ressourcen in großem Umfang zu übertragen und zu verwalten. A metadataTransferJob ermöglicht es Ihnen, Massenoperationen durchzuführen und Ressourcen zwischen AWS IoT TwinMaker und AWS IoT SiteWise und Amazon S3 zu übertragen.

Sie können Massenoperationen in den folgenden Szenarien verwenden:

- Massenmigration von Ressourcen und Daten zwischen Konten, z. B. Migration von einem Entwicklungskonto zu einem Produktionskonto.
- Umfangreiches Asset-Management, z. B. Hochladen und Bearbeiten von AWS IoT Ressourcen in großem Umfang.
- Massenimport Ihrer Assets in AWS IoT TwinMaker und AWS IoT SiteWise.
- Massenimport von AWS IoT TwinMaker Entitäten aus vorhandenen Ontologiedateien wie `revit BIM` Oder-Dateien.

Themen

- Wichtige Konzepte und Terminologie
- Durchführung von Massenimport- und Exportvorgängen
- AWS IoT TwinMaker Auftragsschema für die Übertragung von Metadaten

Wichtige Konzepte und Terminologie

AWS IoT TwinMaker Bei Massenoperationen werden die folgenden Konzepte und Begriffe verwendet:

- Import: Die Aktion, bei der Ressourcen in einen AWS IoT TwinMaker Arbeitsbereich verschoben werden. Zum Beispiel von einer lokalen Datei, einer Datei in einem Amazon S3 S3-Bucket oder von AWS IoT SiteWise zu einem AWS IoT TwinMaker Workspace.
- Export: Die Aktion, Ressourcen von einem AWS IoT TwinMaker Workspace auf einen lokalen Computer oder einen Amazon S3 S3-Bucket zu verschieben.
- Quelle: Der Startort, von dem aus Sie Ressourcen verschieben möchten.

Beispielsweise ist ein Amazon S3 S3-Bucket eine Importquelle und ein AWS IoT TwinMaker Workspace eine Exportquelle.

- Ziel: Der gewünschte Ort, an den Sie Ihre Ressourcen verschieben möchten.

  Beispielsweise ist ein Amazon S3 S3-Bucket ein Exportziel und ein AWS IoT TwinMaker Workspace ein Importziel.
- AWS IoT SiteWise Schema: Ein Schema, das zum Importieren und Exportieren von Ressourcen verwendet wird AWS IoT SiteWise.
- AWS IoT TwinMaker Schema: Ein Schema, das zum Importieren und Exportieren von Ressourcen verwendet wird AWS IoT TwinMaker.
- AWS IoT TwinMaker Ressourcen der obersten Ebene: Ressourcen, die in bestehenden APIs Ressourcen verwendet werden. Insbesondere eine Entität oder ein ComponentType.
- AWS IoT TwinMaker Ressourcen auf Unterebene: Verschachtelte Ressourcentypen, die in Metadatendefinitionen verwendet werden. Insbesondere eine Komponente.
- Metadaten: Wichtige Informationen, die für einen erfolgreichen Import oder Export erforderlich sind, AWS IoT SiteWise sowie AWS IoT TwinMaker Ressourcen.
- metadataTransferJob: Das Objekt, das beim Ausführen erstellt wurde`CreateMetadataTransferJob`.

AWS IoT TwinMaker metadataTransferJob Funktionalität

In diesem Thema wird das Verhalten AWS IoT TwinMaker bei der Ausführung eines Massenvorgangs erläutert, d. h. wie ein verarbeitet metadataTransferJob wird. Außerdem wird erklärt, wie Sie ein Schema mit den Metadaten definieren, die für die Übertragung Ihrer Ressourcen erforderlich sind. AWS IoT TwinMaker Massenoperationen unterstützen die folgenden Funktionen:

- Ressource auf oberster Ebene erstellen oder ersetzen: AWS IoT TwinMaker erstellt neue Ressourcen oder ersetzt alle vorhandenen Ressourcen, die eindeutig durch eine Ressourcen-ID identifiziert werden.

  Wenn beispielsweise eine Entität im System vorhanden ist, wird die Entitätsdefinition durch die neue Entitätsdefinition ersetzt, die in der Vorlage unter dem `Entity` Schlüssel definiert ist.
- Unterressource erstellen oder ersetzen:

Von der EntityComponent Ebene aus können Sie nur eine Komponente erstellen oder ersetzen. Die Entität muss bereits existieren, andernfalls erzeugt die Aktion eine ValidationException.

Auf der Eigenschafts- oder Beziehungsebene können Sie nur eine Eigenschaft oder Beziehung erstellen oder ersetzen, und die Eigenschaft, die sie enthält, EntityComponent muss bereits vorhanden sein.

- Unterressource löschen:

  AWS IoT TwinMaker unterstützt auch das Löschen von Unterressourcen. Eine Unterressource kann eine Komponente, Eigenschaft oder Beziehung sein.

  Wenn Sie eine Komponente löschen möchten, müssen Sie dies auf Entitätsebene tun.

  Wenn Sie eine Eigenschaft oder Beziehung löschen möchten, müssen Sie dies von der Entität oder EntityComponent Ebene aus tun.

  Um eine Unterressource zu löschen, aktualisieren Sie die Ressource der höheren Ebene und lassen die Definition der Unterressource weg.
- Keine Löschung von Ressourcen der obersten Ebene: Es AWS IoT TwinMaker werden niemals Ressourcen der obersten Ebene gelöscht. Eine Ressource der obersten Ebene bezieht sich auf eine Entität oder. ComponentType
- Keine Unterressourcendefinitionen für dieselbe Ressource der obersten Ebene in einer Vorlage:

  Sie können nicht die vollständige Entitätsdefinition und die Subressourcendefinition (wie Eigenschaft) derselben Entität in derselben Vorlage bereitstellen.

  Wenn eine EntityID in Entity verwendet wird, können Sie dieselbe ID nicht in Entity EntityComponent, Property oder Relationship verwenden.

  Wenn eine EntityID- oder ComponentName-Kombination in verwendet wird EntityComponent, können Sie nicht dieselbe Kombination in EntityComponent, Eigenschaft oder Beziehung verwenden.

  Wenn eine Kombination aus EntityID, ComponentName und PropertyName in einer Eigenschaft oder Beziehung verwendet wird, können Sie dieselbe Kombination nicht in der Eigenschaft oder Beziehung verwenden.
- ExternalId ist optional für AWS IoT TwinMaker: Das ExternalId kann verwendet werden, um Ihnen bei der Identifizierung Ihrer Ressourcen zu helfen.

Durchführung von Massenimport- und Exportvorgängen

In diesem Thema wird beschrieben, wie Sie Massenimport- und -exportvorgänge ausführen und wie Sie mit Fehlern bei Ihren Übertragungsaufträgen umgehen. Es enthält Beispiele für Übertragungsjobs mithilfe von CLI-Befehlen.

Die AWS IoT TwinMaker API-Referenz enthält Informationen zu den CreateMetadataTransferJobund anderen API-Aktionen.

Themen

- metadataTransferJob Voraussetzungen
- IAM-Berechtigungen
- Führen Sie einen Massenvorgang aus
- Fehlerbehandlung
- Metadaten-Vorlagen importieren
- AWS IoT TwinMaker metadataTransferJob Beispiele

metadataTransferJob Voraussetzungen

Bitte erfüllen Sie die folgenden Voraussetzungen, bevor Sie eine ausführen metadataTransferJob:

- Erstellen Sie einen AWS IoT TwinMaker Arbeitsbereich. Der Workspace kann das Importziel oder die Exportquelle für einen sein metadataTransferJob. Informationen zum Erstellen eines Workspace finden Sie unterEinen Workspace erstellen.
- Erstellen Sie einen Amazon S3 S3-Bucket zum Speichern von Ressourcen. Weitere Informationen zur Verwendung von Amazon S3 finden Sie unter Was ist Amazon S3?

IAM-Berechtigungen

Wenn Sie Massenoperationen durchführen, müssen Sie eine IAM-Richtlinie mit Berechtigungen erstellen, um den Austausch von AWS Ressourcen zwischen Amazon S3, AWS IoT TwinMaker AWS IoT SiteWise, und Ihrem lokalen Computer zu ermöglichen. Weitere Informationen zur Erstellung von IAM-Richtlinien finden Sie unter IAM-Richtlinien erstellen.

Die AWS IoT TwinMaker Richtlinienerklärungen für AWS IoT SiteWise und Amazon S3 sind hier aufgeführt:

- AWS IoT TwinMaker Richtlinie:

  JSON

```
{
    "Version": "2012-10-17",
    "Statement": [{
        "Effect": "Allow",
        "Action": [
            "s3:PutObject",
            "s3:GetObject",
            "s3:GetBucketLocation",
            "s3:ListBucket",
            "s3:AbortMultipartUpload",
            "s3:ListBucketMultipartUploads",
            "s3:ListMultipartUploadParts"
        ],
        "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "iottwinmaker:GetWorkspace",
                "iottwinmaker:CreateEntity",
                "iottwinmaker:GetEntity",
                "iottwinmaker:UpdateEntity",
                "iottwinmaker:GetComponentType",
                "iottwinmaker:CreateComponentType",
                "iottwinmaker:UpdateComponentType",
                "iottwinmaker:ListEntities",
                "iottwinmaker:ListComponentTypes",
                "iottwinmaker:ListTagsForResource",
                "iottwinmaker:TagResource",
                "iottwinmaker:UntagResource"
            ],
            "Resource": "*"
        }
    ]
}
```

- AWS IoT SiteWise Politik:

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [{
        "Effect": "Allow",
        "Action": [
            "s3:PutObject",
            "s3:GetObject",
            "s3:GetBucketLocation",
            "s3:ListBucket",
            "s3:AbortMultipartUpload",
            "s3:ListBucketMultipartUploads",
            "s3:ListMultipartUploadParts"
        ],
        "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "iotsitewise:CreateAsset",
                "iotsitewise:CreateAssetModel",
                "iotsitewise:UpdateAsset",
                "iotsitewise:UpdateAssetModel",
                "iotsitewise:UpdateAssetProperty",
                "iotsitewise:ListAssets",
                "iotsitewise:ListAssetModels",
                "iotsitewise:ListAssetProperties",
                "iotsitewise:ListAssetModelProperties",
                "iotsitewise:ListAssociatedAssets",
                "iotsitewise:DescribeAsset",
                "iotsitewise:DescribeAssetModel",
                "iotsitewise:DescribeAssetProperty",
                "iotsitewise:AssociateAssets",
                "iotsitewise:DisassociateAssets",
                "iotsitewise:AssociateTimeSeriesToAssetProperty",
                "iotsitewise:DisassociateTimeSeriesFromAssetProperty",
                "iotsitewise:BatchPutAssetPropertyValue",
                "iotsitewise:BatchGetAssetPropertyValue",
                "iotsitewise:TagResource",
                "iotsitewise:UntagResource",
                "iotsitewise:ListTagsForResource"
            ],
```

```
            "Resource": "*"
        }
    ]
}
```

- Amazon S3 S3-Richtlinie:

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

Alternativ können Sie Ihre Amazon S3 S3-Richtlinie so festlegen, dass sie nur auf einen einzigen Amazon S3 S3-Bucket zugreift. Weitere Informationen finden Sie in der folgenden Richtlinie.

Amazon S3 S3-Richtlinie für einen einzelnen Bucket mit Geltungsbereich

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

Richten Sie die Zugriffskontrolle für ein metadataTransferJob

Um zu steuern, auf welche Art von Jobs ein Benutzer zugreifen kann, fügen Sie der Rolle, die für den Anruf AWS IoT TwinMaker verwendet wird, die folgende IAM-Richtlinie hinzu.

> **Note**
>
> Diese Richtlinie ermöglicht nur den Zugriff auf AWS IoT TwinMaker Import- und Exportaufträge, die Ressourcen zu und von Amazon S3 übertragen.

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

Führen Sie einen Massenvorgang aus

In diesem Abschnitt wird beschrieben, wie Sie Massenimport- und -exportvorgänge durchführen.

Daten von Amazon S3 importieren nach AWS IoT TwinMaker

1. Geben Sie die Ressourcen, die Sie übertragen möchten, mithilfe des AWS IoT TwinMaker metadataTransferJob Schemas an. Erstellen und speichern Sie Ihre Schemadatei in Ihrem Amazon S3 S3-Bucket.

Beispiele für Schemas finden Sie unterMetadaten-Vorlagen importieren.

2. Erstellen Sie einen Anforderungstext und speichern Sie ihn als JSON-Datei. Der Anforderungstext gibt die Quelle und das Ziel für den Übertragungsjob an. Stellen Sie sicher, dass Sie Ihren Amazon S3 S3-Bucket als Quelle und Ihren AWS IoT TwinMaker Workspace als Ziel angeben.

   Im Folgenden finden Sie ein Beispiel für einen Anfragetext:

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

   Notieren Sie sich den Dateinamen, den Sie Ihrem Anfragetext gegeben haben. Sie benötigen ihn im nächsten Schritt. In diesem Beispiel wird der Hauptteil der Anfrage benanntcreateMetadataTransferJobImport.json.

3. Führen Sie zum Aufrufen den folgenden CLI-Befehl aus CreateMetadataTransferJob (ersetzen Sie den Namen der input-json-Datei durch den Namen, den Sie Ihrem Anfragetext gegeben haben):

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

   Dadurch wird eine erstellt metadataTransferJob und der Prozess der Übertragung Ihrer ausgewählten Ressourcen gestartet.

Daten von AWS IoT TwinMaker zu Amazon S3 exportieren

1. Erstellen Sie einen JSON-Anforderungstext mit den entsprechenden Filtern, um die Ressourcen auszuwählen, die Sie exportieren möchten. Für dieses Beispiel verwenden wir:

```
{
    "metadataTransferJobId": "your-transfer-job-Id",
    "sources": [{
        "type": "iottwinmaker",
        "iotTwinMakerConfiguration": {
            "workspace": "arn:aws:iottwinmaker:us-
east-1:111122223333:workspace/your-workspace-name",
            "filters": [{
                "filterByEntity": {
                    "entityId": "parent"
                }},
                {
                "filterByEntity": {
                    "entityId": "child"
                }},
                {
                "filterByComponentType": {
                    "componentTypeId": "component.type.minimal"
                }}
            ]
        }
    }],
    "destination": {
        "type": "s3",
        "s3Configuration": {
            "location": "arn:aws:s3:::amzn-s3-demo-bucket"
        }
    }
}
```

   Mit dem `filters` Array können Sie angeben, welche Ressourcen exportiert werden. In diesem Beispiel filtern wir nach`entity`, und`componentType`.

   Stellen Sie sicher, dass Sie Ihren AWS IoT TwinMaker Workspace als Quelle und Ihren Amazon S3 S3-Bucket als Ziel des Metadatentransferjobs angeben.

Speichern Sie Ihren Anfragetext und notieren Sie sich den Dateinamen. Sie benötigen ihn im nächsten Schritt. In diesem Beispiel haben wir unseren Anfragetext benannt`createMetadataTransferJobExport.json`.

2. Führen Sie zum Aufrufen den folgenden CLI-Befehl aus `CreateMetadataTransferJob` (ersetzen Sie den Namen der input-json-Datei durch den Namen, den Sie Ihrem Anfragetext gegeben haben):

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

Dadurch wird eine erstellt metadataTransferJob und der Prozess der Übertragung Ihrer ausgewählten Ressourcen gestartet.

Verwenden Sie die folgenden Befehle, um den Status eines Übertragungsauftrags zu überprüfen oder zu aktualisieren:

- Verwenden Sie die CancelMetadataTransferJobAPI-Aktion, um einen Job abzubrechen. Wenn Sie aufrufen CancelMetadataTransferJob, bricht die API nur eine Ausführung ab metadataTransferJob, und alle Ressourcen, die bereits exportiert oder importiert wurden, sind von diesem API-Aufruf nicht betroffen.
- Verwenden Sie die GetMetadataTransferJobAPI-Aktion, um Informationen zu einem bestimmten Job abzurufen.

  Oder Sie können einen vorhandenen Übertragungsauftrag mit dem folgenden CLI-Befehl aufrufen GetMetadataTransferJob :

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

  Wenn Sie einen nicht vorhandenen AWS IoT TwinMaker Import- oder Exportauftrag aufrufen GetMetadataTransferJob , erhalten Sie als Antwort eine `ResourceNotFoundException` Fehlermeldung.
- Verwenden Sie die ListMetadataTransferJobsAPI-Aktion, um aktuelle Jobs aufzulisten.

  Hier ist ein CLI-Beispiel, das ListMetadataTransferJobs mit AWS IoT TwinMaker als DestinationType und `s3` als sourceType aufruft:

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

> **Note**
>
> Sie können die Werte für die Parameter sourceType und DestinationType so ändern, dass sie der Quelle und dem Ziel Ihres Import- oder Exportjobs entsprechen.

Weitere Beispiele für CLI-Befehle, die diese API-Aktionen aufrufen, finden Sie unterAWS IoT TwinMaker metadataTransferJob Beispiele.

Falls Sie während des Übertragungsvorgangs auf Fehler stoßen, finden Sie weitere Informationen unterFehlerbehandlung.

Fehlerbehandlung

Nachdem Sie einen Übertragungsauftrag erstellt und ausgeführt haben, können Sie ihn anrufen, GetMetadataTransferJob um alle aufgetretenen Fehler zu diagnostizieren:

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

Sobald der Status des Auftrags angezeigt wird`COMPLETED`, können Sie die Ergebnisse des Auftrags überprüfen. GetMetadataTransferJob gibt ein aufgerufenes Objekt zurück `MetadataTransferJobProgress`, das die folgenden Felder enthält:

- FailedCount: Gibt die Anzahl der Ressourcen an, die während des Übertragungsvorgangs ausgefallen sind.
- skippedCount: Gibt die Anzahl der Ressourcen an, die während des Übertragungsvorgangs übersprungen wurden.
- succeededCount: Gibt die Anzahl der Ressourcen an, die während des Übertragungsvorgangs erfolgreich waren.
- TotalCount: Gibt die Gesamtzahl der Ressourcen an, die am Übertragungsprozess beteiligt sind.

Zusätzlich wird ein reportURL-Element zurückgegeben, das eine vorsignierte URL enthält. Wenn Ihr Übertragungsauftrag Fehler enthält, die Sie weiter untersuchen möchten, können Sie über diese URL einen vollständigen Fehlerbericht herunterladen.

Metadaten-Vorlagen importieren

Sie können viele Komponenten, Komponententypen oder Entitäten mit einem einzigen Massenimportvorgang importieren. Die Beispiele in diesem Abschnitt zeigen, wie das geht.

template: Importing entities

Verwenden Sie das folgende Vorlagenformat für einen Job, der Entitäten importiert:

```
{
  "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

Verwenden Sie das folgende Vorlagenformat für einen Job, der ComponentTypes importiert:

```
{
  "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

Verwenden Sie das folgende Vorlagenformat für einen Job, der Komponenten importiert:

```
{
  "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 Beispiele

Verwenden Sie die folgenden Befehle, um Ihre Metadatentransfers zu verwalten:

- CreateMetadataTransferJobAPI-Aktion.

  Beispiel für einen CLI-Befehl:

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

- Verwenden Sie die CancelMetadataTransferJobAPI-Aktion, um einen Job abzubrechen.

  Beispiel für einen CLI-Befehl:

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

```
--metadata-transfer-job-id job-to-cancel-id
```

Wenn Sie aufrufen CancelMetadataTransferJob, wird nur ein bestimmter Metadatentransferauftrag abgebrochen, und alle bereits exportierten oder importierten Ressourcen sind davon nicht betroffen.

- Verwenden Sie die GetMetadataTransferJobAPI-Aktion, um Informationen zu einem bestimmten Job abzurufen.

  Beispiel für einen CLI-Befehl:

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

- Verwenden Sie die ListMetadataTransferJobsAPI-Aktion, um aktuelle Jobs aufzulisten.

  Sie können die zurückgegebenen Ergebnisse ListMetadataTransferJobs mithilfe einer JSON-Datei filtern. Sehen Sie sich das folgende Verfahren mit der CLI an:

  1. Erstellen Sie eine CLI-Eingabe-JSON-Datei, um die Filter anzugeben, die Sie verwenden möchten:

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

     Speichern Sie es und notieren Sie sich den Dateinamen. Sie benötigen ihn, wenn Sie den CLI-Befehl eingeben.

  2. Verwenden Sie die JSON-Datei als Argument für den folgenden CLI-Befehl:

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

AWS IoT TwinMaker Auftragsschema für die Übertragung von Metadaten

metadataTransferJob Schema importieren: Verwenden Sie dieses AWS IoT TwinMaker Metadatenschema, um Ihre Daten zu validieren, wenn Sie sie in einen Amazon S3 S3-Bucket hochladen:

```
{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "title": "IoTTwinMaker",
  "description": "Metadata transfer job resource schema for IoTTwinMaker",
  "definitions": {
    "ExternalId": {
      "type": "string",
      "minLength": 1,
      "maxLength": 128,
      "pattern": "[a-zA-Z0-9][a-zA-Z_\\-0-9.:]*[a-zA-Z0-9]+"
    },
    "Description": {
      "type": "string",
      "minLength": 0,
      "maxLength": 512
    },
    "DescriptionWithDefault": {
      "type": "string",
      "minLength": 0,
      "maxLength": 512,
      "default": ""
    },
    "ComponentTypeName": {
      "description": "A friendly name for the component type.",
      "type": "string",
      "pattern": ".*[^\\u0000-\\u001F\\u007F]*.*",
      "minLength": 1,
      "maxLength": 256
    },
    "ComponentTypeId": {
      "description": "The ID of the component type.",
      "type": "string",
      "pattern": "[a-zA-Z_.\\-0-9:]+",
      "minLength": 1,
      "maxLength": 256
```

```
    },
    "ComponentName": {
      "description": "The name of the component.",
      "type": "string",
      "pattern": "[a-zA-Z_\\-0-9]+",
      "minLength": 1,
      "maxLength": 256
    },
    "EntityId": {
      "description": "The ID of the entity.",
      "type": "string",
      "minLength": 1,
      "maxLength": 128,
      "pattern": "[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}|^[a-zA-
Z0-9][a-zA-Z_\\-0-9.:]*[a-zA-Z0-9]+"
    },
    "EntityName": {
      "description": "The name of the entity.",
      "type": "string",
      "minLength": 1,
      "maxLength": 256,
      "pattern": "[a-zA-Z_0-9-.][a-zA-Z_0-9-. ]*[a-zA-Z0-9]+"
    },
    "ParentEntityId": {
      "description": "The ID of the parent entity.",
      "type": "string",
      "minLength": 1,
      "maxLength": 128,
      "pattern": "\\$ROOT|^[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]
{12}|^[a-zA-Z0-9][a-zA-Z_\\-0-9.:]*[a-zA-Z0-9]+",
      "default": "$ROOT"
    },
    "DisplayName": {
      "description": "A friendly name for the property.",
      "type": "string",
      "pattern": ".*[^\\u0000-\\u001F\\u007F]*.*",
      "minLength": 0,
      "maxLength": 256
    },
    "Tags": {
      "description": "Metadata that you can use to manage the entity / componentType",
      "patternProperties": {
        "^([\\p{L}\\p{Z}\\p{N}_.:/=+\\-@]*)$": {
          "type": "string",
```

```
          "minLength": 1,
          "maxLength": 256
        }
      },
      "existingJavaType": "java.util.Map<String,String>",
      "minProperties": 0,
      "maxProperties": 50
    },
    "Relationship": {
      "description": "The type of the relationship.",
      "type": "object",
      "properties": {
        "relationshipType": {
          "description": "The type of the relationship.",
          "type": "string",
          "pattern": ".*",
          "minLength": 1,
          "maxLength": 256
        },
        "targetComponentTypeId": {
          "description": "The ID of the target component type associated with this
relationship.",
          "$ref": "#/definitions/ComponentTypeId"
        }
      },
      "additionalProperties": false
    },
    "DataValue": {
      "description": "An object that specifies a value for a property.",
      "type": "object",
      "properties": {
        "booleanValue": {
          "description": "A Boolean value.",
          "type": "boolean"
        },
        "doubleValue": {
          "description": "A double value.",
          "type": "number"
        },
        "expression": {
          "description": "An expression that produces the value.",
          "type": "string",
          "pattern": "(^\\$\\{Parameters\\.[a-zA-z]+([a-zA-z_0-9]*)}$)",
          "minLength": 1,
```

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

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

```
          "$ref": "#/definitions/Relationship"
        },
        "type": {
          "description": "The underlying type of the data type.",
          "type": "string",
          "enum": [
            "RELATIONSHIP",
            "STRING",
            "LONG",
            "BOOLEAN",
            "INTEGER",
            "DOUBLE",
            "LIST",
            "MAP"
          ]
        },
        "unitOfMeasure": {
          "description": "The unit of measure used in this data type.",
          "type": "string",
          "pattern": ".*",
          "minLength": 1,
          "maxLength": 256
        }
      },
      "required": [
        "type"
      ],
      "additionalProperties": false
    },
    "PropertyDefinition": {
      "description": "An object that specifies information about a property.",
      "type": "object",
      "properties": {
        "configuration": {
          "description": "An object that specifies information about a property.",
          "patternProperties": {
            "[a-zA-Z_\\-0-9]+": {
              "type": "string",
              "pattern": "[a-zA-Z_\\-0-9]+",
              "minLength": 1,
              "maxLength": 256
            }
          },
          "existingJavaType": "java.util.Map<String,String>"
```

```
        },
        "dataType": {
          "description": "An object that contains information about the data type.",
          "$ref": "#/definitions/DataType"
        },
        "defaultValue": {
          "description": "An object that contains the default value.",
          "$ref": "#/definitions/DataValue"
        },
        "displayName": {
          "description": "An object that contains the default value.",
          "$ref": "#/definitions/DisplayName"
        },
        "isExternalId": {
          "description": "A Boolean value that specifies whether the property ID comes
from an external data store.",
          "type": "boolean",
          "default": null
        },
        "isRequiredInEntity": {
          "description": "A Boolean value that specifies whether the property is
required.",
          "type": "boolean",
          "default": null
        },
        "isStoredExternally": {
          "description": "A Boolean value that specifies whether the property is stored
externally.",
          "type": "boolean",
          "default": null
        },
        "isTimeSeries": {
          "description": "A Boolean value that specifies whether the property consists
of time series data.",
          "type": "boolean",
          "default": null
        }
      },
      "additionalProperties": false
    },
    "PropertyDefinitions": {
      "type": "object",
      "patternProperties": {
        "[a-zA-Z_\\-0-9]+": {
```

```
          "$ref": "#/definitions/PropertyDefinition"
        }
      },
      "additionalProperties": {
        "$ref": "#/definitions/PropertyDefinition"
      }
    },
    "Property": {
      "type": "object",
      "properties": {
        "definition": {
          "description": "The definition of the property",
          "$ref": "#/definitions/PropertyDefinition"
        },
        "value": {
          "description": "The value of the property.",
          "$ref": "#/definitions/DataValue"
        }
      },
      "additionalProperties": false
    },
    "Properties": {
      "type": "object",
      "patternProperties": {
        "[a-zA-Z_\\-0-9]+": {
          "$ref": "#/definitions/Property"
        }
      },
      "additionalProperties": {
        "$ref": "#/definitions/Property"
      }
    },
    "PropertyName": {
      "type": "string",
      "pattern": "[a-zA-Z_\\-0-9]+"
    },
    "PropertyGroup": {
      "description": "An object that specifies information about a property group.",
      "type": "object",
      "properties": {
        "groupType": {
          "description": "The type of property group.",
          "type": "string",
          "enum": [
```

```
            "TABULAR"
          ]
        },
        "propertyNames": {
          "description": "The list of property names in the property group.",
          "type": "array",
          "minItems": 1,
          "maxItems": 256,
          "uniqueItems": true,
          "insertionOrder": false,
          "items": {
            "$ref": "#/definitions/PropertyName"
          },
          "default": null
        }
      },
      "additionalProperties": false
    },
    "PropertyGroups": {
      "type": "object",
      "patternProperties": {
        "[a-zA-Z_\\-0-9]+": {
          "$ref": "#/definitions/PropertyGroup"
        }
      },
      "additionalProperties": {
        "$ref": "#/definitions/PropertyGroup"
      }
    },
    "Component": {
      "type": "object",
      "properties": {
        "componentTypeId": {
          "$ref": "#/definitions/ComponentTypeId"
        },
        "description": {
          "$ref": "#/definitions/Description"
        },
        "properties": {
          "description": "An object that maps strings to the properties to set in the
component type. Each string in the mapping must be unique to this object.",
          "$ref": "#/definitions/Properties"
        },
        "propertyGroups": {
```

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

```
      "additionalProperties": false
    },
    "Function": {
      "description": "The function of component type.",
      "type": "object",
      "properties": {
        "implementedBy": {
          "description": "The data connector.",
          "$ref": "#/definitions/DataConnector"
        },
        "requiredProperties": {
          "description": "The required properties of the function.",
          "type": "array",
          "minItems": 1,
          "maxItems": 256,
          "uniqueItems": true,
          "insertionOrder": false,
          "items": {
            "$ref": "#/definitions/RequiredProperty"
          },
          "default": null
        },
        "scope": {
          "description": "The scope of the function.",
          "type": "string",
          "enum": [
            "ENTITY",
            "WORKSPACE"
          ]
        }
      },
      "additionalProperties": false
    },
    "Entity": {
      "type": "object",
      "properties": {
        "description": {
          "description": "The description of the entity.",
          "$ref": "#/definitions/DescriptionWithDefault"
        },
        "entityId": {
          "$ref": "#/definitions/EntityId"
        },
        "entityExternalId": {
```

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

```
        },
        "componentTypeName": {
          "$ref": "#/definitions/ComponentTypeName"
        },
        "extendsFrom": {
          "description": "Specifies the parent component type to extend.",
          "type": "array",
          "minItems": 1,
          "maxItems": 256,
          "uniqueItems": true,
          "insertionOrder": false,
          "items": {
            "$ref": "#/definitions/ComponentTypeId"
          },
          "default": null
        },
        "functions": {
          "description": "a Map of functions in the component type. Each function's key
must be unique to this map.",
          "type": "object",
          "patternProperties": {
            "[a-zA-Z_\\-0-9]+": {
              "$ref": "#/definitions/Function"
            }
          },
          "additionalProperties": {
            "$ref": "#/definitions/Function"
          }
        },
        "isSingleton": {
          "description": "A Boolean value that specifies whether an entity can have
more than one component of this type.",
          "type": "boolean",
          "default": false
        },
        "propertyDefinitions": {
          "description": "An map of the property definitions in the component type.
Each property definition's key must be unique to this map.",
          "$ref": "#/definitions/PropertyDefinitions"
        },
        "propertyGroups": {
          "description": "An object that maps strings to the property groups to set in
the component type. Each string in the mapping must be unique to this object.",
          "$ref": "#/definitions/PropertyGroups"
```

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

```
        "componentName"
      ],
      "additionalProperties": false
    }
  },
  "additionalProperties": false,
  "properties": {
    "entities": {
      "type": "array",
      "uniqueItems": false,
      "items": {
        "$ref": "#/definitions/Entity"
      }
    },
    "componentTypes": {
      "type": "array",
      "uniqueItems": false,
      "items": {
        "$ref": "#/definitions/ComponentType"
      }
    },
    "entityComponents": {
      "type": "array",
      "uniqueItems": false,
      "items": {
        "$ref": "#/definitions/EntityComponent"
      },
      "default": null
    }
  }
}
```

Hier ist ein Beispiel, das einen neuen ComponentType namens `component.type.intial` und eine Entität mit dem Namen `initial` erstellt:

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

```
  ],
  "entities": [
    {
      "entityName": "initial",
      "entityId": "initial"
    }
  ]
}
```

Hier ist ein Beispiel, das bestehende Entitäten aktualisiert:

```
{
  "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 Datenanschlüsse

AWS IoT TwinMaker verwendet eine connectorbasierte Architektur, sodass Sie Daten aus Ihrem eigenen Datenspeicher mit verbinden können. AWS IoT TwinMaker Das bedeutet, dass Sie vor der Verwendung keine Daten migrieren müssen. AWS IoT TwinMaker AWS IoT TwinMaker Unterstützt derzeit Erstanbieter-Konnektoren für AWS IoT SiteWise. Wenn Sie Modellierungs- und Eigenschaftsdaten in speichern AWS IoT SiteWise, müssen Sie keine eigenen Konnektoren implementieren. Wenn Sie Ihre Modellierungs- oder Eigenschaftsdaten in anderen Datenspeichern wie Timestream, DynamoDB oder Snowflake speichern, müssen Sie AWS Lambda Konnektoren mit der AWS IoT TwinMaker Datenkonnektorschnittstelle implementieren, damit Sie Ihren Konnektor bei Bedarf aufrufen AWS IoT TwinMaker können.

Themen

- AWS IoT TwinMaker Datenkonnektoren
- AWS IoT TwinMaker Athena tabellarischer Datenkonnektor
- Entwicklung von Konnektoren für Zeitreihendaten AWS IoT TwinMaker

AWS IoT TwinMaker Datenkonnektoren

Konnektoren benötigen Zugriff auf Ihren zugrunde liegenden Datenspeicher, um gesendete Abfragen zu lösen und entweder Ergebnisse oder einen Fehler zurückzugeben.

Informationen zu den verfügbaren Konnektoren, ihren Anforderungs- und Antwortschnittstellen finden Sie in den folgenden Themen.

Weitere Informationen zu den in den Konnektorschnittstellen verwendeten Eigenschaften finden Sie in der GetPropertyValueHistoryAPI-Aktion.

> **Note**
>
> Einige Konnektoren verfügen sowohl in der Anforderungs- als auch in der Antwortschnittstelle über zwei Zeitstempelfelder für die Eigenschaften der Startzeit und der Endzeit. Beide `startDateTime` `endDateTime` verwenden eine lange Zahl, um die Epochensekunde darzustellen, was nicht mehr unterstützt wird. Aus Gründen der Abwärtskompatibilität senden wir weiterhin einen Zeitstempelwert an dieses Feld. Wir empfehlen jedoch, die

`endTime` Felder `startTime` und zu verwenden, die mit unserem API-Zeitstempelformat übereinstimmen.

Themen

- Konnektor für den Schema-Initialisierer
- DataReaderByEntity
- DataReaderByComponentType
- DataReader
- AttributePropertyValueReaderByEntity
- DataWriter
- Beispiele

Konnektor für den Schema-Initialisierer

Sie können den Schemainitialisierer im Komponententyp- oder Entitätslebenszyklus verwenden, um den Komponententyp oder die Komponenteneigenschaften aus der zugrunde liegenden Datenquelle abzurufen. Der Schemainitialisierer importiert automatisch Komponententyp- oder Komponenteneigenschaften, ohne explizit eine API-Aktion zur Einrichtung aufzurufen. `properties`

SchemaInitializer Schnittstelle anfordern

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

> **Note**
>
> Die Zuordnung der Eigenschaften in dieser Anforderungsschnittstelle ist eine`PropertyRequest`. Weitere Informationen finden Sie unter PropertyRequest.

SchemaInitializer Antwortschnittstelle

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

> **Note**
>
> Die Zuordnung der Eigenschaften in dieser Anforderungsschnittstelle ist eine`PropertyResponse`. Weitere Informationen finden Sie unter PropertyResponse.

DataReaderByEntity

DataReaderByEntity ist ein Datenebenenanschluss, der verwendet wird, um die Zeitreihenwerte von Eigenschaften in einer einzelnen Komponente abzurufen.

Informationen zu den Eigenschaftstypen, der Syntax und dem Format dieses Konnektors finden Sie in der GetPropertyValueHistoryAPI-Aktion.

DataReaderByEntitySchnittstelle anfordern

```
{
  "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"
  }
```

DataReaderByEntityAntwortschnittstelle

```
{
  "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

Verwenden Sie den Datenebenenanschluss DataReaderByEntity, um die Zeitreihenwerte gemeinsamer Eigenschaften zu ermitteln, die von demselben Komponententyp stammen. Wenn Sie beispielsweise Zeitreiheneigenschaften für den Komponententyp definieren und mehrere Komponenten diesen Komponententyp verwenden, können Sie diese Eigenschaften für alle Komponenten in einem bestimmten Zeitraum abfragen. Ein häufiger Anwendungsfall hierfür ist,

wenn Sie den Alarmstatus mehrerer Komponenten abfragen möchten, um eine globale Ansicht Ihrer Entitäten zu erhalten.

Informationen zu den Eigenschaftstypen, der Syntax und dem Format dieses Konnektors finden Sie in der GetPropertyValueHistoryAPI-Aktion.

DataReaderByComponentType Schnittstelle anfordern

```
{
  "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 Antwortschnittstelle

```
{
  "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 ist ein Datenebenenanschluss, der sowohl den Fall von DataReaderByEntity als auch verarbeiten kann DataReaderByComponentType.

Informationen zu den Eigenschaftstypen, der Syntax und dem Format dieses Konnektors finden Sie in der GetPropertyValueHistoryAPI-Aktion.

DataReader Schnittstelle anfordern

`EntityId` und `componentName` sind optional.

```
{
  "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 Antwortschnittstelle

```
{
  "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 ist ein Datenebenen-Konnektor, mit dem Sie den Wert statischer Eigenschaften in einer einzelnen Entität abrufen können.

Informationen zu den Eigenschaftstypen, der Syntax und dem Format dieses Konnektors finden Sie in der GetPropertyValueAPI-Aktion.

AttributePropertyValueReaderByEntity Schnittstelle anfordern

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

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

AttributePropertyValueReaderByEntity Antwortschnittstelle

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

DataWriter

DataWriter ist ein Datenebenenanschluss, mit dem Sie Zeitreihen-Datenpunkte für Eigenschaften in einer einzelnen Komponente in den zugrunde liegenden Datenspeicher zurückschreiben können.

Informationen zu den Eigenschaftstypen, der Syntax und dem Format dieses Konnektors finden Sie in der BatchPutPropertyValuesAPI-Aktion.

DataWriter Schnittstelle anfordern

```
{
  "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 Antwortschnittstelle

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

Beispiele

Die folgenden JSON-Beispiele sind Beispiele für die Antwort- und Anforderungssyntax für mehrere Konnektoren.

- SchemaInitializer:

  Die folgenden Beispiele zeigen den Schemainitialisierer in einem Lebenszyklus eines Komponententyps.

  Anfrage:

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

Antwort:

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

```
    },
    "myProperty2": {
      "definition": {
        "dataType": { "type": "STRING" },
        "isTimeSeries": false,
        "defaultValue": {
          "stringValue": "property2Value"
        }
      }
    }
  }
}
```

- Schemainitialisierer im Entitätslebenszyklus:

  Anfrage:

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

Antwort:

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

- DataReaderByEntity und: DataReader

  Anfrage:

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

Antwort:

```
{
  "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:

  Anfrage:

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

Antwort:

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

- DataWriter:

  Anfrage:

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

Antwort:

```
{
    "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 Athena tabellarischer Datenkonnektor

Mit dem Athena Tabular Data Connector können Sie auf Ihre Athena-Datenspeicher in zugreifen und diese verwenden. AWS IoT TwinMaker Sie können Ihre Athena-Daten verwenden, um digitale Zwillinge ohne intensiven Datenmigrationsaufwand zu erstellen. Sie können entweder den vorgefertigten Connector verwenden oder einen benutzerdefinierten Athena-Connector erstellen, um auf Daten aus Ihren Athena-Datenquellen zuzugreifen.

AWS IoT TwinMaker Voraussetzungen für den Athena-Datenkonnektor

Bevor Sie den Athena Tabular Data Connector verwenden, müssen Sie die folgenden Voraussetzungen erfüllen:

- Erstellen Sie verwaltete Athena-Tabellen und die zugehörigen Amazon S3 S3-Ressourcen. Informationen zur Verwendung von Athena finden Sie in der Athena-Dokumentation.
- Erstellen Sie einen Workspace AWS IoT TwinMaker . Sie können einen Workspace in der AWS IoT TwinMaker Konsole erstellen.
- Aktualisieren Sie Ihre Workspace-IAM-Rolle mit Athena-Berechtigungen. Weitere Informationen finden Sie unter Ändern Sie Ihre Workspace-IAM-Rolle, um den Athena-Datenkonnektor zu verwenden.
- Mache dich mit dem AWS IoT TwinMaker Entity-Component-System und der Erstellung von Entitäten vertraut. Weitere Informationen finden Sie unter Erstellen Sie Ihre erste Entität.
- Machen Sie sich mit den AWS IoT TwinMaker Datenkonnektoren vertraut. Weitere Informationen finden Sie unter AWS IoT TwinMaker Datenanschlüsse.

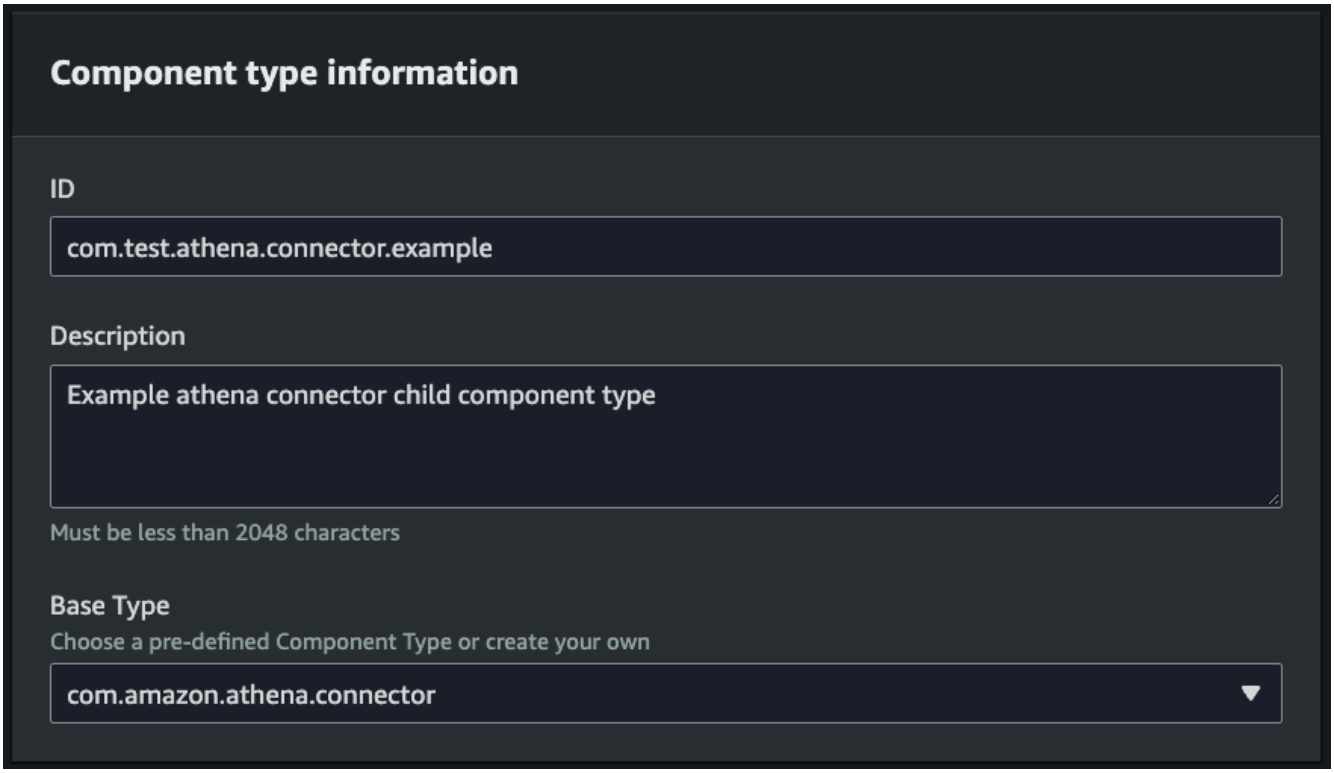
Verwenden des Athena-Datenkonnektors

Um den Athena-Datenkonnektor verwenden zu können, müssen Sie eine Komponente erstellen und dabei den Athena-Konnektor als Komponententyp verwenden. Anschließend fügen Sie die Komponente einer Entität in Ihrer Szene zur Verwendung in hinzu. AWS IoT TwinMaker

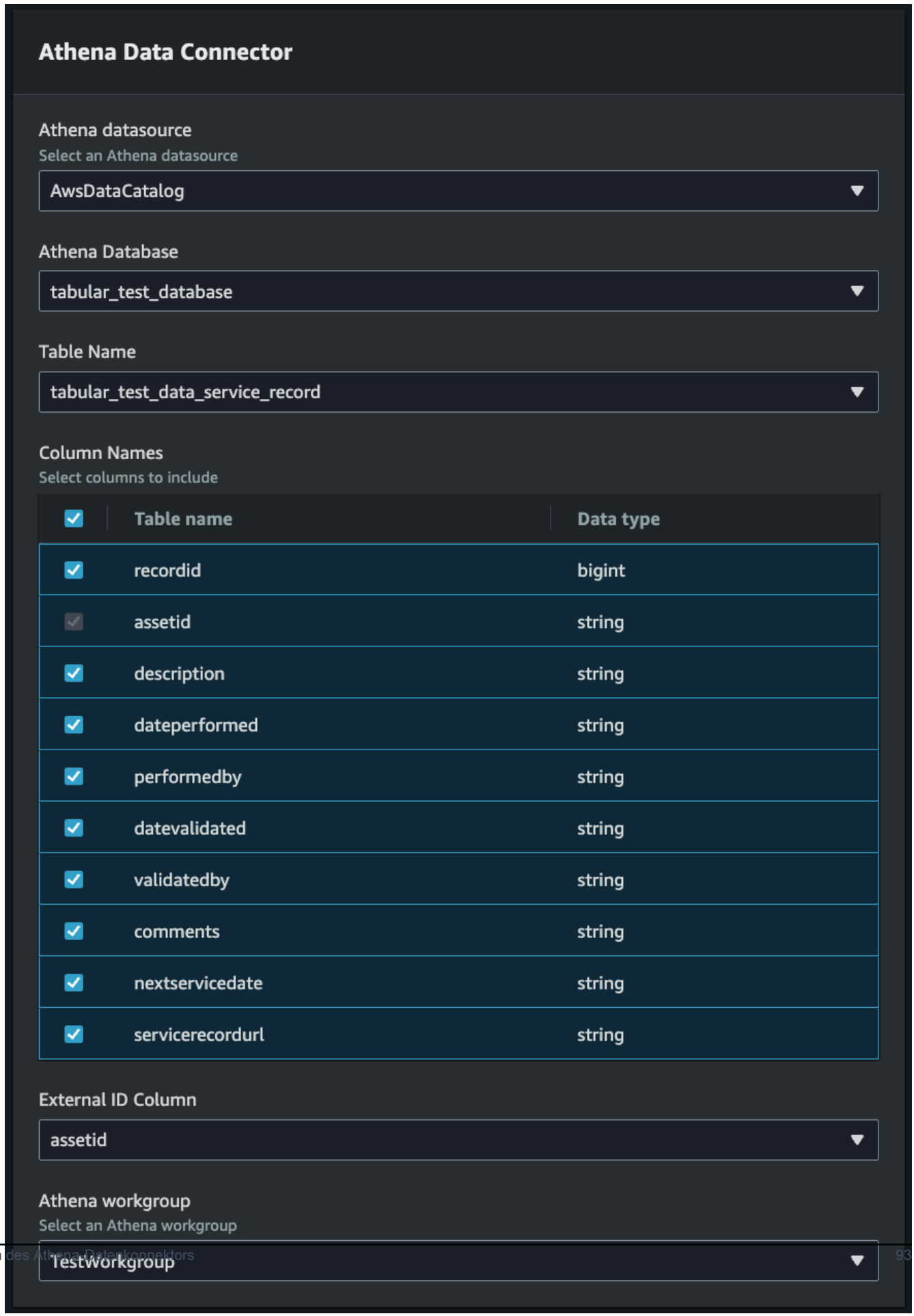
Erstellen Sie einen Komponententyp mit dem Athena-Datenkonnektor

Gehen Sie wie folgt vor, um einen AWS IoT TwinMaker Komponententyp mit dem tabellarischen Datenkonnektor von Athena zu erstellen:

1. Navigieren Sie zur AWS IoT TwinMaker -Konsole.
2. Öffnen Sie einen vorhandenen Workspace oder erstellen Sie einen neuen.
3. Wählen Sie im linken Navigationsmenü Komponententypen und anschließend Komponententyp erstellen aus, um die Seite zur Erstellung des Komponententyps zu öffnen.
4. Füllen Sie auf der Seite Komponententyp erstellen das ID-Feld mit einer ID aus, die Ihrem Anwendungsfall entspricht.



5. Wählen Sie den Basistyp. Wählen Sie aus der Drop-down-Liste den Athena Tabular Data Connector mit der Bezeichnung com.amazon.athena.connector aus.
6. Konfigurieren Sie die Datenquelle des Komponententyps, indem Sie Athena-Ressourcen für die folgenden Felder auswählen:
   - Wählen Sie eine Athena-Datenquelle.
   - Wählen Sie eine Athena-Datenbank.
   - Wählen Sie einen Tabellennamen.
   - Wählen Sie eine Athena-Arbeitsgruppe.
7. Nachdem Sie die Athena-Ressourcen ausgewählt haben, die Sie als Datenquelle verwenden möchten, wählen Sie aus, welche Spalten aus der Tabelle Sie einbeziehen möchten.
8. Wählen Sie einen externen ID-Spaltennamen aus. Wählen Sie eine Spalte aus dem vorherigen Schritt aus, die als externe ID-Spalte dienen soll. Die externe ID ist die ID, die verwendet wird, um ein Athena-Asset darzustellen und es einer AWS IoT TwinMaker Entität zuzuordnen.


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

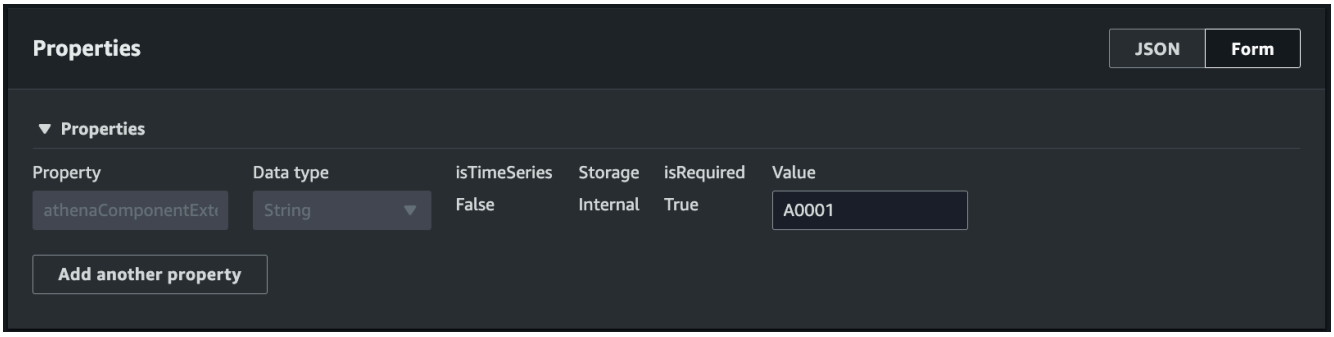
9. (Optional) Fügen Sie diesen Ressourcen AWS Tags hinzu, damit Sie sie gruppieren und organisieren können.
10. Wählen Sie Komponententyp erstellen, um die Erstellung des Komponententyps abzuschließen.

Erstellen Sie eine Komponente mit dem Athena-Datenkonnektortyp und hängen Sie sie an eine Entität an

Gehen Sie wie folgt vor, um eine AWS IoT TwinMaker Komponente mit dem Tabellendatenkonnektor von Athena zu erstellen und sie an eine Entität anzuhängen:

> **Note**
>
> Sie müssen über einen vorhandenen Komponententyp verfügen, der den tabellarischen Athena-Datenkonnektor als Datenquelle verwendet, um dieses Verfahren abschließen zu können. Sehen Sie sich das vorherige Verfahren Erstellen Sie einen Komponententyp mit dem Athena-Datenkonnektor an, bevor Sie mit dieser exemplarischen Vorgehensweise beginnen.

1. Navigieren Sie zur AWS IoT TwinMaker -Konsole.
2. Öffnen Sie einen vorhandenen Arbeitsbereich oder erstellen Sie einen neuen.
3. Wählen Sie im linken Navigationsmenü Entitäten und wählen Sie die Entität aus, der Sie die Komponente hinzufügen oder eine neue Entität erstellen möchten.
4. Erstellen Sie eine neue Entität.
5. Wählen Sie als Nächstes Komponente hinzufügen aus. , füllen Sie das Feld Komponentenname mit einem Namen aus, der Ihrem Anwendungsfall entspricht.
6. Wählen Sie im Dropdownmenü Komponententyp die Komponententyp-ID aus, die Sie im vorherigen Verfahren erstellt haben.
7. Geben Sie Komponenteninformationen und einen Komponentennamen ein und wählen Sie das zuvor ComponentType erstellte untergeordnete Objekt aus. Dies ist der, den ComponentType Sie mit dem Athena-Datenkonnektor erstellt haben.
8. Geben Sie im Abschnitt Eigenschaften die athenaComponentExternalID für die Komponente ein.



9. Wählen Sie Komponente hinzufügen, um die Erstellung der Komponente abzuschließen.

Sie haben jetzt erfolgreich eine Komponente mit dem Athena-Datenkonnektor als Komponententyp erstellt und an eine Entität angehängt.

Verwenden der JSON-Referenz für den Tabellendatenkonnektor von Athena

Das folgende Beispiel ist die vollständige JSON-Referenz für den tabellarischen Datenkonnektor von Athena. Verwenden Sie dies als Ressource, um benutzerdefinierte Datenkonnektoren und Komponententypen zu erstellen.

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

Verwenden des Athena-Datenkonnektors

Sie können Ihre Entitäten, die Athena-Tabellen verwenden, in Grafana anzeigen. Weitere Informationen finden Sie unter AWS IoT TwinMaker Grafana-Dashboard-Integration.

In der Athena-Dokumentation finden Sie Informationen zum Erstellen und Verwenden von Athena-Tabellen zum Speichern von Daten.

Fehlerbehebung beim Athena-Datenkonnektor

In diesem Thema werden häufig auftretende Probleme behandelt, die bei der Konfiguration des Athena-Datenkonnektors auftreten können.

Standort der Athena-Arbeitsgruppe:

Bei der Erstellung des Athena-Connectors ComponentType muss für eine Athena-Arbeitsgruppe der Ausgabeort eingerichtet sein. Erfahren Sie, wie Arbeitsgruppen funktionieren.

Fehlende IAM-Rollenberechtigungen:

Der Workspace-Rolle fehlt möglicherweise die Athena-API-Zugriffsberechtigung AWS IoT TwinMaker, wenn ein ComponentType erstellt, einer Entität eine Ca-Komponente hinzugefügt oder die GetPropertyValue API ausgeführt wird. Informationen zum Aktualisieren der IAM-Berechtigungen finden Sie unter Erstellen und Verwalten einer Servicerolle für. AWS IoT TwinMaker

Visualisieren Sie Athena-Tabellendaten in Grafana

Ein Grafana-Plugin ist auch verfügbar, um Ihre tabellarischen Daten auf Grafana zu visualisieren, ein Dashboard-Panel mit zusätzlichen Funktionen wie Sortierung und Filterung basierend auf ausgewählten Eigenschaften AWS IoT TwinMaker, ohne API-Aufrufe oder Interaktionen mit Athena zu tätigen. In diesem Thema erfahren Sie, wie Sie Grafana für die Visualisierung von Athena-Tabellendaten konfigurieren.

Voraussetzungen

Bevor Sie ein Grafana-Panel für die Visualisierung von Athena-Tabellendaten konfigurieren, sollten Sie die folgenden Voraussetzungen überprüfen:

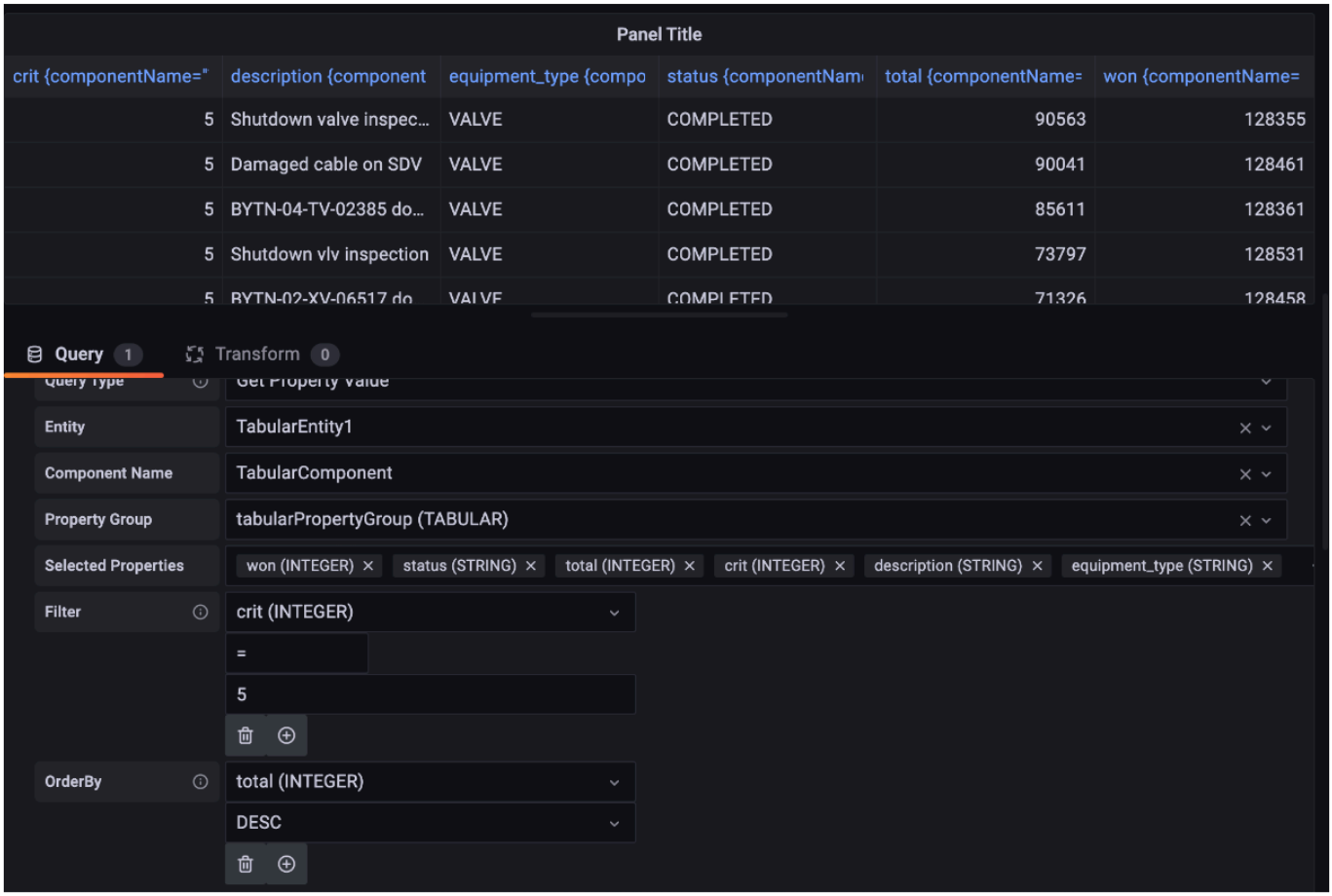
- Sie haben eine Grafana-Umgebung eingerichtet. Weitere Informationen finden Sie unter AWS IoT TwinMaker Grafana-Integration.
- Sie können eine Grafana-Datenquelle konfigurieren. Weitere Informationen finden Sie unter Grafana AWS IoT TwinMaker.
- Sie sind mit dem Erstellen eines neuen Dashboards und dem Hinzufügen eines neuen Panels vertraut.

Visualisieren Sie Athena-Tabellendaten in Grafana

Dieses Verfahren zeigt Ihnen, wie Sie ein Grafana-Panel einrichten, um Athena-Tabellendaten zu visualisieren.

1. Öffnen Sie Ihr AWS IoT TwinMaker Grafana-Dashboard.
2. Wählen Sie in den Panel-Einstellungen das Tabellenfenster aus.
3. Wählen Sie Ihre Datenquelle in der Abfragekonfiguration aus.
4. Wählen Sie die Abfrage „Eigenschaftswert abrufen“ aus.
5. Wählen Sie eine Entity aus.
6. Wählen Sie eine Komponente mit einem ComponentType aus, der den Athena-Basiskomponententyp erweitert.
7. Wählen Sie die Eigenschaftsgruppe Ihres Athena-Tisches aus.
8. Wählen Sie eine beliebige Anzahl von Eigenschaften aus der Eigenschaftsgruppe aus.
9. Konfigurieren Sie die tabellarischen Bedingungen anhand einer Liste von Filtern und Eigenschaftenreihenfolgen. Mit den folgenden Optionen:

   - Filter: Definieren Sie einen Ausdruck für einen Eigenschaftswert, um Ihre Daten zu filtern.
   - OrderBy: Geben Sie an, ob Daten für eine Eigenschaft in aufsteigender oder absteigender Reihenfolge zurückgegeben werden sollen.



Entwicklung von Konnektoren für Zeitreihendaten AWS IoT TwinMaker

In diesem Abschnitt wird erklärt, wie Sie einen Zeitreihen-Datenkonnektor in einem step-by-step Prozess entwickeln. Darüber hinaus stellen wir ein Beispiel für einen Zeitreihen-Datenkonnektor vor, der auf dem gesamten Cookie-Factory-Beispiel basiert und 3D-Modelle, Entitäten, Komponenten, Alarme und Konnektoren umfasst. Die Cookie-Factory-Beispielquelle ist im AWS IoT TwinMaker GitHub Samples-Repository verfügbar.

Themen

- AWS IoT TwinMaker Voraussetzungen für Zeitreihen-Datenkonnektoren
- Hintergrund des Datenkonnektors für Zeitreihen
- Entwicklung eines Datenkonnektors für Zeitreihen

- Verbesserung Ihres Datenkonnektors
- Testen Sie Ihren Connector
- Sicherheit
- AWS IoT TwinMaker Ressourcen erstellen
- Was kommt als Nächstes
- AWS IoT TwinMaker Beispiel für einen Zeitreihenanschluss in Cookie Factory

AWS IoT TwinMaker Voraussetzungen für Zeitreihen-Datenkonnektoren

Bevor Sie Ihren Zeitreihen-Datenkonnektor entwickeln, empfehlen wir, dass Sie die folgenden Aufgaben ausführen:

- Erstellen Sie einen AWS IoT TwinMaker Arbeitsbereich.
- Erstellen Sie AWS IoT TwinMaker Komponententypen.
- AWS IoT TwinMaker Entitäten erstellen.
- (Optional) Lesen Sie Komponententypen verwenden und erstellen.
- (Optional) Lesen Sie die AWS IoT TwinMaker Datenkonnektorschnittstelle, um sich ein allgemeines Verständnis von AWS IoT TwinMaker Datenkonnektoren zu verschaffen.

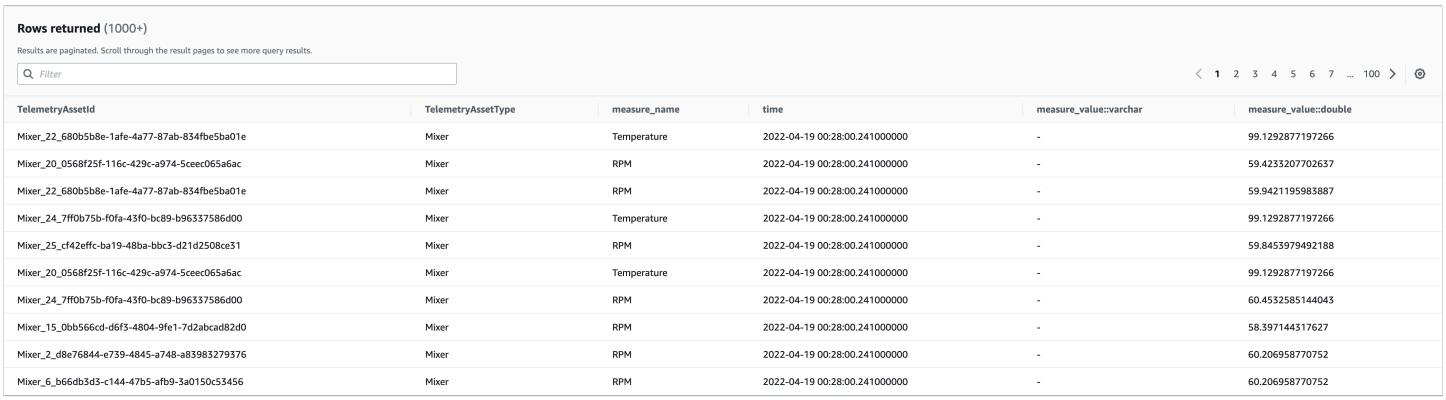
> **Note**
>
> Ein Beispiel für einen vollständig implementierten Konnektor finden Sie in unserer Cookie-Factory-Beispielimplementierung.

Hintergrund des Datenkonnektors für Zeitreihen

Stellen Sie sich vor, Sie arbeiten mit einer Fabrik zusammen, die über eine Reihe von Keksmixern und einen Wassertank verfügt. Sie möchten AWS IoT TwinMaker digitale Zwillinge dieser physischen Einheiten erstellen, sodass Sie ihren Betriebszustand anhand verschiedener Zeitreihenmetriken überwachen können.

Sie haben Sensoren vor Ort eingerichtet und streamen bereits Messdaten in eine Timestream-Datenbank. Sie möchten in der Lage sein, die Messdaten AWS IoT TwinMaker mit minimalem Aufwand anzuzeigen und zu organisieren. Sie können diese Aufgabe mithilfe eines Datenkonnektors

für Zeitreihen ausführen. Die folgende Abbildung zeigt ein Beispiel für eine Telemetrietabelle, die mithilfe eines Zeitreihenverbinders gefüllt wird.



Rows returned (1000+)

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

Filter

1 2 3 4 5 6 7 ... 100

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

Die in diesem Screenshot verwendeten Datensätze und die Timestream-Tabelle sind im Beispiel-Repository verfügbar.AWS IoT TwinMaker GitHub Sehen Sie sich auch den Cookie Factory-Beispielkonnektor für die Implementierung an, der das im vorherigen Screenshot gezeigte Ergebnis liefert.

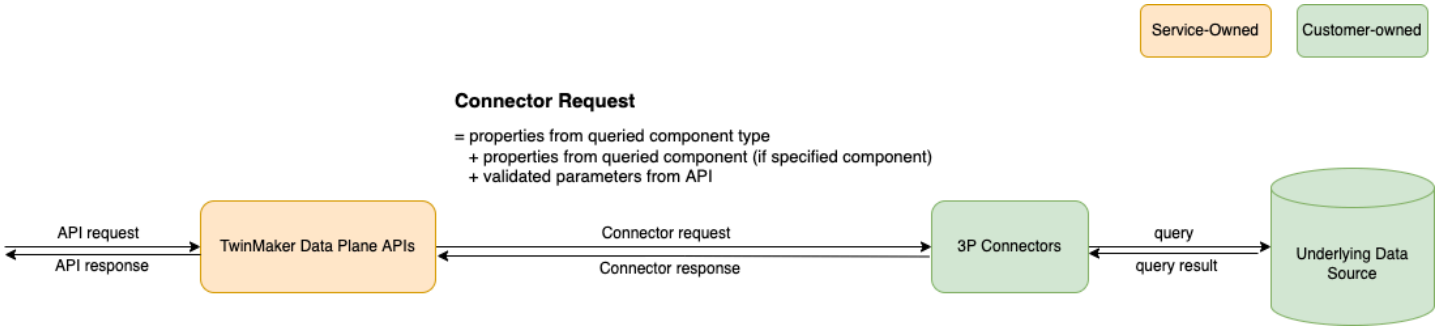
Datenfluss von Zeitreihen-Datenkonnektoren

Ruft bei Abfragen auf Datenebene AWS IoT TwinMaker die entsprechenden Eigenschaften von Komponenten und Komponententypen aus Komponenten- und Komponententypdefinitionen ab. AWS IoT TwinMaker leitet Eigenschaften zusammen mit allen API-Abfrageparametern in der Abfrage an AWS Lambda Funktionen weiter.

AWS IoT TwinMaker verwendet Lambda-Funktionen, um auf Abfragen aus Datenquellen zuzugreifen und diese zu lösen und die Ergebnisse dieser Abfragen zurückzugeben. Die Lambda-Funktionen verwenden die Komponenten- und Komponententypeigenschaften aus der Datenebene, um die ursprüngliche Anfrage aufzulösen.

Die Ergebnisse der Lambda-Abfrage werden einer API-Antwort zugeordnet und an Sie zurückgegeben.

AWS IoT TwinMaker definiert die Datenkonnektorschnittstelle und verwendet diese, um mit Lambda-Funktionen zu interagieren. Mithilfe von Datenkonnektoren können Sie Ihre Datenquelle ohne Datenmigrationsaufwand über die AWS IoT TwinMaker API abfragen. Die folgende Abbildung zeigt den grundlegenden Datenfluss, der in den vorherigen Absätzen beschrieben wurde.



Entwicklung eines Datenkonnektors für Zeitreihen

Das folgende Verfahren beschreibt ein Entwicklungsmodell, das schrittweise zu einem funktionierenden Zeitreihen-Datenkonnektor aufbaut. Die grundlegenden Schritte sind wie folgt:

1. Erstellen Sie einen gültigen Basiskomponententyp

   In einem Komponententyp definieren Sie gemeinsame Eigenschaften, die von Ihren Komponenten gemeinsam genutzt werden. Weitere Informationen zur Definition von Komponententypen finden Sie unter Komponententypen verwenden und erstellen.

   AWS IoT TwinMaker verwendet ein Modellierungsmuster für Entitätskomponenten, sodass jede Komponente einer Entität zugeordnet ist. Wir empfehlen, dass Sie jedes physische Element als Einheit modellieren und verschiedene Datenquellen mit ihren eigenen Komponententypen modellieren.

   Das folgende Beispiel zeigt einen Timestream-Vorlagenkomponententyp mit einer Eigenschaft:

```
{"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
        }
    }
}
```

Die wichtigsten Elemente des Komponententyps sind die folgenden:

- Die `telemetryId` Eigenschaft identifiziert den eindeutigen Schlüssel des physischen Elements in der entsprechenden Datenquelle. Der Datenkonnektor verwendet diese Eigenschaft als Filterbedingung, um nur Werte abzufragen, die dem angegebenen Element zugeordnet sind. Wenn Sie den `telemetryId` Eigenschaftswert in die API-Antwort der Datenebene aufnehmen, nimmt die Clientseite außerdem die ID und kann bei Bedarf eine umgekehrte Suche durchführen.
- Das `lambdaArn` Feld identifiziert die Lambda-Funktion, mit der der Komponententyp interagiert.
- Das `isRequiredInEntity` Flag erzwingt die ID-Erstellung. Dieses Flag ist erforderlich, damit bei der Erstellung der Komponente auch die ID des Elements instanziiert wird.
- Die `TelemetryId` wird dem Komponententyp als externe ID hinzugefügt, sodass das Element in der Timestream-Tabelle identifiziert werden kann.

2. Erstellen Sie eine Komponente mit dem Komponententyp

   Um den von Ihnen erstellten Komponententyp zu verwenden, müssen Sie eine Komponente erstellen und sie an die Entität anhängen, von der Sie Daten abrufen möchten. In den folgenden Schritten wird der Prozess der Erstellung dieser Komponente detailliert beschrieben:

a. Navigieren Sie zur AWS IoT TwinMaker -Konsole.

b. Wählen und öffnen Sie denselben Workspace, in dem Sie die Komponententypen erstellt haben.

c. Navigieren Sie zur Entitätsseite.

d. Erstellen Sie eine neue Entität oder wählen Sie eine vorhandene Entität aus der Tabelle aus.

e. Nachdem Sie die Entität ausgewählt haben, die Sie verwenden möchten, wählen Sie Komponente hinzufügen, um die Seite Komponente hinzufügen zu öffnen.

f. Geben Sie der Komponente einen Namen und wählen Sie als Typ den Komponententyp aus, den Sie mit der Vorlage in 1 erstellt haben. Erstellen Sie einen gültigen Basiskomponententyp.

3. Lassen Sie Ihren Komponententyp einen Lambda-Connector aufrufen

Der Lambda-Konnektor muss auf die Datenquelle zugreifen und die Abfrageanweisung basierend auf der Eingabe generieren und an die Datenquelle weiterleiten. Das folgende Beispiel zeigt eine JSON-Anforderungsvorlage, die dies tut.

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

Die wichtigsten Elemente der Anfrage:

- Dies `selectedProperties` ist eine Liste, die Sie mit den Eigenschaften füllen, für die Sie Timestream-Messungen wünschen.
- Die `endTime` Felder `startDateTimestartTime`,`EndDateTime`, und geben einen Zeitraum für die Anforderung an. Dadurch wird der Stichprobenbereich für die zurückgegebenen Messungen bestimmt.
- Das `entityId` ist der Name der Entität, von der Sie Daten abfragen.
- Das `componentName` ist der Name der Komponente, von der Sie Daten abfragen.

- Verwenden Sie das `orderByTime` Feld, um die Reihenfolge zu organisieren, in der die Ergebnisse angezeigt werden.

In der obigen Beispielanfrage würden wir erwarten, innerhalb des angegebenen Zeitfensters für das angegebene Objekt eine Reihe von Proben für die ausgewählten Immobilien in der ausgewählten Zeitreihenfolge zu erhalten. Die Antwortaussage kann wie folgt zusammengefasst werden:

```
{
  "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. Aktualisieren Sie Ihren Komponententyp, sodass er über zwei Eigenschaften verfügt

Die folgende JSON-Vorlage zeigt einen gültigen Komponententyp mit zwei Eigenschaften:

```
{
    "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. Aktualisieren Sie den Lambda-Konnektor, um die zweite Eigenschaft zu verarbeiten

   Die AWS IoT TwinMaker Datenebene-API unterstützt das Abfragen mehrerer Eigenschaften in einer einzigen Anfrage und AWS IoT TwinMaker folgt einer einzelnen Anfrage an einen Konnektor, indem sie eine Liste von bereitstellt. `selectedProperties`

   Die folgende JSON-Anfrage zeigt eine geänderte Vorlage, die jetzt eine Anfrage für zwei Eigenschaften unterstützt.

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

In ähnlicher Weise wird auch die entsprechende Antwort aktualisiert, wie im folgenden Beispiel gezeigt:

```
{
  "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**
>
> Was die Seitennummerierung für diesen Fall angeht, gilt die Seitengröße in der Anfrage für alle Eigenschaften. Das bedeutet, dass Sie bei fünf Eigenschaften in der Abfrage und einer Seitengröße von 100, wenn genügend Datenpunkte in der Quelle vorhanden sind, mit 100 Datenpunkten pro Eigenschaft rechnen müssen, was insgesamt 500 Datenpunkten entspricht.

Eine Beispielimplementierung finden Sie unter Beispiel für einen Snowflake-Connector. GitHub

Verbesserung Ihres Datenkonnektors

Umgang mit Ausnahmen

Es ist sicher, dass der Lambda-Connector Ausnahmen auslöst. Beim API-Aufruf der Datenebene wartet der AWS IoT TwinMaker Dienst darauf, dass die Lambda-Funktion eine Antwort zurückgibt. Wenn die Connector-Implementierung eine Ausnahme auslöst, wird der Ausnahmetyp in A AWS IoT TwinMaker übersetzt`ConnectorFailure`, sodass der API-Client weiß, dass innerhalb des Connectors ein Problem aufgetreten ist.

Umgang mit der Paginierung

In diesem Beispiel bietet Timestream eine Hilfsfunktion, die dabei helfen kann, die Paginierung nativ zu unterstützen. Bei einigen anderen Abfrageschnittstellen, wie z. B. SQL, kann es jedoch zusätzlichen Aufwand erfordern, einen effizienten Paginierungsalgorithmus zu implementieren.

Es gibt ein Beispiel für einen Snowflake-Connector, der die Paginierung in einer SQL-Schnittstelle verarbeitet.

Wenn das neue Token AWS IoT TwinMaker über die Connector-Antwortschnittstelle zurückgegeben wird, wird das Token verschlüsselt, bevor es an den API-Client zurückgegeben wird. Wenn das Token in einer anderen Anfrage enthalten ist, wird es AWS IoT TwinMaker entschlüsselt, bevor es an den Lambda-Connector weitergeleitet wird. Wir empfehlen, dem Token keine vertraulichen Informationen hinzuzufügen.

Testen Sie Ihren Connector

Sie können die Implementierung zwar immer noch aktualisieren, nachdem Sie den Konnektor mit dem Komponententyp verknüpft haben, wir empfehlen jedoch dringend, den Lambda-Konnektor vor der Integration mit AWS IoT TwinMaker zu überprüfen.

Es gibt mehrere Möglichkeiten, Ihren Lambda-Connector zu testen: Sie können den Lambda-Connector in der Lambda-Konsole oder lokal im testen. AWS CDK

Weitere Informationen zum Testen Ihrer Lambda-Funktionen finden Sie unter Testen von Lambda-Funktionen und Lokales Testen AWS CDK von Anwendungen.

Sicherheit

Eine Dokumentation zu bewährten Sicherheitsmethoden mit Timestream finden Sie unter Sicherheit in Timestream.

Ein Beispiel für die Verhinderung von SQL-Injection finden Sie im folgenden Python-Skript im AWS IoT TwinMaker Samples GitHub Repository.

AWS IoT TwinMaker Ressourcen erstellen

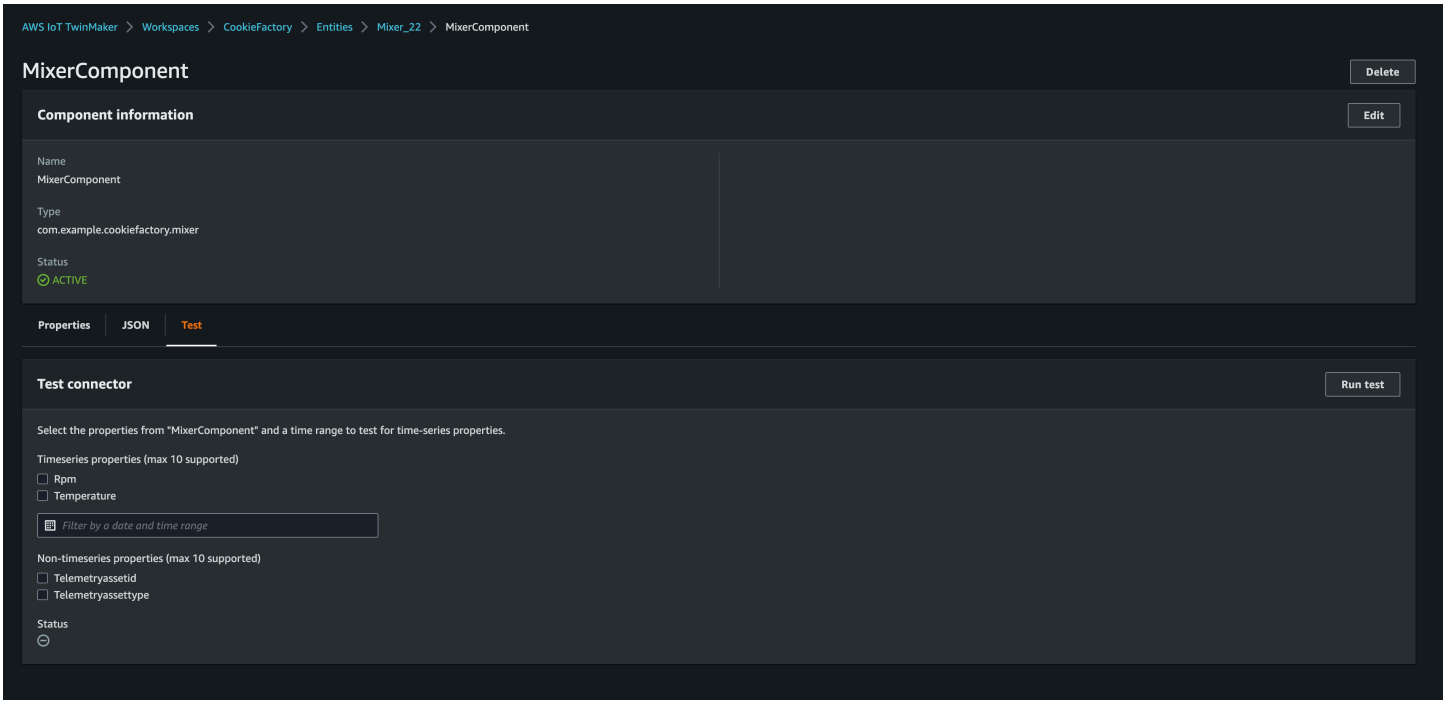
Sobald Sie die Lambda-Funktion implementiert haben, können Sie AWS IoT TwinMaker Ressourcen wie Komponententypen, Entitäten und Komponenten über die AWS IoT TwinMaker Konsole oder API erstellen.

> **Note**
>
> Wenn Sie die Setup-Anweisungen im GitHub Beispiel befolgen, sind alle AWS IoT TwinMaker Ressourcen automatisch verfügbar. Sie können die Komponententypdefinitionen im AWS IoT

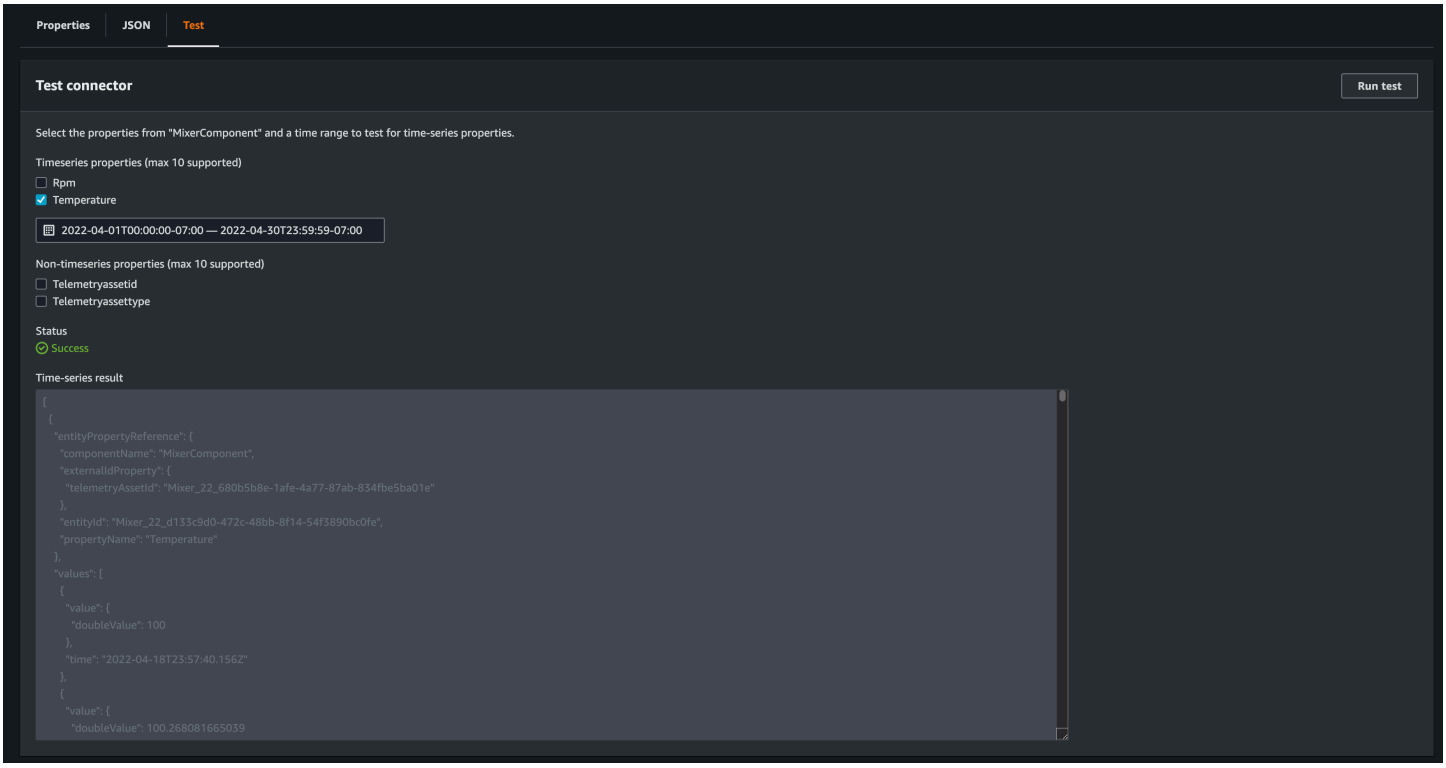
TwinMaker GitHub Beispiel überprüfen. Sobald der Komponententyp von einer Komponente verwendet wird, können die Eigenschaftsdefinitionen und Funktionen des Komponententyps nicht mehr aktualisiert werden.

Integrationstests

Wir empfehlen, einen integrierten Test durchzuführen AWS IoT TwinMaker , um zu überprüfen, ob die Datenebenenabfrage funktioniert end-to-end. Sie können dies über die GetPropertyValueHistoryAPI oder einfach in der AWS IoT TwinMaker Konsole durchführen.



Gehen Sie in der AWS IoT TwinMaker Konsole zu den Komponentendetails. Dort werden unter Test alle Eigenschaften der Komponente aufgelistet. Im Testbereich der Konsole können Sie sowohl Eigenschaften als non-time-series auch Eigenschaften von Zeitreihen testen. Für Zeitreiheneigenschaften können Sie auch die GetPropertyValueHistoryAPI und für non-time-series Eigenschaften die API verwenden GetPropertyValue. Wenn Ihr Lambda-Connector die Abfrage mehrerer Eigenschaften unterstützt, können Sie mehr als eine Eigenschaft auswählen.



Was kommt als Nächstes

Sie können jetzt ein AWS IoT TwinMaker Grafana-Dashboard zur Visualisierung von Metriken einrichten. Sie können auch andere Beispiele für Datenkonnektoren im AWS IoT TwinMaker GitHub Beispiel-Repository untersuchen, um festzustellen, ob sie zu Ihrem Anwendungsfall passen.

AWS IoT TwinMaker Beispiel für einen Zeitreihenanschluss in Cookie Factory

Der vollständige Code der Cookie-Factory-Lambda-Funktion ist verfügbar unter GitHub. Sie können die Implementierung zwar immer noch aktualisieren, nachdem Sie den Konnektor mit dem Komponententyp verknüpft haben, wir empfehlen jedoch dringend, den Lambda-Konnektor vor der Integration mit AWS IoT TwinMaker zu überprüfen. Sie können Ihre Lambda-Funktion in der Lambda-Konsole oder lokal in der testen. AWS CDK Weitere Informationen zum Testen Ihrer Lambda-Funktionen finden Sie unter Testen von Lambda-Funktionen und Lokales Testen AWS CDK von Anwendungen.

Beispiele für Cookie-Factory-Komponententypen

In einem Komponententyp definieren wir gemeinsame Eigenschaften, die von allen Komponenten gemeinsam genutzt werden. Für das Cookie-Factory-Beispiel haben physische Komponenten

desselben Typs dieselben Abmessungen, sodass wir das Messschema im Komponententyp definieren können. Als Beispiel wird der Mischertyp im folgenden Beispiel definiert.

```
{
    "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
        }
    }
}
```

Beispielsweise kann eine physische Komponente Messungen in einer Timestream-Datenbank, Wartungsaufzeichnungen in einer SQL-Datenbank oder Alarmdaten in Alarmsystemen enthalten. Wenn Sie mehrere Komponenten erstellen und sie einer Entität zuordnen, werden verschiedene Datenquellen mit der Entität verknüpft und das Diagramm mit den Entitätskomponenten gefüllt. In diesem Zusammenhang benötigt jede Komponente eine `telemetryId` Eigenschaft, um den eindeutigen Schlüssel der Komponente in der entsprechenden Datenquelle zu identifizieren. Die Angabe der `telemetryId` Eigenschaft hat zwei Vorteile: Die Eigenschaft kann im Datenkonnektor als Filterbedingung verwendet werden, um nur Werte der angegebenen Komponente abzufragen. Wenn Sie den `telemetryId` Eigenschaftswert in die API-Antwort der Datenebene aufnehmen, nimmt der Client die ID und kann bei Bedarf eine umgekehrte Suche durchführen.

Wenn Sie `TelemetryId` dem Komponententyp die als externe ID hinzufügen, identifiziert sie die Komponente in der `TimeStream` Tabelle.

```
{
    "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
        }
    }
}
```

In ähnlicher Weise haben wir den Komponententyp für`WaterTank`, wie im folgenden JSON-Beispiel gezeigt.

```
{
  "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
    }
  }
}
```

Das `TelemetryType` ist eine optionale Eigenschaft im Komponententyp, wenn sie darauf abzielt, Eigenschaftswerte im Entitätsbereich abzufragen. Ein Beispiel finden Sie in den definierten Komponententypen im AWS IoT TwinMaker GitHub Beispiel-Repository. Es gibt auch Alarmtypen, die in dieselbe Tabelle eingebettet sind, sodass die definiert `TelemetryType` ist und Sie gemeinsame Eigenschaften wie das `TelemetryId` und `TelemetryType` in einen übergeordneten Komponententyp extrahieren, damit andere untergeordnete Typen sie gemeinsam nutzen können.

Beispiel Lambda

Der Lambda-Konnektor muss auf die Datenquelle zugreifen und die Abfrageanweisung basierend auf der Eingabe generieren und an die Datenquelle weiterleiten. Eine Beispielanforderung, die an das Lambda gesendet wurde, wird im folgenden JSON-Beispiel gezeigt.

```
{
    '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'
}
```

Das Ziel der Lambda-Funktion besteht darin, historische Messdaten für eine bestimmte Entität abzufragen. AWS IoT TwinMaker stellt eine Zuordnung der Komponenteneigenschaften bereit, und Sie sollten einen instanziierten Wert für die Komponenten-ID angeben. Um beispielsweise die Abfrage auf Komponententypebene zu bearbeiten (was bei Alarmanwendungsfällen üblich ist) und den Alarmstatus aller Komponenten im Workspace zurückzugeben, enthält die Eigenschaftenübersicht Definitionen der Komponententypeigenschaften.

Im einfachsten Fall benötigen wir, wie in der vorherigen Anfrage, eine Reihe von Temperaturmesswerten für die jeweilige Komponente während des angegebenen Zeitfensters in aufsteigender Reihenfolge. Die Abfrageanweisung kann wie folgt zusammengefasst werden:

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

AWS IoT TwinMaker Szenen erstellen und bearbeiten

Szenen sind dreidimensionale Visualisierungen Ihres digitalen Zwillings. Sie sind die wichtigste Methode für Sie, Ihren digitalen Zwilling zu bearbeiten. Erfahren Sie, wie Sie Ihrer Szene Alarme, Zeitreihendaten, Farbüberlagerungen, Tags und visuelle Regeln hinzufügen, um Ihre digitalen Zwillingsvisualisierungen an Ihren realen Anwendungsfall anzupassen.

Dieser Abschnitt deckt die folgenden Themen ab:

- Bevor Sie Ihre erste Szene erstellen
- Laden Sie Ressourcen in die AWS IoT TwinMaker Ressourcenbibliothek hoch
- Erstelle deine Szenen
- Fügt Objekten feste Kameras hinzu
- Bearbeitung mit verbesserter Szene
- Bearbeiten Sie Ihre Szenen
- Modellformat für 3D-Kacheln
- Dynamische Szenen

Bevor Sie Ihre erste Szene erstellen

Szenen sind auf Ressourcen angewiesen, um Ihren digitalen Zwilling darzustellen. Diese Ressourcen bestehen aus 3D-Modellen, Daten oder Texturdateien. Die Größe und Komplexität Ihrer Ressourcen, Elemente in der Szene wie Beleuchtung und Ihre Computerhardware wirken sich auf die Leistung von AWS IoT TwinMaker Szenen aus. Verwenden Sie die Informationen in diesem Thema, um Verzögerungen und Ladezeiten zu reduzieren und die Bildrate Ihrer Szenen zu verbessern.

Optimieren Sie Ihre Ressourcen, bevor Sie sie importieren AWS IoT TwinMaker

Sie können AWS IoT TwinMaker damit in Echtzeit mit Ihrem digitalen Zwilling interagieren. Für ein optimales Erlebnis mit Ihren Szenen empfehlen wir, Ihre Ressourcen für den Einsatz in einer Echtzeitumgebung zu optimieren.

Ihre 3D-Modelle können erhebliche Auswirkungen auf die Leistung haben. Komplexe Modellgeometrien und Netze können die Leistung beeinträchtigen. Beispielsweise weisen industrielle CAD-Modelle einen hohen Detaillierungsgrad auf. Wir empfehlen, die Netze dieser Modelle zu

komprimieren und ihre Polygonanzahl zu reduzieren, bevor Sie sie in AWS IoT TwinMaker Szenen verwenden. Wenn Sie neue 3D-Modelle für erstellen AWS IoT TwinMaker, sollten Sie einen bestimmten Detaillierungsgrad festlegen und diesen für alle Ihre Modelle beibehalten. Entfernen Sie Details aus Modellen, die sich nicht auf die Visualisierung oder Interpretation Ihres Anwendungsfalls auswirken.

Verwenden Sie Open-Source-Tools zur Netzkomprimierung wie DRACO 3D-Datenkomprimierung, um Modelle zu komprimieren und die Dateigröße zu reduzieren.

Nicht optimierte Texturen können sich auch auf die Leistung auswirken. Wenn Sie keine Transparenz in Ihren Texturen benötigen, sollten Sie das PEG-Bildformat dem PNG-Format vorziehen. Sie können Ihre Texturdateien komprimieren, indem Sie Open-Source-Tools zur Texturkomprimierung wie Basis Universal Texture Compression verwenden.

Bewährte Methoden für die Leistung in AWS IoT TwinMaker

Beachten Sie die folgenden Einschränkungen und bewährten Methoden AWS IoT TwinMaker, um eine optimale Leistung bei zu erzielen.

- AWS IoT TwinMaker Die Leistung beim Rendern von Szenen hängt von der Hardware ab. Die Leistung variiert je nach Computerhardwarekonfiguration.
- Wir empfehlen eine Gesamtanzahl von Polygonen von unter 1 Million für alle Ihre Objekte in Ihrem AWS IoT TwinMaker.
- Wir empfehlen insgesamt 200 Objekte pro Szene. Wenn Sie die Anzahl der Objekte in einer Szene auf über 200 erhöhen, kann sich die Bildrate Ihrer Szene verringern.
- Wir empfehlen, dass die Gesamtgröße aller einzigartigen 3D-Assets in Ihrer Szene 100 Megabyte nicht überschreitet. Andernfalls kann es je nach Browser und Hardware zu langsamen Ladezeiten oder Leistungseinbußen kommen.
- Szenen verfügen standardmäßig über Umgebungsbeleuchtung. Sie können einer Szene zusätzliche Lichter hinzufügen, um bestimmte Objekte in den Fokus zu rücken oder Schatten auf Objekte zu werfen. Wir empfehlen, ein Licht pro Szene zu verwenden. Verwenden Sie bei Bedarf Lichter und vermeiden Sie es, reale Lichter innerhalb einer Szene zu replizieren.

Weitere Informationen

Verwenden Sie diese Ressourcen, um mehr über Optimierungstechniken zu erfahren, mit denen Sie die Leistung in Ihren Szenen verbessern können.

- Wie konvertiert und komprimiert man OBJ-Modelle in GLTF zur Verwendung mit AWS IoT TwinMaker
- Optimieren Sie Ihre 3D-Modelle für Webinhalte
- Optimieren von Szenen für eine bessere WebGL-Leistung

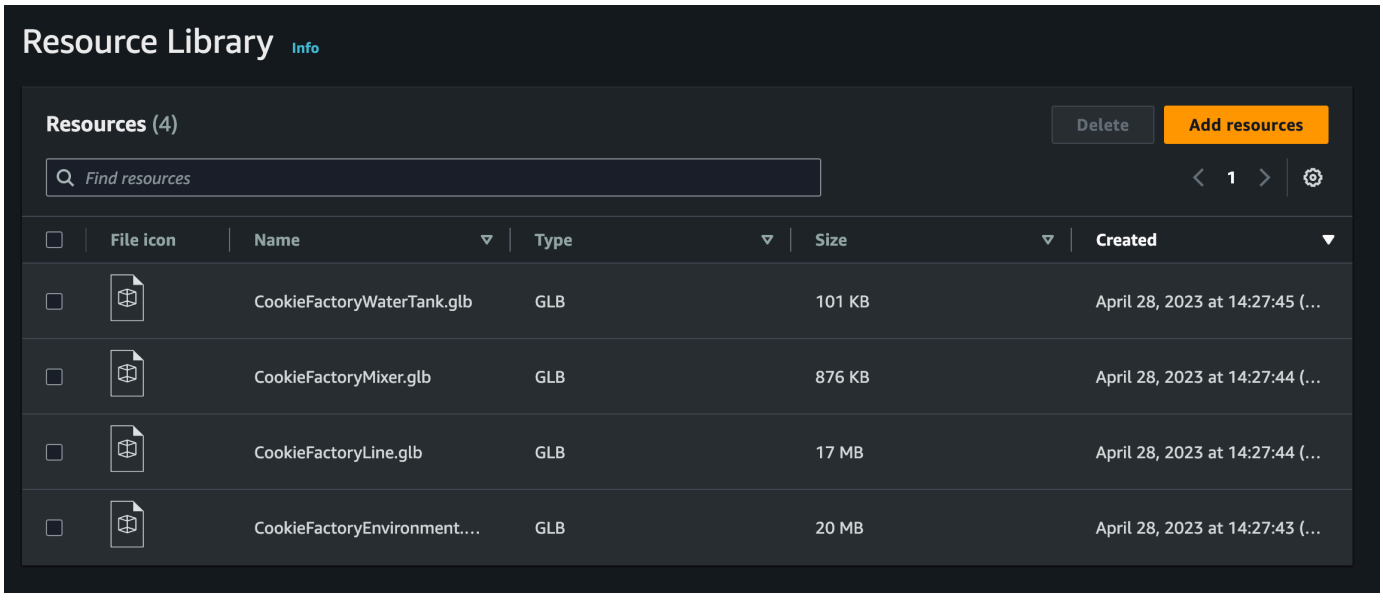
Laden Sie Ressourcen in die AWS IoT TwinMaker Ressourcenbibliothek hoch

Sie können die Ressourcenbibliothek verwenden, um alle Ressourcen zu steuern und zu verwalten, die Sie in Szenen für Ihre digitale Zwillingsanwendung platzieren möchten. Um auf die Ressourcen AWS IoT TwinMaker aufmerksam zu machen, laden Sie sie über die Konsolenseite der Ressourcenbibliothek hoch.

Laden Sie Dateien mithilfe der Konsole in die Ressourcenbibliothek hoch

Gehen Sie wie folgt vor, um mithilfe der AWS IoT TwinMaker Konsole Dateien zur Ressourcenbibliothek hinzuzufügen.

1. Wählen Sie im linken Navigationsmenü unter Arbeitsbereiche die Option Ressourcenbibliothek aus.
2. Wählen Sie Ressourcen hinzufügen und wählen Sie die Dateien aus, die Sie hochladen möchten.



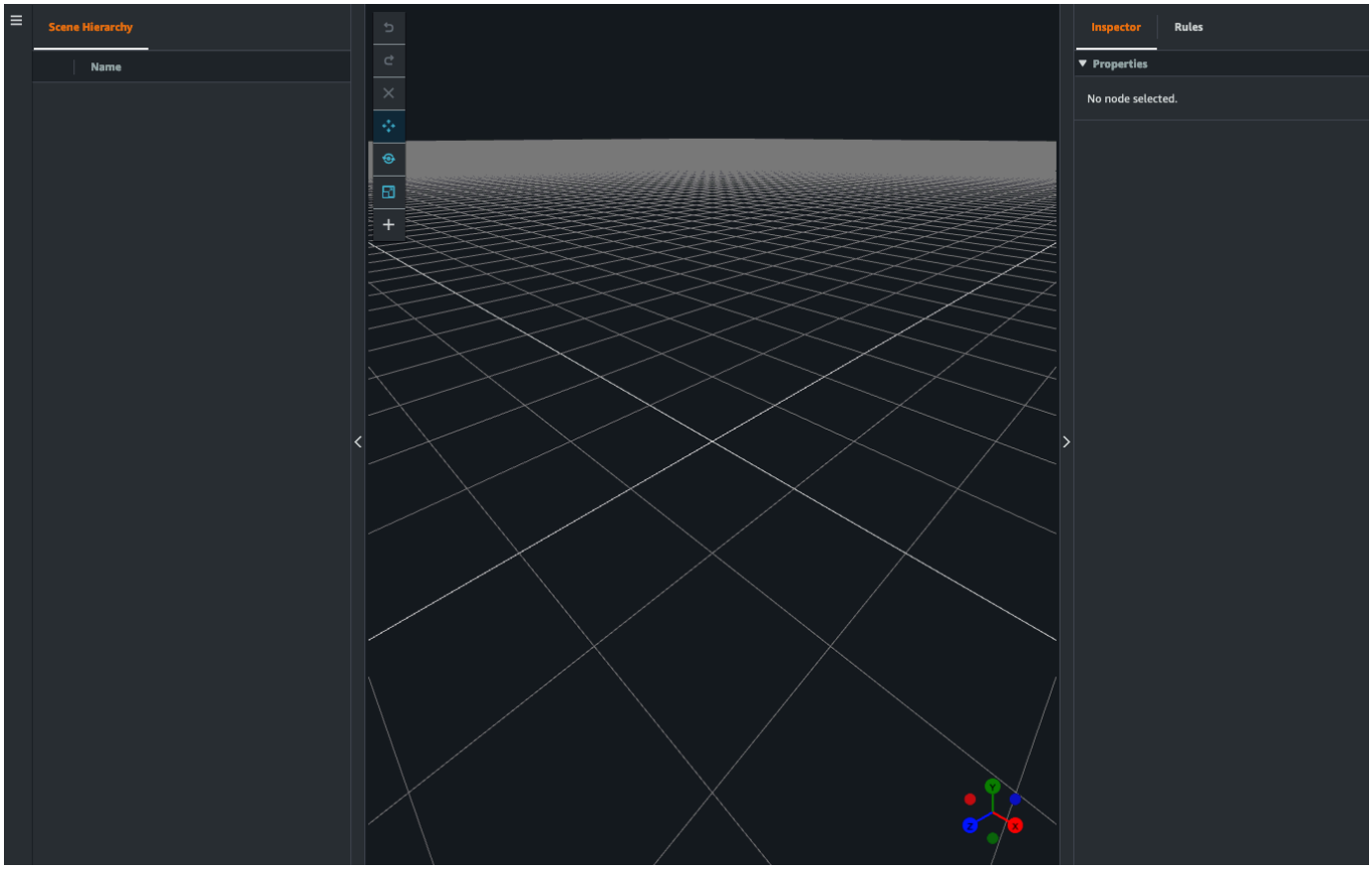
Erstelle deine Szenen

In diesem Abschnitt richten Sie eine Szene ein, damit Sie Ihren digitalen Zwilling bearbeiten können. Sie können ein 3D-Modell importieren, das in die Ressourcenbibliothek hochgeladen wurde, dann Widgets hinzufügen und Eigenschaftsdaten an Objekte binden, um Ihren digitalen Zwilling zu vervollständigen. Szenenobjekte können ein ganzes Gebäude oder einen ganzen Raum oder einzelne Geräte umfassen, die an ihrem physischen Standort positioniert sind.

> **Note**
>
> Bevor Sie eine Szene erstellen, müssen Sie einen Arbeitsbereich erstellen.

Gehen Sie wie folgt vor, um Ihre Szene in zu erstellen AWS IoT TwinMaker.

1. Um den Szenenbereich zu öffnen, wählen Sie in der linken Navigationsleiste Ihres Workspace die Option Szenen aus.
2. Wählen Sie Create scene (Szene erstellen) aus. Das Fenster zur neuen Szenenerstellung wird geöffnet.
3. Geben Sie im Bereich zur Szenenerstellung einen Namen und eine Beschreibung für Ihre neue Szene ein. Wenn Sie über einen Standard- oder ein gestaffeltes Paket verfügen, können Sie Ihren Szenentyp auswählen. Es wird empfohlen, eine dynamische Szene zu verwenden.
4. Wenn Sie bereit sind, die Szene zu erstellen, wählen Sie Szene erstellen. Die neue Szene wird geöffnet und Sie können damit arbeiten.

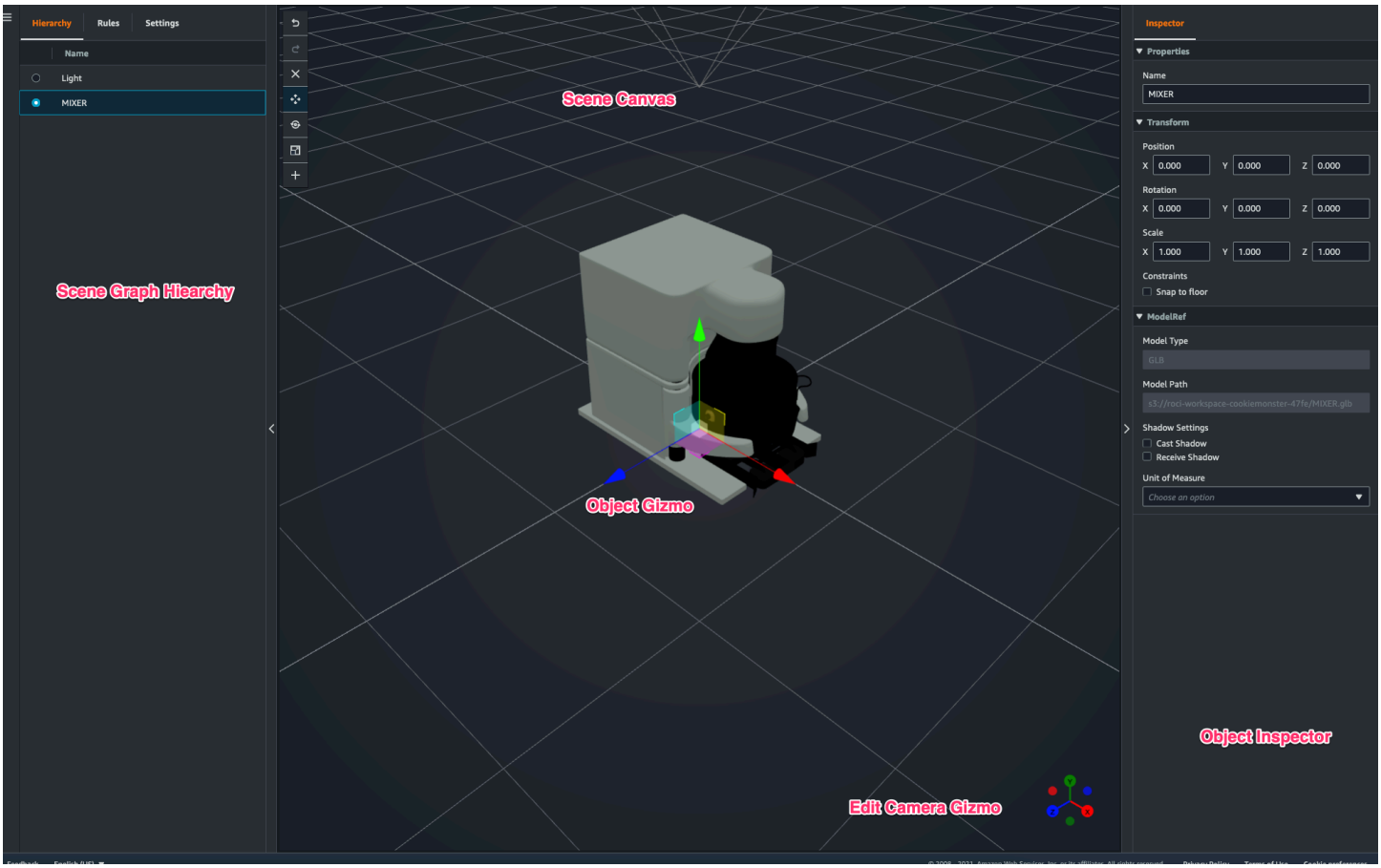


Verwenden Sie die 3D-Navigation AWS IoT TwinMaker in Ihren In-Szenen

Die AWS IoT TwinMaker Szene verfügt über eine Reihe von Navigationssteuerungen, mit denen Sie effizient durch den 3D-Raum Ihrer Szene navigieren können. Um mit dem 3D-Raum und den in Ihrer Szene dargestellten Objekten zu interagieren, verwenden Sie die folgenden Widgets und Menüoptionen.

- Inspector: Verwenden Sie das Inspektor-Fenster, um Eigenschaften und Einstellungen einer ausgewählten Entität oder Komponente in Ihrer Hierarchie anzuzeigen und zu bearbeiten.
- Scene Canvas: Die Scene Canvas ist der 3D-Raum, in dem Sie alle 3D-Ressourcen, die Sie verwenden möchten, positionieren und ausrichten können.
- Hierarchie des Szenendiagramms: In diesem Bereich können Sie sich alle in Ihrer Szene vorhandenen Entitäten anzeigen lassen. Es erscheint auf der linken Seite des Fensters.
- Objekt-Gizmo: Verwenden Sie dieses Gizmo, um Objekte auf der Leinwand zu bewegen. Es wird in der Mitte eines ausgewählten 3D-Objekts im Scene Canvas angezeigt.

- Kamera-Gizmo bearbeiten: Verwenden Sie das Kamera-Gizmo bearbeiten, um schnell die aktuelle Ausrichtung der Szenenansichtskamera anzuzeigen und den Betrachtungswinkel zu ändern. Sie finden dieses Gizmo in der unteren rechten Ecke der Szenenansicht.
- Zoom-Steuerelemente: Um im Scene Canvas zu navigieren, klicken Sie mit der rechten Maustaste und ziehen Sie in die Richtung, in die Sie sich bewegen möchten. Um zu drehen, klicken Sie mit der linken Maustaste und ziehen Sie, um zu drehen. Verwenden Sie zum Zoomen das Scrollrad an der Maus oder drücken Sie die Finger auf dem Trackpad Ihres Laptops zusammen und bewegen Sie sie auseinander.



Für die Szenenschaltflächen im Hierarchiebereich sind die folgenden Funktionen in der Reihenfolge ihres Layouts aufgeführt:

- Rückgängig machen: Macht Ihre letzte Änderung in der Szene rückgängig.
- Wiederherstellen: Mache deine letzte Änderung in der Szene wieder her.

- Plus (+): Verwenden Sie diese Schaltfläche, um Zugriff auf die folgenden Aktionen zu erhalten: Leeren Knoten hinzufügen, 3D-Modell hinzufügen, Tag hinzufügen, Licht hinzufügen und Modell-Shader hinzufügen.
- Navigationsmethode ändern: Greifen Sie auf die Navigationsoptionen der Szenenkamera, Orbit und Pan, zu.
- Papierkorb (löschen): Verwenden Sie diese Schaltfläche, um ein ausgewähltes Objekt in Ihrer Szene zu löschen.
- Werkzeuge zur Objektmanipulation: Verwenden Sie diese Schaltfläche, um das ausgewählte Objekt zu verschieben, zu drehen und zu skalieren.

Fügt Objekten feste Kameras hinzu

Sie können Ihren Entitäten in Ihren AWS IoT TwinMaker Szenen feste Kameraansichten hinzufügen. Diese Kameras bieten eine feste Perspektive auf ein 3D-Modell, sodass Sie Ihre Perspektive in einer Szene schnell und einfach auf eine Zieleinheit umstellen können.

1. Navigieren Sie in der AWS IoT TwinMaker Konsole zu Ihrer Szene.
2. Wählen Sie im Szenenhierarchie-Menü die Entität aus, an die Sie die Kamera anhängen möchten.
3. Drücken Sie die Taste + und wählen Sie aus den Drop-down-Optionen die Option Kamera aus aktueller Ansicht hinzufügen aus. Um eine Kamera mit der aktuellen Perspektive auf das Objekt anzuwenden.
4. Im Inspektor können Sie Ihre Kamera konfigurieren und die folgenden Einstellungen vornehmen:
   - Ein Kameraname
   - Die Position und Drehung der Kamera
   - Die Brennweite der Kamera
   - Die Zoomstufe
   - Schnittebenen in der Nähe und in der Ferne
5. Um auf Ihre Kamera zuzugreifen, nachdem Sie sie platziert haben. Wählen Sie die Entität, zu der Sie die Kamera hinzugefügt haben, in der Hierarchie aus. Suchen Sie nach dem Kameranamen, der unter der Entität aufgeführt ist.
6. Sobald Sie die platzierte Kamera aus Ihrer Entität ausgewählt haben, wird die Kameraansicht der Szene auf die eingestellte Perspektive der platzierten Kamera ausgerichtet.

Bearbeitung mit verbesserter Szene

AWS IoT TwinMaker Szenen verfügen über eine Reihe von Werkzeugen zur Verbesserung sowie Bearbeitung und Manipulation der in Ihrer Szene vorhandenen Ressourcen.

In den folgenden Themen erfahren Sie, wie Sie die erweiterten Bearbeitungsfunktionen in Ihren AWS IoT TwinMaker Szenen verwenden.

- Gezielte Platzierung von Szenenobjekten
- Auswahl eines Untermodells
- Bearbeiten Sie Entitäten in der Szenenhierarchie

Gezielte Platzierung von Szenenobjekten

AWS IoT TwinMaker ermöglicht es Ihnen, Objekte präzise zu platzieren und zu Ihrer Szene hinzuzufügen. Diese erweiterte Bearbeitungsfunktion gibt Ihnen mehr Kontrolle darüber, wo Sie Tags, Objekte, Lichter und Modelle in Ihrer Szene platzieren.

1. Navigieren Sie in der AWS IoT TwinMaker Konsole zu Ihrer Szene.
2. Drücken Sie die Taste + und wählen Sie aus den Drop-down-Optionen eine der Optionen aus. Dies kann ein Modell, eine Leuchte, ein Etikett oder etwas anderes aus dem Menü „+“ sein.

   Wenn Sie den Cursor im 3D-Raum Ihrer Szene bewegen, sollten Sie ein Ziel um Ihren Cursor herum sehen.
3. Verwenden Sie das Ziel, um Elemente präzise in Ihrer Szene zu platzieren.

Auswahl eines Untermodells

AWS IoT TwinMaker ermöglicht es Ihnen, Untermodelle von 3D-Modellen in Szenen auszuwählen und ihnen Standardeigenschaften wie Beschriftungen, Lichter oder Regeln zuzuweisen.

3D-Modelldateiformate enthalten Metadaten, mit denen Teilbereiche des Modells als Untermodelle innerhalb des größeren Modells angegeben werden können. Ein Modell könnte beispielsweise ein Filtrationssystem sein. Einzelne Teile des Systems wie Tanks, Rohre oder ein Motor sind als Untermodelle des 3D-Modells der Filtration gekennzeichnet.

Unterstützte 3D-Dateiformate in Szenen: GLB und GLTF.

1. Navigieren Sie in der Konsole zu Ihrer Szene.AWS IoT TwinMaker
2. Wenn Sie in Ihrer Szene keine Modelle haben, stellen Sie sicher, dass Sie eines hinzufügen, indem Sie die Option aus dem Menü + auswählen.
3. Wählen Sie das Modell aus, das in Ihrer Szenenhierarchie aufgeführt ist. Sobald Sie es ausgewählt haben, sollte die Hierarchie alle Untermodelle unter dem Modell anzeigen.

   > **Note**
   >
   > Wenn keine Untermodelle aufgeführt sind, wurde das Modell wahrscheinlich nicht so konfiguriert, dass es keine Untermodelle enthält.

4. Um die Sichtbarkeit eines Untermodells umzuschalten, drücken Sie auf das Augensymbol, das sich in der Hierarchie rechts neben dem Namen des Untermodells befindet.
5. Um Daten eines Untermodells zu bearbeiten, z. B. seinen Namen oder seine Position, öffnet sich der Szeneninspektor, wenn ein Untermodell ausgewählt wird. Verwenden Sie das Inspektormenü, um die Daten des Submodells zu aktualisieren oder zu ändern.
6. Um Untermodellen Beschriftungen, Lichter, Regeln oder andere Eigenschaften hinzuzufügen, drücken Sie die Taste +, während das Untermodell in der Hierarchie ausgewählt ist.

Bearbeiten Sie Entitäten in der Szenenhierarchie

AWS IoT TwinMaker Mit Szenen können Sie Eigenschaften von Entitäten in der Hierarchietabelle direkt bearbeiten. Das folgende Verfahren zeigt Ihnen, welche Aktionen Sie über das Hierarchie-Menü an einer Entität ausführen können.

1. Navigieren Sie in der AWS IoT TwinMaker Konsole zu Ihrer Szene.
2. Öffnen Sie die Szenenhierarchie und wählen Sie ein Unterelement einer Entität aus, die Sie manipulieren möchten.
3. Sobald das Element ausgewählt ist, drücken Sie die Taste + und wählen Sie aus dem Drop-down-Menü eine der Optionen aus:

   - Leeren Knoten hinzufügen
   - 3D-Modell hinzufügen
   - Licht hinzufügen
   - Kamera aus der aktuellen Ansicht hinzufügen

- Tag hinzufügen
- Modell-Shader hinzufügen
- Bewegungsanzeige hinzufügen

4. Nachdem Sie eine der Optionen aus dem Drop-down-Menü ausgewählt haben, wird die Auswahl als untergeordnetes Element des ausgewählten Elements aus Schritt 2 auf die Szene angewendet.
5. Sie können untergeordnete Elemente und übergeordnete Elemente neu anordnen, indem Sie ein untergeordnetes Element auswählen und es in der Hierarchie auf ein neues übergeordnetes Element ziehen.

Fügen Sie Anmerkungen zu Entitäten hinzu

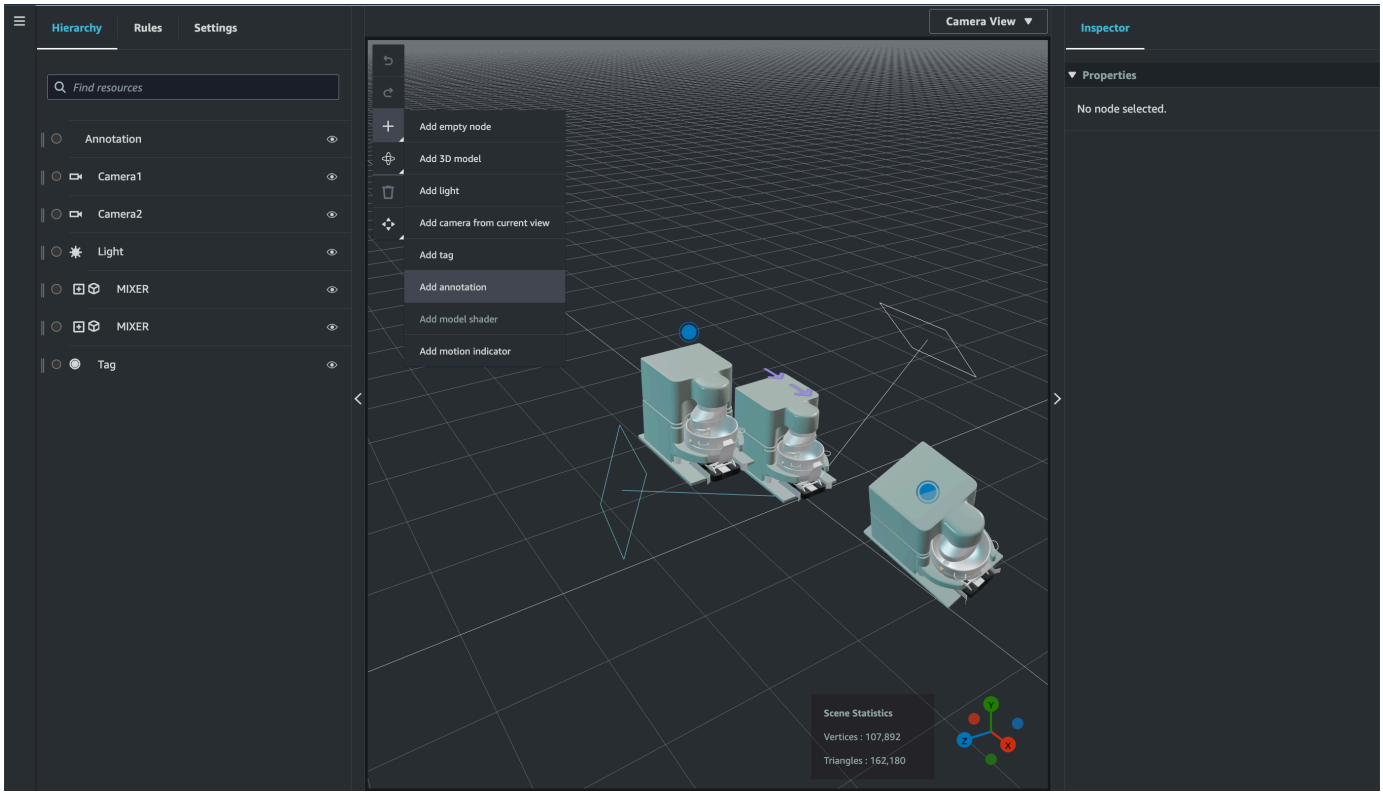
Mit dem AWS IoT TwinMaker Scene Composer können Sie jedes Element in Ihrer Szenenhierarchie kommentieren. Die Anmerkung wurde in Markdown verfasst.

Weitere Informationen zum Schreiben in Markdown finden Sie in der offiziellen Dokumentation zur Markdown-Syntax, Basic Syntax.

> **Note**
>
> AWS IoT TwinMaker Nur Anmerkungen und Overlay-Markdown-Syntax und kein HTML.

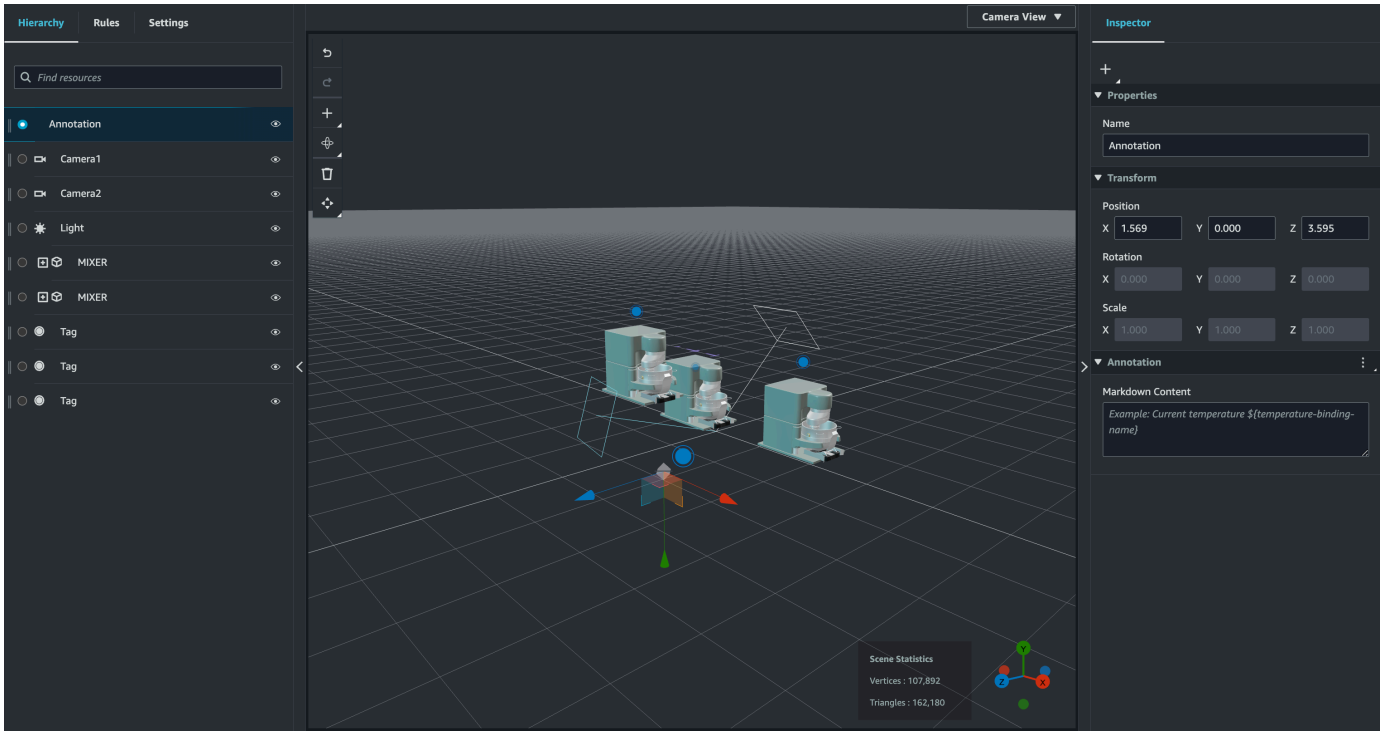
Fügen Sie einer Entität eine Anmerkung hinzu

1. Navigieren Sie in der AWS IoT TwinMaker Konsole zu Ihrer Szene.
2. Wählen Sie ein Element aus der Szenenhierarchie aus, das Sie mit Anmerkungen versehen möchten. Wenn kein Element in der Hierarchie ausgewählt ist, können Sie dem Stamm eine Anmerkung hinzufügen.
3. Drücken Sie das Pluszeichen + und wählen Sie die Option Anmerkung hinzufügen.

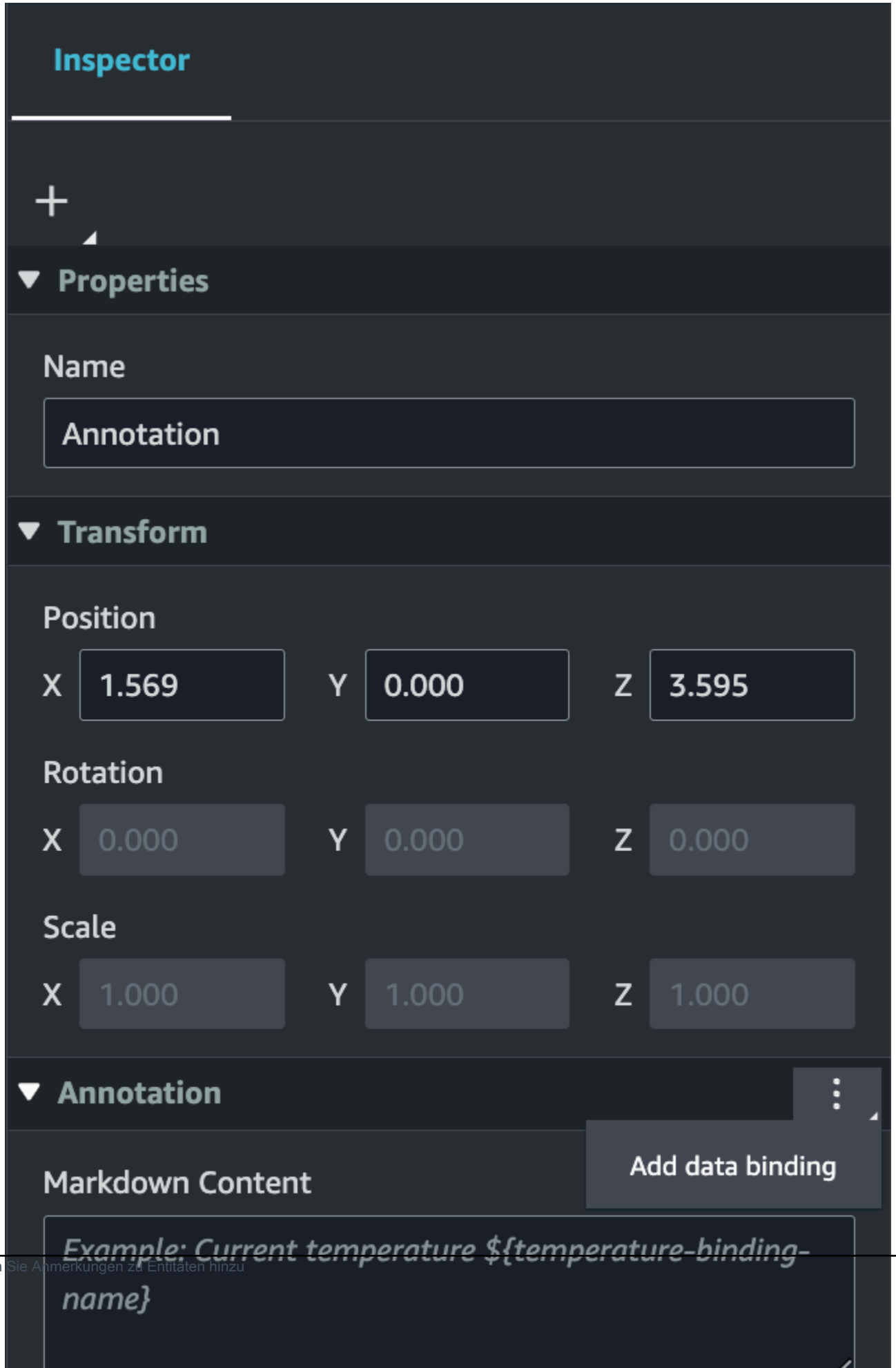


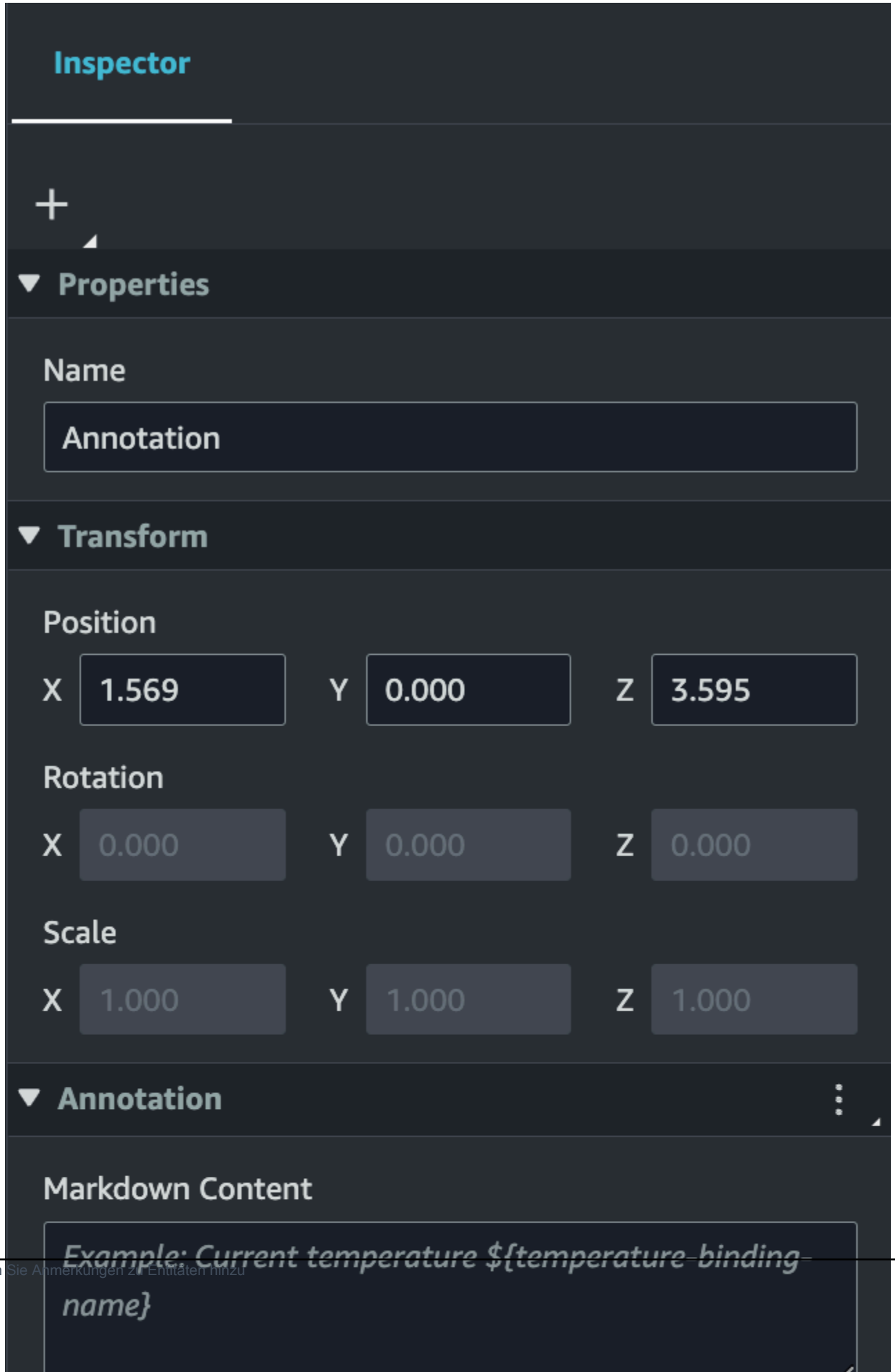
4. Scrollen Sie im Inspector-Fenster auf der linken Seite nach unten zum Bereich Anmerkungen. Schreiben Sie mithilfe der Markdown-Syntax den Text, den Ihre Anmerkung anzeigen soll.

   Weitere Informationen zum Schreiben in Markdown finden Sie in der offiziellen Dokumentation zur Markdown-Syntax, Basic Syntax.



5. Um Ihre AWS IoT TwinMaker Szenendaten an eine Anmerkung zu binden, wählen Sie Datenbindung hinzufügen, fügen Sie die Entitäts-ID hinzu und wählen Sie dann den Komponentennamen und den Eigenschaftsnamen der Entität aus, von der Sie Daten anzeigen möchten. Sie können den Bindungsnamen aktualisieren, um ihn als Markdown-Variable zu verwenden und die Daten in der Anmerkung anzuzeigen.

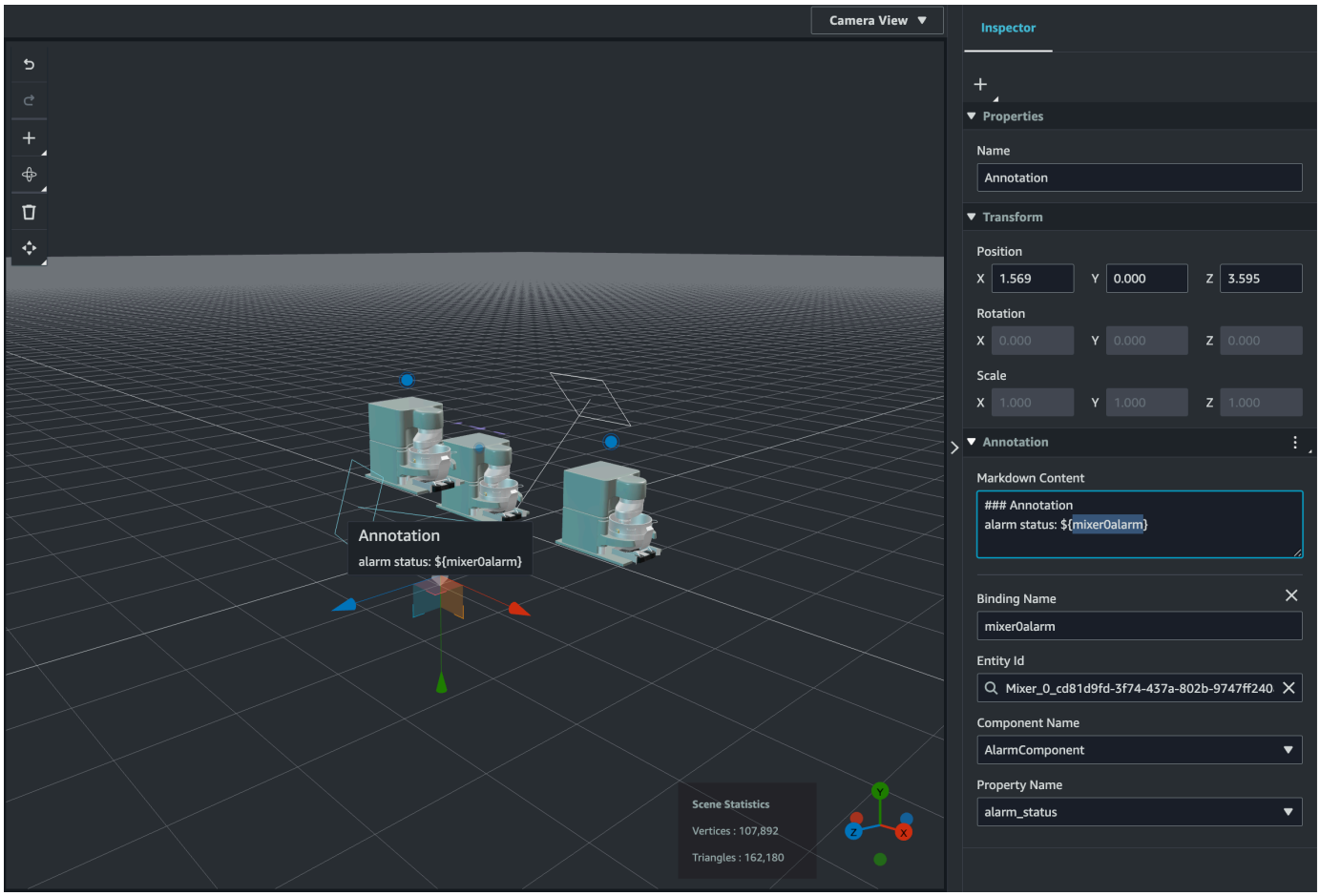

Inspector
Properties
Name
Annotation
Transform
Position
X 1.569
Y 0.000
Z 3.595
Rotation
X 0.000
Y 0.000
Z 0.000
Scale
X 1.000
Y 1.000
Z 1.000
Annotation
Add data binding
Markdown Content
Example: Current temperature ${temperature-binding-name}


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. Der Bindungsname wird verwendet, um die Variable der Anmerkung darzustellen.

   Geben Sie einen Bindungsnamen ein, um den letzten historischen Wert einer Entitätszeitreihe in der Variablensyntax der Anmerkung anzuzeigen: AWS IoT TwinMaker${*variable-name*}

   Als Beispiel zeigt dieses Overlay den Wert von`mixer0alarm`, in der Anmerkung mit der Syntax an. `${mixer0alarm}`



Fügen Sie Überlagerungen zu Tags hinzu

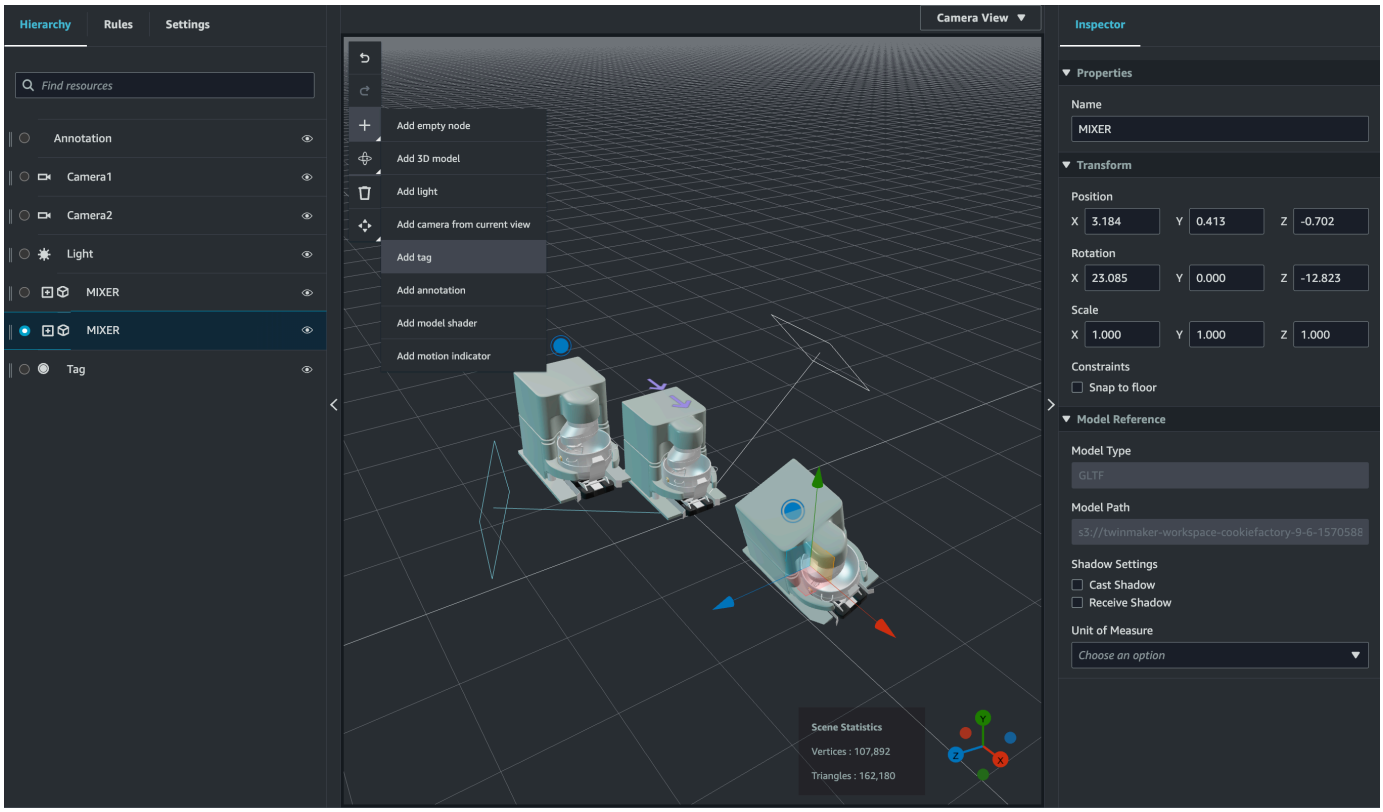
Sie können Überlagerungen für Ihre AWS IoT TwinMaker Szenen erstellen. Szenenüberlagerungen sind mit Tags verknüpft und können verwendet werden, um wichtige Daten zu Ihren Szenenentitäten anzuzeigen. Das Overlay wurde in Markdown erstellt und gerendert.

Weitere Informationen zum Schreiben in Markdown finden Sie in der offiziellen Dokumentation zur Markdown-Syntax, Basic Syntax.

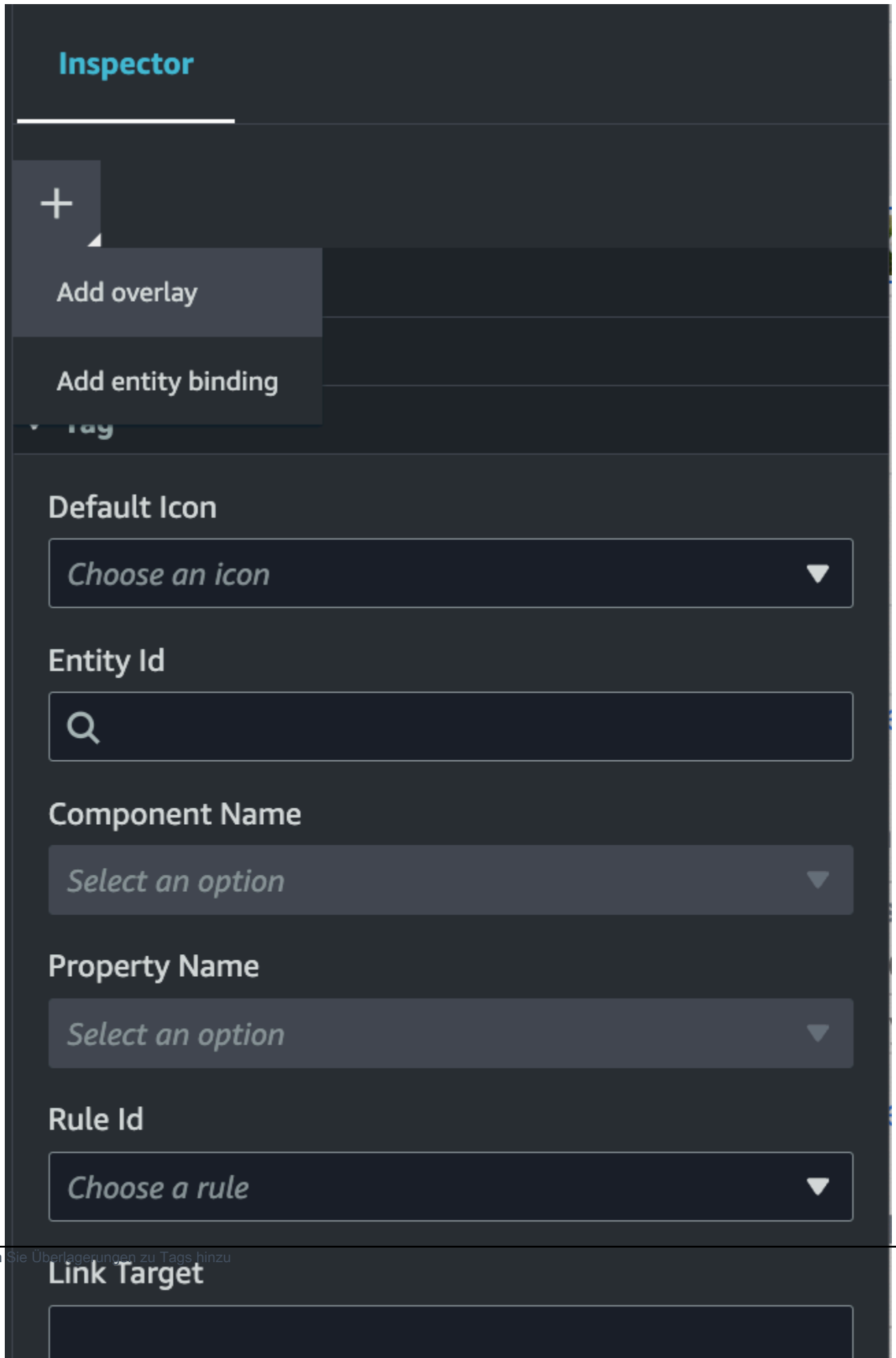
> **Note**
>
> Standardmäßig ist ein Overlay in einer Szene nur sichtbar, wenn das zugehörige Tag ausgewählt ist. Sie können dies in den Szeneneinstellungen so einstellen, dass alle Überlagerungen gleichzeitig sichtbar sind.

1. Navigiere in der Konsole zu deiner Szene.AWS IoT TwinMaker
2. Das AWS IoT TwinMaker Overlay ist mit einer Tag-Szene verknüpft. Sie können ein vorhandenes Tag aktualisieren oder ein neues hinzufügen.

   Drücken Sie die Plus-Taste und wählen Sie die Option Tag hinzufügen.



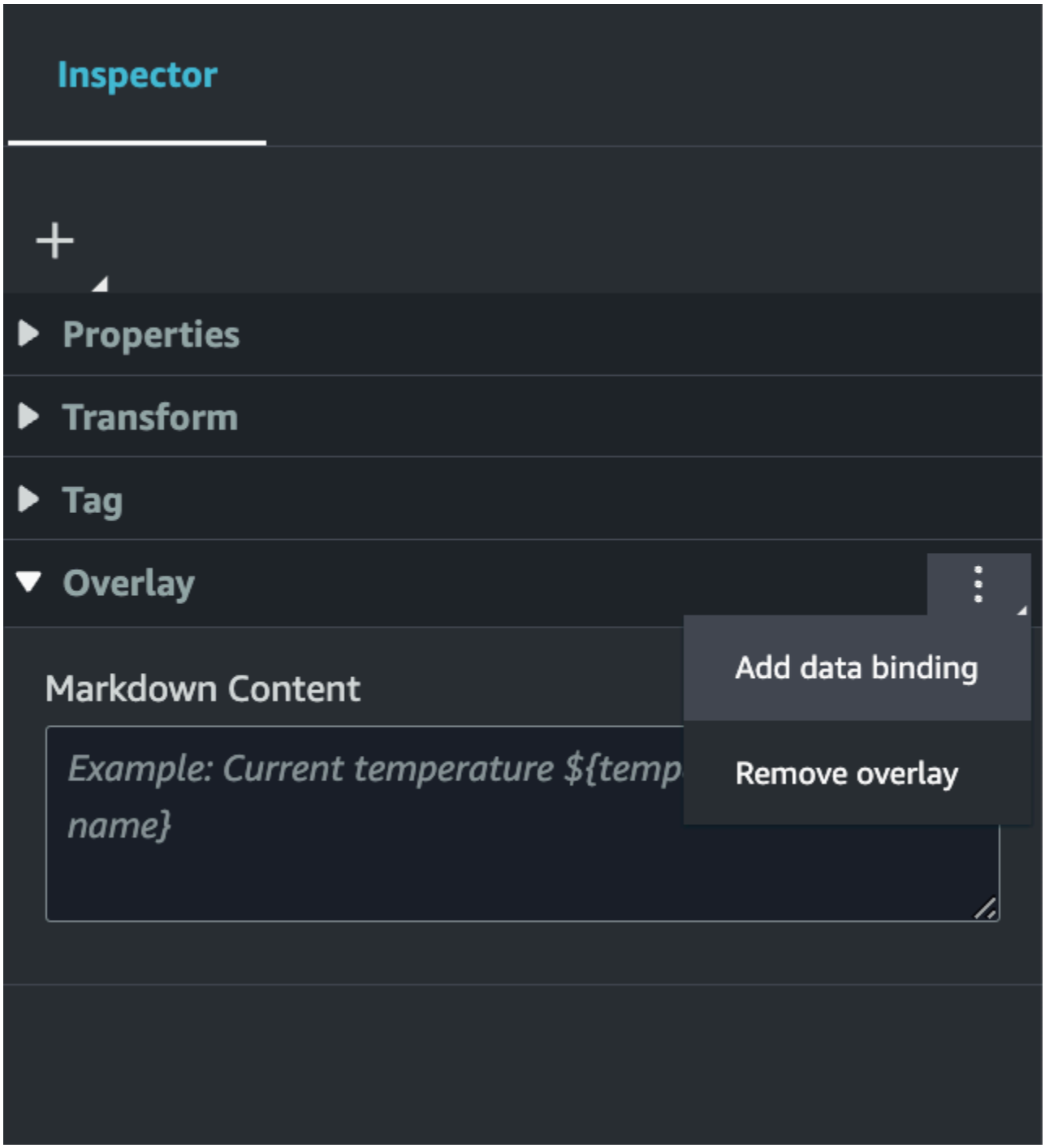
3. Wählen Sie im Inspektorfenster auf der rechten Seite die Schaltfläche + (Plussymbol) und dann Overlay hinzufügen.


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

4. Geben Sie in der Markdown-Syntax den Text ein, der in Ihrem Overlay angezeigt werden soll.

   Weitere Informationen zum Schreiben in Markdown finden Sie in der offiziellen Dokumentation zur Markdown-Syntax, Basic Syntax.

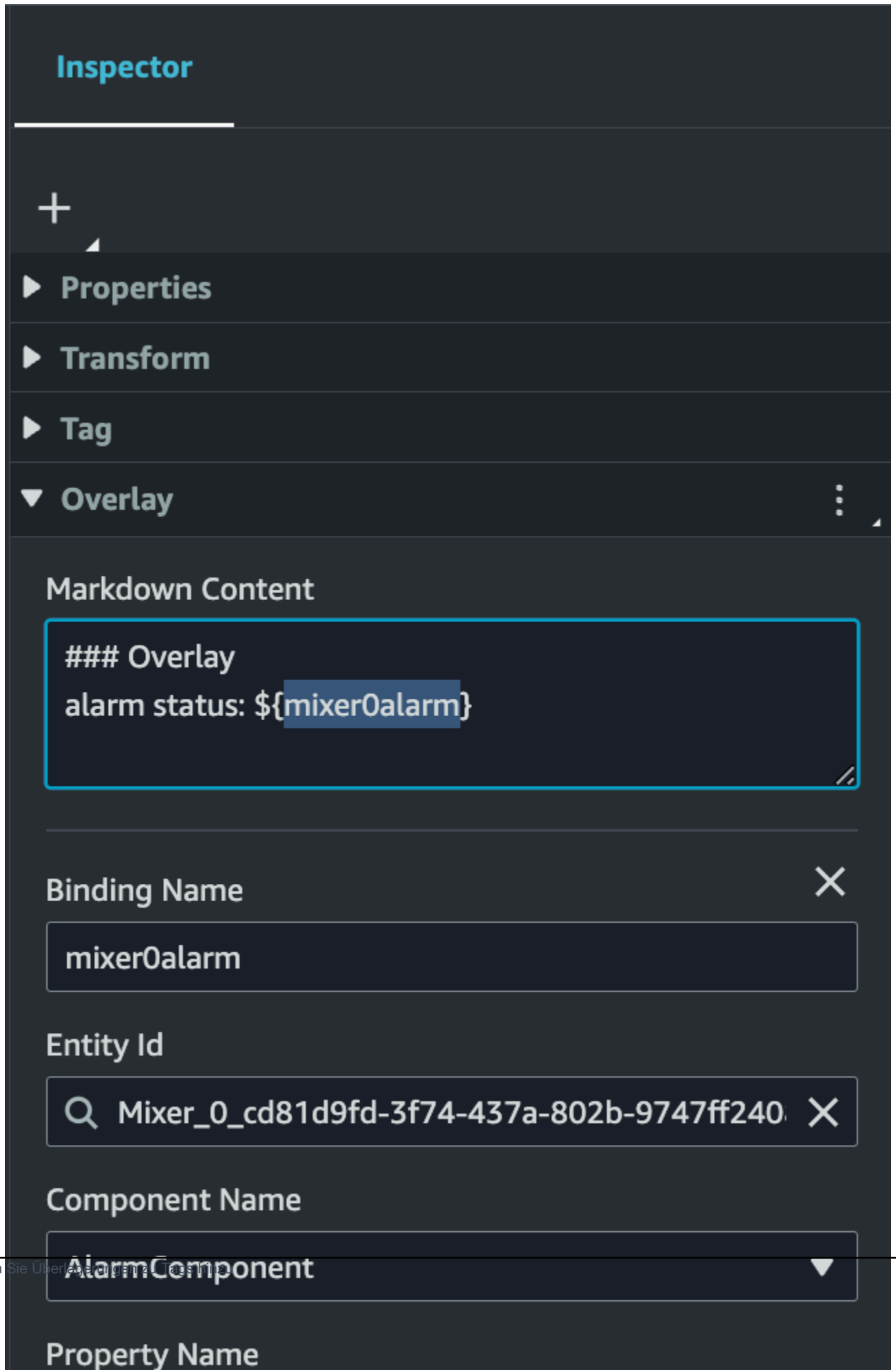
5. Um Ihre AWS IoT TwinMaker Szenendaten an ein Overlay zu binden, wählen Sie Datenbindung hinzufügen aus.



Fügen Sie den Bindungsnamen und die Entitäts-ID hinzu und wählen Sie dann den Komponentennamen und den Eigenschaftsnamen der Entität aus, von der Sie Daten anzeigen möchten.

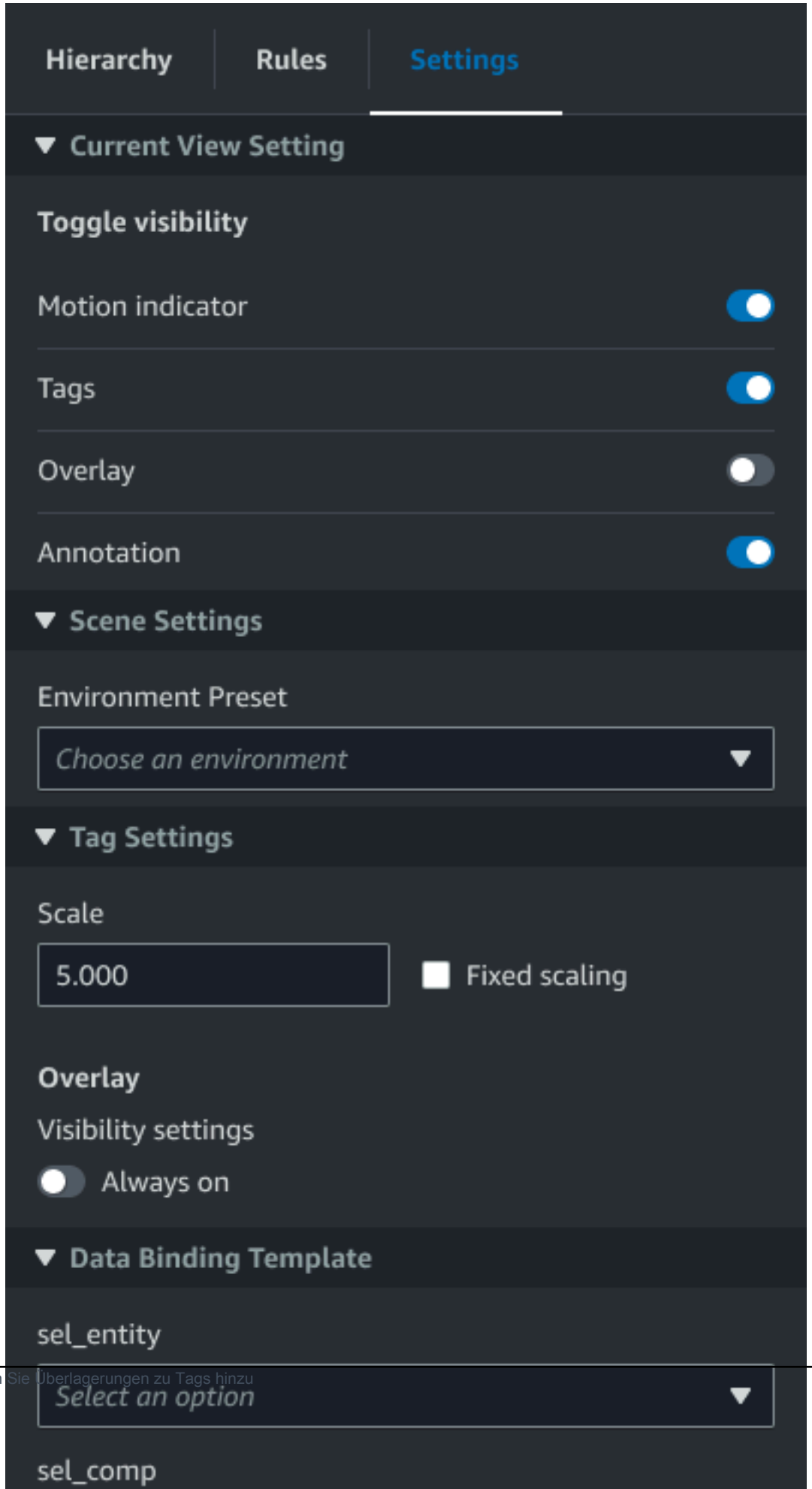
6. Sie können den letzten historischen Wert der Zeitreihendaten einer Entität in der Variablensyntax des AWS IoT TwinMaker Overlays anzeigen:. `${variable-name}`

   Als Beispiel zeigt dieses Overlay den Wert von`mixer0alarm`, in der Überlagerung mit der folgenden Syntax an. `${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. Um die Sichtbarkeit von Overlays zu aktivieren, öffnen Sie oben links den Tab Einstellungen und stellen Sie sicher, dass der Schalter für Overlay aktiviert ist, sodass alle Overlays gleichzeitig sichtbar sind.

> **Note**
>
> Standardmäßig ist ein Overlay in einer Szene nur sichtbar, wenn das zugehörige Tag ausgewählt ist.


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

Bearbeiten Sie Ihre Szenen

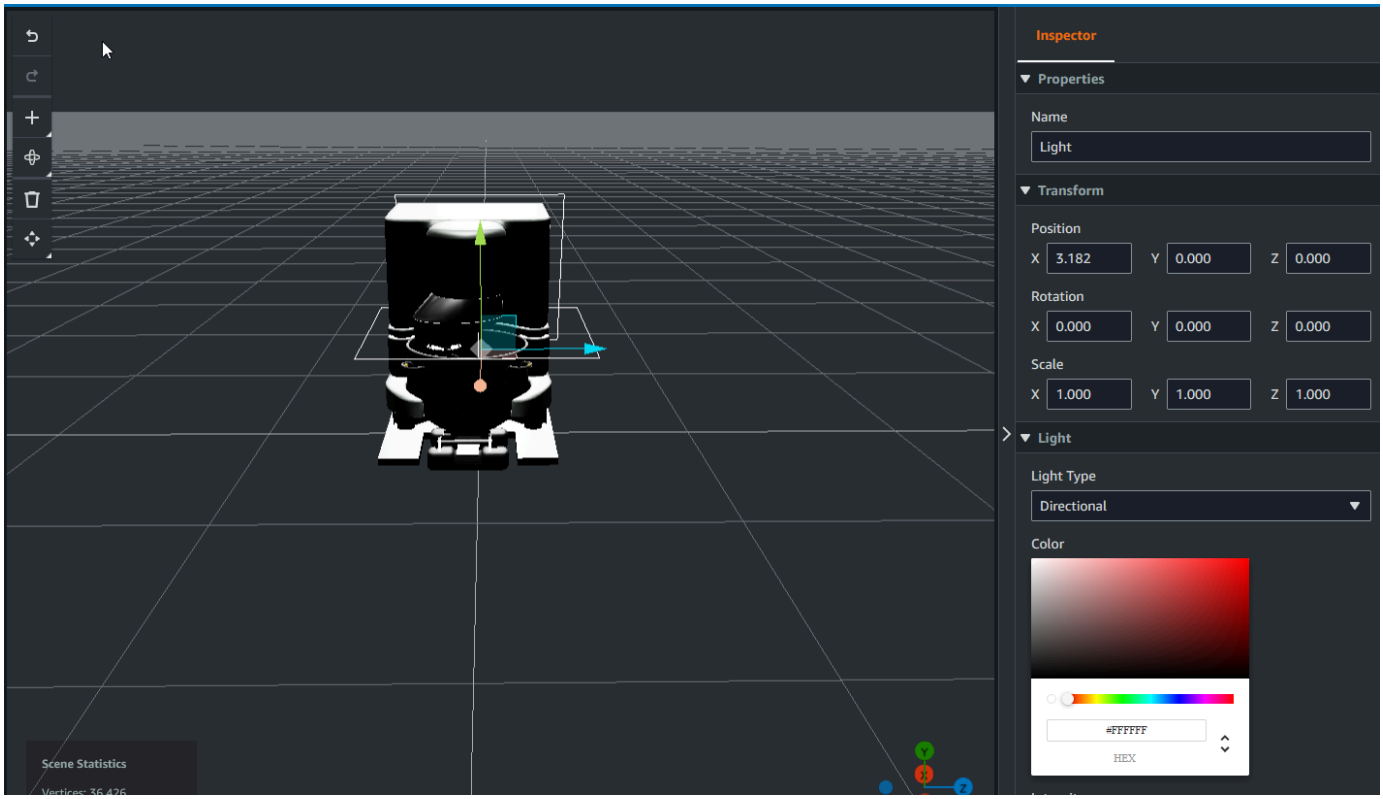
Nachdem Sie eine Szene erstellt haben, können Sie Ihrer Szene Entitäten und Komponenten hinzufügen und erweiterte Widgets konfigurieren. Verwenden Sie Entitätskomponenten und Widgets, um Ihren digitalen Zwilling zu modellieren und Funktionen bereitzustellen, die Ihrem Anwendungsfall entsprechen.

Fügen Sie Ihren Szenen Modelle hinzu

Gehen Sie wie folgt vor, um Ihrer Szene Modelle hinzuzufügen.

> **Note**
>
> Um Ihrer Szene Modelle hinzuzufügen, müssen Sie die Modelle zunächst in die AWS IoT TwinMaker Ressourcenbibliothek hochladen. Weitere Informationen finden Sie unter Laden Sie Ressourcen in die AWS IoT TwinMaker Ressourcenbibliothek hoch.

1. Wählen Sie auf der Scene Composer-Seite das Pluszeichen (+) und dann 3D-Modell hinzufügen aus.
2. Wählen Sie im Fenster „Ressource aus Ressourcenbibliothek hinzufügen“ die Datei „CookieFactorMixer.glb“ und dann „Hinzufügen“ aus. Scene Composer wird geöffnet.
3. Optional: Wählen Sie das Pluszeichen (+) und dann Licht hinzufügen.
4. Wählen Sie jede Lichtoption aus, um zu sehen, wie sie sich auf die Szene auswirken.



> **Note**
>
> Szenen verfügen über eine standardmäßige Umgebungsbeleuchtung. Um einen Verlust der Bildrate zu vermeiden, sollten Sie erwägen, die Anzahl der zusätzlichen Lichter in Ihrer Szene zu begrenzen.

Fügen Sie Ihrer Szene Widgets für die erweiterte Benutzeroberfläche von Model Shader hinzu

Modell-Shader-Widgets können die Farbe eines Objekts unter von Ihnen definierten Bedingungen ändern. Sie können beispielsweise ein Farb-Widget erstellen, das die Farbe eines Cookie-Mixers in Ihrer Szene auf der Grundlage der Temperaturdaten des Mixers ändert.

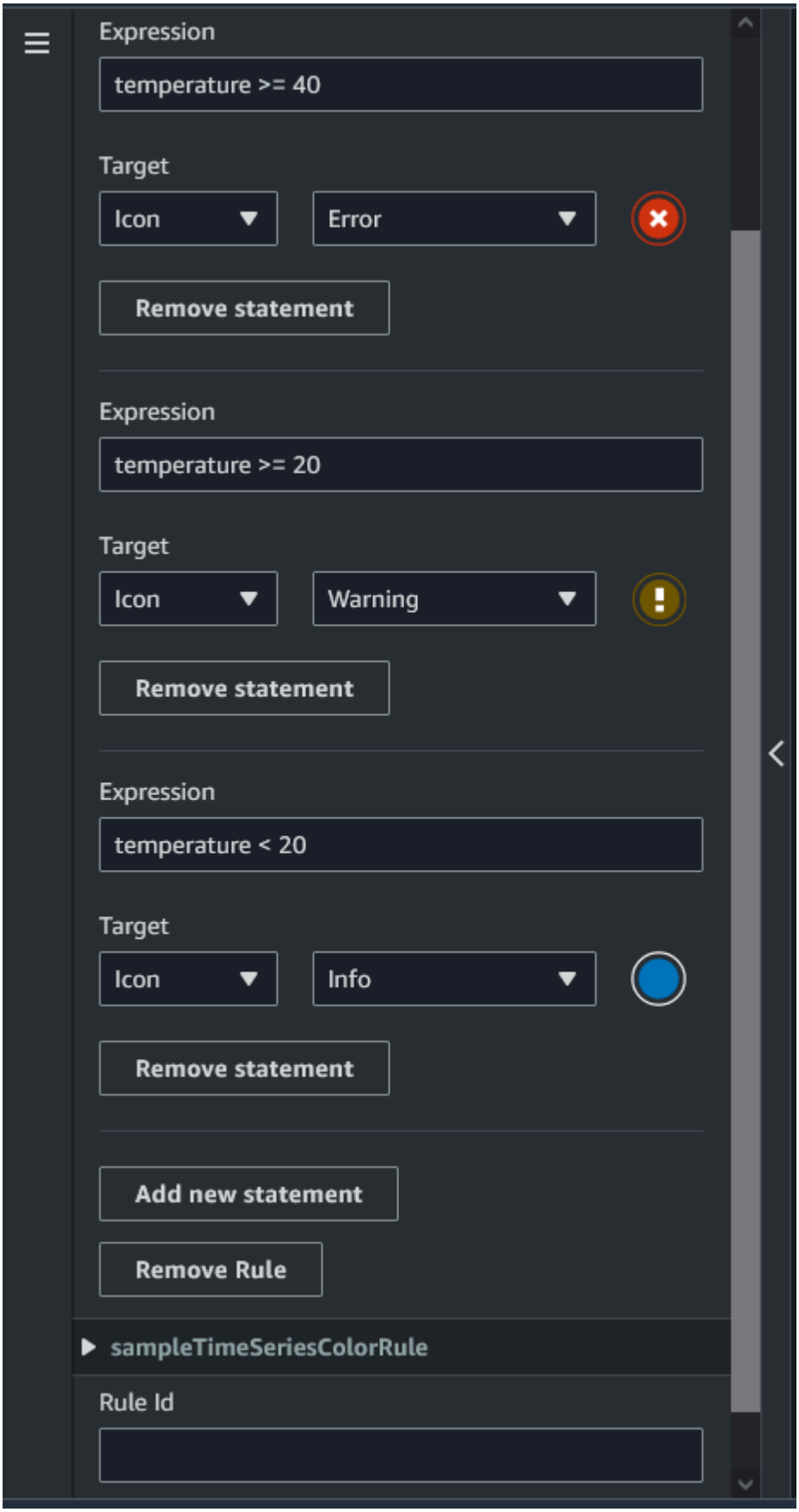
Gehen Sie wie folgt vor, um einem ausgewählten Objekt Modell-Shader-Widgets hinzuzufügen.

1. Wählen Sie ein Objekt in der Hierarchie aus, dem Sie ein Widget hinzufügen möchten. Drücken Sie die Taste + und wählen Sie dann Model Shader.

2. Um eine neue visuelle Regelgruppe hinzuzufügen, folgen Sie zunächst den nachstehenden Anweisungen, um die zu erstellen ColorRule, und wählen Sie dann im Inspektorfenster für das Objekt mit der Regel-ID ColorRule.
3. Wählen Sie die EntityID,, aus ComponentName, an die PropertyName Sie den Modell-Shader binden möchten.

Erstellen Sie visuelle Regeln für Ihre Szenen

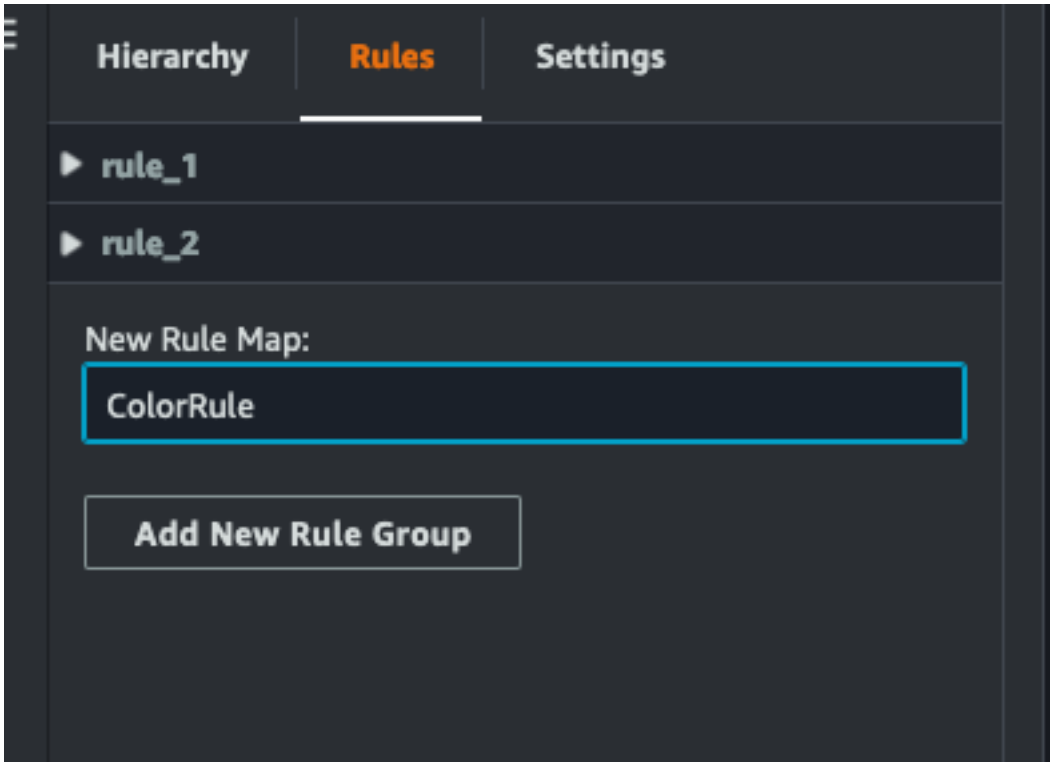
Mithilfe von visuellen Regelkarten können Sie die datengesteuerten Bedingungen angeben, die das visuelle Erscheinungsbild eines Widgets mit erweiterter Benutzeroberfläche, z. B. eines Tags oder eines Modellshaders, verändern. Es stehen Beispielregeln zur Verfügung, aber Sie können auch Ihre eigenen Regeln erstellen. Das folgende Beispiel zeigt eine visuelle Regel.


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

Das Bild oben zeigt eine Regel, nach der eine zuvor definierte Dateneigenschaft mit der ID „Temperatur“ mit einem bestimmten Wert verglichen wird. Wenn die „Temperatur“ beispielsweise größer oder gleich 40 ist, ändert der Status das Aussehen des Tags in einen roten Kreis. Wenn das Ziel im Grafana-Dashboard ausgewählt wird, füllt es einen Detailbereich, der für die Verwendung derselben Datenquelle konfiguriert ist.

Das folgende Verfahren zeigt Ihnen, wie Sie eine neue visuelle Regelgruppe für die erweiterte Benutzeroberflächenebene zur Mesh-Färbung hinzufügen.

1. Geben Sie auf der Registerkarte Regeln in der Konsole einen Namen ein, z. B. ColorRule in das Textfeld, und wählen Sie Neue Regelgruppe hinzufügen aus.



2. Definieren Sie eine neue Regel für Ihren Anwendungsfall. Sie können beispielsweise eine anhand der Dateneigenschaft „Temperatur“ erstellen, bei der der gemeldete Wert unter 20 liegt. Verwenden Sie die folgende Syntax für Regelausdrücke: Weniger als ist **<**, größer als ist **>**, kleiner als oder gleich ist **<=**, größer als oder gleich ist **>=** und gleich ist **==**. (Weitere Informationen finden Sie in der Apache Commons JEXL-Syntax.)
3. Setzt das Ziel auf eine Farbe. Verwenden Sie beispielsweise Hex-Werte, um eine Farbe zu definieren. `#fcba03` (Weitere Informationen zu Hexadezimalwerten finden Sie unter Hexadezimal.)

Erstellen Sie Tags für Ihre Szenen

Ein Tag ist eine Anmerkung, die zu einer bestimmten `x,y,z` Koordinatenposition einer Szene hinzugefügt wird. Das Tag verwendet eine Entitätseigenschaft, um einen Szenenteil mit dem Knowledge Graph zu verbinden. Sie können ein Tag verwenden, um das Verhalten oder das visuelle Erscheinungsbild eines Elements in der Szene zu konfigurieren, z. B. eines Alarms.

> **Note**
>
> Um Tags Funktionen hinzuzufügen, wenden Sie visuelle Regeln auf sie an.

Gehen Sie wie folgt vor, um Ihrer Szene Tags hinzuzufügen.

1. Wählen Sie ein Objekt in der Hierarchie aus, klicken Sie auf die Schaltfläche „+“ und dann auf „Tag hinzufügen“.
2. Benennen Sie das Tag. Um dann eine visuelle Regel anzuwenden, wählen Sie eine visuelle Gruppen-ID aus.
3. Wählen Sie in den Dropdownlisten die EntityID, und ComponentName aus. PropertyName
4. Um das Feld Datenpfad aufzufüllen, wählen Sie Create aus. DataFrameLabel

Modellformat für 3D-Kacheln

Verwenden von 3D-Kacheln in Ihrer Szene

Wenn beim Laden von 3D-Szenen lange Wartezeiten auftreten AWS IoT TwinMaker oder die Renderleistung beim Navigieren in einem komplexen 3D-Modell schlecht ist, sollten Sie Ihre Modelle in 3D-Kacheln konvertieren. In diesem Abschnitt werden das 3D-Kachelformat und die verfügbaren Tools von Drittanbietern beschrieben. Lesen Sie weiter, um zu entscheiden, ob 3D-Kacheln für Ihren Anwendungsfall geeignet sind, und um Hilfe bei den ersten Schritten zu erhalten.

Anwendungsfall für komplexe Modelle

Ein 3D-Modell in Ihrer AWS IoT TwinMaker Szene kann zu Leistungsproblemen wie langsamen Ladezeiten und verzögerter Navigation führen, wenn das Modell:

- Groß: Die Dateigröße ist größer als 100 MB.

- Dicht: Es besteht aus Hunderten oder Tausenden von unterschiedlichen Maschen.
- Komplex: Die Netzgeometrie besteht aus Millionen von Dreiecken, die komplexe Formen bilden.

Format für 3D-Kacheln

Das Format 3D Tiles ist eine Lösung zum Streamen von Modellgeometrie und zur Verbesserung der 3D-Renderleistung. Es ermöglicht das sofortige Laden von 3D-Modellen in einer AWS IoT TwinMaker Szene und optimiert 3D-Interaktionen, indem Teile eines Modells auf der Grundlage dessen geladen werden, was in der Kameraansicht sichtbar ist.

Das Format 3D Tiles wurde von Cesium erstellt. Cesium verfügt über einen verwalteten Dienst zur Konvertierung von 3D-Modellen in 3D-Kacheln namens Cesium Ion. Dies ist derzeit die beste Lösung für die Erstellung von 3D-Kacheln. Wir empfehlen dies für Ihre komplexen Modelle in den unterstützten Formaten. Sie können Cesium registrieren und auf der Preisseite von Cesium das passende Abonnement auswählen, das Ihren Geschäftsanforderungen entspricht.

Um ein 3D-Tiles-Modell vorzubereiten, das Sie einer AWS IoT TwinMaker Szene hinzufügen können, folgen Sie den von Cesium Ion dokumentierten Anweisungen:

- Importieren Sie ein Modell in Cesium Ion

Laden Sie Caesium 3D-Kacheln hoch auf AWS

Sobald Ihr Modell in 3D-Kacheln konvertiert wurde, laden Sie die Modelldateien herunter und laden Sie sie dann in Ihren AWS IoT TwinMaker Workspace-Amazon S3-Bucket hoch:

1. Erstellen Sie Ihr 3D Tiles-Modellarchiv und laden Sie es herunter.
2. Entpacken Sie das Archiv in einen Ordner.
3. Laden Sie den gesamten Ordner 3D Tiles in den Amazon S3 S3-Bucket hoch, der Ihrem AWS IoT TwinMaker Workspace zugeordnet ist. (Siehe Hochladen von Objekten im Amazon S3 S3-Benutzerhandbuch.)
4. Wenn Ihr 3D Tiles-Modell erfolgreich hochgeladen wurde, sehen Sie in Ihrer AWS IoT TwinMaker Ressourcenbibliothek einen Amazon S3 S3-Ordnerpfad mit Typ `Tiles3D`.

> **Note**
>
> Die AWS IoT TwinMaker Ressourcenbibliothek unterstützt das direkte Hochladen von 3D Tiles-Modellen nicht.

Verwenden von 3D-Kacheln in AWS IoT TwinMaker

AWS IoT TwinMaker kennt jedes 3D Tiles-Modell, das in Ihren Workspace S3-Bucket hochgeladen wurde. Für das Modell müssen a `tileset.json` und alle abhängigen Dateien (.gltf, .b3dm, .i3dm, .cmpt, .pnts) im selben Amazon S3 S3-Verzeichnis verfügbar sein. Der Amazon S3 S3-Verzeichnispfad wird in der Ressourcenbibliothek mit dem Typ angezeigt`Tiles3D`.

Gehen Sie wie folgt vor, um das Modell 3D Tiles zu Ihrer Szene hinzuzufügen:

1. Wählen Sie auf der Scene Composer-Seite das Pluszeichen (+) und dann 3D-Modell hinzufügen aus.
2. Wählen Sie im Fenster „Ressource aus Ressourcenbibliothek hinzufügen“ den Pfad zu Ihrem 3D-Kachelmodell mit dem Typ `Tiles3D` aus und klicken Sie dann auf Hinzufügen.
3. Klicken Sie auf die Leinwand, um das Modell in Ihrer Szene zu platzieren.

Unterschiede zwischen 3D-Fliesen

3D-Kacheln unterstützt derzeit keine geometrischen und semantischen Metadaten, was bedeutet, dass die Netzhierarchie des Originalmodells für die Funktion zur Auswahl von Untermodellen nicht verfügbar ist. Sie können Ihrem 3D-Kachelmodell immer noch Widgets hinzufügen, aber Sie können keine Funktionen verwenden, die speziell auf Untermodelle abgestimmt sind: Modell-Shader, separate 3D-Transformationen oder Entitätsbindung für ein Submodell-Mesh.

Es wird empfohlen, die Konvertierung von 3D-Kacheln für große Objekte zu verwenden, die als Kontext für den Hintergrund einer Szene dienen. Wenn Sie möchten, dass ein Untermodell weiter aufgeschlüsselt und mit Anmerkungen versehen wird, sollte es als separates glTF/glb Asset extrahiert und direkt der Szene hinzugefügt werden. Dies kann mit kostenlosen und gängigen 3D-Tools wie Blender geschehen.

Beispiel für einen Anwendungsfall:

- Sie haben ein 1-GB-Modell einer Fabrik mit detaillierten Maschinenräumen und Fußböden, Schaltkästen und Sanitärrohren. Die Schaltkästen und Leitungen müssen rot leuchten, wenn die zugehörigen Eigenschaftsdaten einen Schwellenwert überschreiten.
- Sie isolieren die Box- und Rohrnetze im Modell und exportieren sie mit Blender in ein separates glTF.
- Sie konvertieren die Fabrik ohne Elektro- und Sanitärelemente in ein 3D-Kachelmodell und laden es auf S3 hoch.
- Sie fügen einer AWS IoT TwinMaker Szene am Ursprung (0,0,0) sowohl das 3D-Kachelmodell als auch das glTF-Modell hinzu.
- Sie fügen Modellshader-Komponenten zu den Schaltkasten- und Leitungsuntermodellen der glTF hinzu, um die Netze auf der Grundlage von Eigenschaftsregeln rot zu färben.

Dynamische Szenen

AWS IoT TwinMaker Szenen nutzen das Potenzial des Wissensgraphen, indem sie Szenenknoten und Einstellungen in einer Entitätskomponente speichern. Verwenden Sie die AWS IoT TwinMaker Konsole, um dynamische Szenen zu erstellen, um 3D-Szenen einfacher zu verwalten, zu erstellen und zu rendern.

Die wichtigsten Funktionen:

- Alle Objekte, Einstellungen und Datenbindungen von 3D-Szenenknoten werden auf der Grundlage von Knowledge-Graph-Abfragen „dynamisch“ gerendert.
- Wenn Sie den schreibgeschützten Scene Viewer in einer Grafana- oder benutzerdefinierten Anwendung verwenden, können Sie Aktualisierungen Ihrer Szenen in Intervallen von 30 Sekunden abrufen.

Statische und dynamische Szenen

Statische Szenen bestehen aus einer in S3 gespeicherten JSON-Szenendatei, die Details zu allen Szenenknoten und Einstellungen enthält. Jede Änderung an der Szene muss am JSON-Dokument vorgenommen und in S3 gespeichert werden. Eine statische Szene ist die einzige Option, wenn Sie einen Basispreisplan haben.

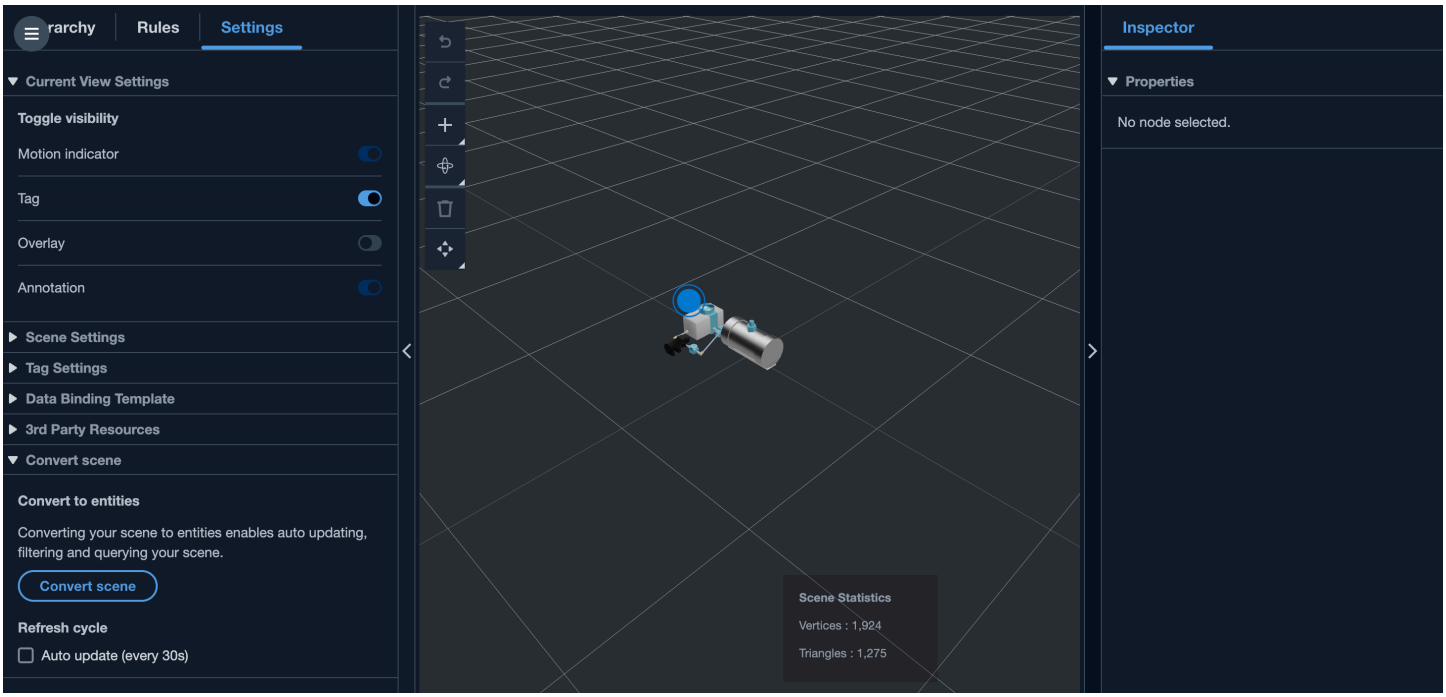
Dynamische Szenen bestehen aus einer JSON-Szenendatei mit globalen Einstellungen für die Szene, während alle anderen Szenenknoten und Knoteneinstellungen als Entitätskomponenten

im Knowledge Graph gespeichert werden. Dynamische Szenen werden nur in Standard- und gestaffelten Paket-Preisplänen unterstützt. Informationen AWS IoT TwinMaker Preismodus wechseln zum Upgrade Ihres Preisplans finden Sie unter).

Gehen Sie wie folgt vor, um eine vorhandene statische Szene in eine dynamische Szene zu konvertieren:

- Navigieren Sie in der AWS IoT TwinMaker Konsole zu Ihrer Szene.
- Klicken Sie im linken Bereich auf die Registerkarte Einstellungen.
- Erweitern Sie den Bereich Szene konvertieren am unteren Rand des Bedienfelds.
- Klicken Sie auf die Schaltfläche „Szene konvertieren“ und anschließend auf „Bestätigen“.

> **Warning**
>
> Die Konvertierung von einer statischen in eine dynamische Szene ist irreversibel.



Typen und Entitäten von Szenenkomponenten

Um szenenspezifische Entitätskomponenten zu erstellen, werden die folgenden 1P-Komponententypen unterstützt:

- com.amazon.iottwinmaker.3d.component.camera Ein Komponententyp, der die Einstellungen eines Kamera-Widgets speichert.
- com.amazon.iottwinmaker.3d.component.dataoverlay Ein Komponententyp, der die Einstellungen für ein Overlay eines Annotations- oder Tag-Widgets speichert.
- com.amazon.iottwinmaker.3d.component.light Ein Komponententyp, der die Einstellungen eines Light-Widgets speichert.
- com.amazon.iottwinmaker.3d.component.modelref Ein Komponententyp, der die Einstellungen und die S3-Position eines in einer Szene verwendeten 3D-Modells speichert.
- com.amazon.iottwinmaker.3d.component.modelshader Ein Komponententyp, der die Einstellungen eines Modellshaders in einem 3D-Modell speichert.
- com.amazon.iottwinmaker.3d.component.motionindicator Ein Komponententyp, der die Einstellungen eines Bewegungsanzeiger-Widgets speichert.
- com.amazon.iottwinmaker.3d.component.submodelref Ein Komponententyp, der die Einstellungen eines Untermodells eines 3D-Modells speichert.
- com.amazon.iottwinmaker.3d.component.tag Ein Komponententyp, der die Einstellungen eines Tag-Widgets speichert.
- com.amazon.iottwinmaker.3d.node Ein Komponententyp, der die grundlegenden Einstellungen eines Szenenknotens wie seine 3D-Transformation, seinen Namen und seine generischen Eigenschaften speichert.

Dynamische Szenenkonzepte

Dynamische Szenenentitäten werden unter einer globalen Entität mit der Bezeichnung `$SCENES` gespeichert. Jede Szene besteht aus einer Stammeltität und einer Hierarchie von untergeordneten Entitäten, die der Szenenknotenhierarchie entsprechen. Jeder Szenenknoten unter dem Stamm hat eine com.amazon.iottwinmaker.3d.node-Komponente und eine Komponente für den Knotentyp (3D-Modell, Widget usw.).

> **Warning**
>
> Löschen Sie keine Szenen-Entitäten manuell, da sich Ihre Szene sonst möglicherweise in einem defekten Zustand befindet. Wenn Sie eine Szene teilweise oder vollständig löschen möchten, verwenden Sie die Scene Composer-Seite, um Szenenknoten hinzuzufügen und zu löschen, und verwenden Sie die Szenenseite, um eine Szene auszuwählen und zu löschen.

Erstellen Sie eine benutzerdefinierte Webanwendung mithilfe von AWS IoT TwinMaker UI-Komponenten

AWS IoT TwinMaker bietet Open-Source-UI-Komponenten für AWS IoT Anwendungsentwickler. Mithilfe dieser UI-Komponenten können Entwickler maßgeschneiderte Webanwendungen mit aktivierten AWS IoT TwinMaker Funktionen für ihre digitalen Zwillinge erstellen.

AWS IoT TwinMaker UI-Komponenten sind Teil des AWS IoT Application Kit, einer clientseitigen Open-Source-Bibliothek, mit der Entwickler von IoT-Anwendungen die Entwicklung komplexer IoT-Anwendungen vereinfachen können

AWS IoT TwinMaker Zu den UI-Komponenten gehören:

- AWS IoT TwinMaker Quelle:

  Eine Datenverbindungskomponente, mit der Sie Daten abrufen und mit Ihren AWS IoT TwinMaker Daten und digitalen Zwillingen interagieren können.

  Weitere Informationen finden Sie in der AWS IoT TwinMaker Quelldokumentation.
- Szenenbetrachter:

  Eine darauf aufgebaute 3D-Rendering-Komponente`@react-three/fiber`, die Ihren digitalen Zwilling rendert und Ihnen die Interaktion mit ihm ermöglicht.

  Weitere Informationen finden Sie in der Scene Viewer-Dokumentation.
- Videoplayer:

  Eine Video-Player-Komponente, mit der Sie ein Video aus den Kinesis Video Streams streamen können. AWS IoT TwinMaker

  Weitere Informationen finden Sie in der Video Player-Dokumentation.

Weitere Informationen zur Verwendung von AWS IoT Application Kit finden Sie auf der Github-Seite von AWS IoT Application Kit.

Anweisungen zum Starten einer neuen Webanwendung mit dem AWS IoT Application Kit finden Sie auf der offiziellen IoT App Kit-Dokumentationsseite.

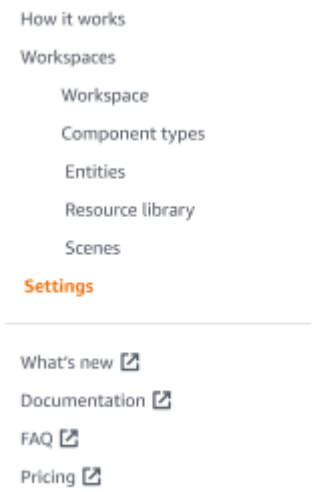
AWS IoT TwinMaker Preismodus wechseln

AWS IoT TwinMaker verfügt derzeit über drei Preismodi: Basic, Standard oder gestaffeltes Paket. Der Standard-Preismodus ist als Standardpreismodus festgelegt.

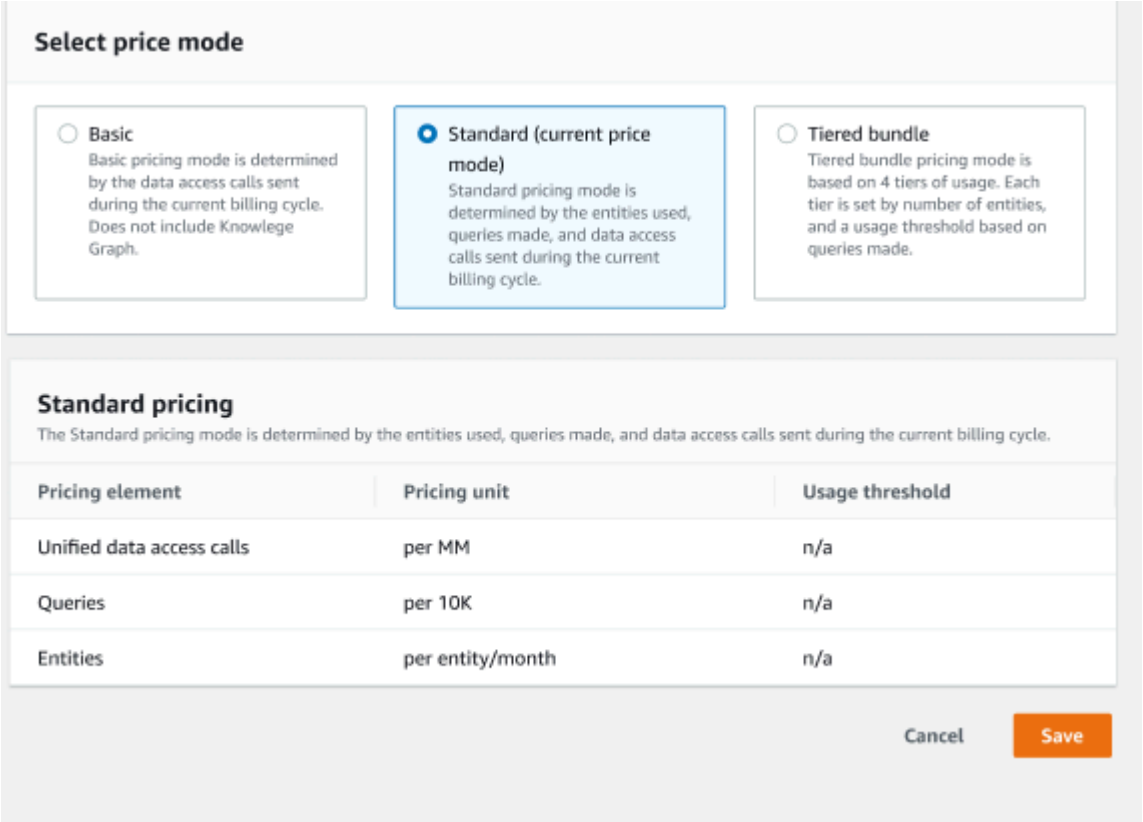
Sie können jederzeit vom nutzungsbasierten zum gestaffelten Preismodell wechseln, die Änderung wird jedoch zu Beginn Ihres nächsten Abrechnungszeitraums wirksam. Sobald Sie vom nutzungsbasierten zum gestaffelten Preismodell gewechselt sind, können Sie für die nächsten drei Nutzungszyklen nicht mehr zum nutzungsbasierten Preismodell zurückkehren. Wenn Sie vom Basismodell zum Standardmodell wechseln, wird die Änderung sofort wirksam. Einzelheiten und Kosteninformationen finden Sie unter AWS IoT TwinMaker Preise

Dieses Verfahren zeigt Ihnen, wie Sie Ihren Preismodus in der AWS IoT TwinMaker Konsole ändern können:

1. Öffnen Sie die AWS IoT TwinMaker -Konsole.
2. Wählen Sie im linken Navigationsbereich Einstellungen aus. Die Seite mit den Preisen wird geöffnet.



3. Wählen Sie den Modus „Preis ändern“.
4. Wählen Sie entweder den Standard - oder den gestaffelten Paketmodus aus, wie im folgenden Screenshot gezeigt.



5. Wählen Sie Speichern, um Ihren neuen Preismodus zu bestätigen.
6. Sie haben jetzt Ihren Preismodus geändert.

> **Note**
>
> Sie können jederzeit vom nutzungsbasierten zum gestaffelten Preismodell wechseln, die Änderung wird jedoch zu Beginn Ihres nächsten Abrechnungszeitraums wirksam. Sobald Sie vom nutzungsbasierten zum gestaffelten Preismodell gewechselt sind, können Sie für die nächsten drei Nutzungszyklen nicht mehr zum nutzungsbasierten Preismodell zurückkehren. Wenn Sie vom Basismodell zum Standardmodell wechseln, wird die Änderung sofort wirksam.

AWS IoT TwinMaker Wissensgraph

Der AWS IoT TwinMaker Wissensgraph organisiert alle in Ihren AWS IoT TwinMaker Arbeitsbereichen enthaltenen Informationen und präsentiert sie in einem visuellen Grafikformat. Sie können Abfragen für Ihre Entitäten, Komponenten und Komponententypen ausführen, um visuelle Diagramme zu erstellen, die Ihnen die Beziehungen zwischen Ihren AWS IoT TwinMaker Ressourcen zeigen.

In den folgenden Themen erfahren Sie, wie Sie den Knowledge Graph verwenden und integrieren.

Themen

- AWS IoT TwinMaker Kernkonzepte des Wissensgraphen
- Wie führe ich AWS IoT TwinMaker Knowledge-Graph-Abfragen aus
- Integration von Knowledge Graph-Szenen
- Wie benutzt man AWS IoT TwinMaker Knowledge Graph mit Grafana
- AWS IoT TwinMaker Knowledge Graph, zusätzliche Ressourcen

AWS IoT TwinMaker Kernkonzepte des Wissensgraphen

Dieses Thema behandelt die wichtigsten Konzepte und das Vokabular der Knowledge-Graph-Funktion.

So funktioniert der Wissensgraph:

Der Wissensgraph stellt Beziehungen zwischen Entitäten und ihren Komponenten zum vorhandenen CreateEntityoder her UpdateEntity APIs. Eine Beziehung ist nur eine Eigenschaft eines speziellen Datentyps BEZIEHUNG, die für eine Komponente einer Entität definiert ist. AWS IoT TwinMaker Knowledge Graph ruft die ExecuteQueryAPI auf, um eine Abfrage auf der Grundlage von Daten in den Entitäten oder den Beziehungen zwischen ihnen durchzuführen. Knowledge Graph verwendet die flexible PartiQL-Abfragesprache (die von vielen AWS Diensten verwendet wird), die neu hinzugefügte Graph-Match-Syntaxunterstützung bietet, um Ihnen beim Schreiben Ihrer Abfragen zu helfen. Nachdem die Aufrufe getätigt wurden, können Sie die Ergebnisse als Tabelle anzeigen oder sie als Diagramm verbundener Knoten und Kanten visualisieren.

Wichtige Begriffe im Knowledge Graph:

- Entitätsgraph: Eine Sammlung von Knoten und Kanten innerhalb eines Workspace.

- Knoten: Jede Entität in Ihrem Workspace wird zu einem Knoten im Entitätsdiagramm.
- Kante: Jede Beziehungseigenschaft, die für eine Komponente einer Entität definiert ist, wird zu einer Kante im Entitätsgraphen. Darüber hinaus wird eine hierarchische Eltern-Kind-Beziehung, die mithilfe des parentEntityId Felds einer Entität definiert wurde, auch zu einer Kante im Entitätsdiagramm mit einem Beziehungsnamen "isChildOf". Alle Kanten sind Richtungskanten.
- Beziehung: Eine AWS IoT TwinMaker Beziehung ist ein besonderer Eigenschaftstyp einer Entitätskomponente. Sie können die UpdateEntityAPI AWS IoT TwinMaker CreateEntityoder verwenden, um eine Beziehung zu definieren und zu bearbeiten. AWS IoT TwinMaker In muss eine Beziehung in einer Komponente einer Entität definiert werden. Eine Beziehung kann nicht als isolierte Ressource definiert werden. Eine Beziehung muss von einer Entität zur anderen gerichtet sein.

Wie führe ich AWS IoT TwinMaker Knowledge-Graph-Abfragen aus

Bevor Sie den AWS IoT TwinMaker Knowledge Graph verwenden, stellen Sie sicher, dass Sie die folgenden Voraussetzungen erfüllt haben:

- Erstellen Sie einen AWS IoT TwinMaker Arbeitsbereich. Sie können einen Workspace in der AWS IoT TwinMaker Konsole erstellen.
- Machen Sie sich mit dem AWS IoT TwinMaker Entity-Component-System und der Erstellung von Entitäten vertraut. Weitere Informationen finden Sie unter Erstellen Sie Ihre erste Entität.
- Machen Sie sich mit den AWS IoT TwinMaker Datenkonnektoren vertraut. Weitere Informationen finden Sie unter AWS IoT TwinMaker Datenanschlüsse.

> **Note**
>
> Um den AWS IoT TwinMaker Knowledge Graph verwenden zu können, müssen Sie entweder den Standard - oder den gestaffelten Paketpreis gewählt haben. Weitere Informationen finden Sie unter AWS IoT TwinMaker Preismodus wechseln.

Die folgenden Verfahren zeigen Ihnen, wie Sie Abfragen schreiben, ausführen, speichern und bearbeiten.

Öffnen Sie den Abfrage-Editor

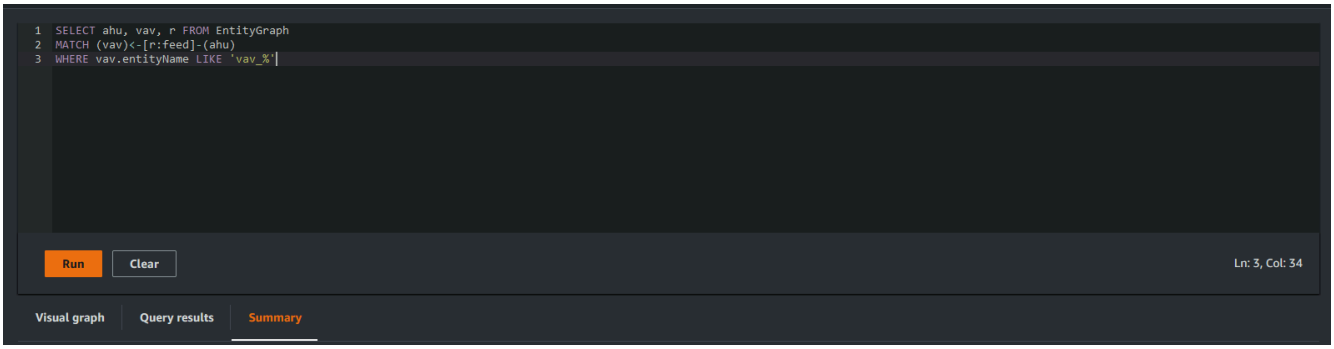
Um zum Knowledge Graph-Abfrage-Editor zu navigieren

1. Öffnen Sie die AWS IoT TwinMaker -Konsole.
2. Öffnen Sie den Arbeitsbereich, in dem Sie den Knowledge Graph verwenden möchten.
3. Wählen Sie im linken Navigationsmenü den Abfrage-Editor.
4. Der Abfrage-Editor wird geöffnet. Sie sind jetzt bereit, Abfragen für die Ressourcen Ihres Workspace auszuführen.

Führen Sie eine Abfrage aus

Um eine Abfrage auszuführen und ein Diagramm zu generieren

1. Wählen Sie im Abfrage-Editor die Registerkarte Editor, um den Syntaxeditor zu öffnen.
2. Schreiben Sie im Editor-Bereich die Abfrage, die Sie für die Ressourcen Ihres Workspace ausführen möchten.



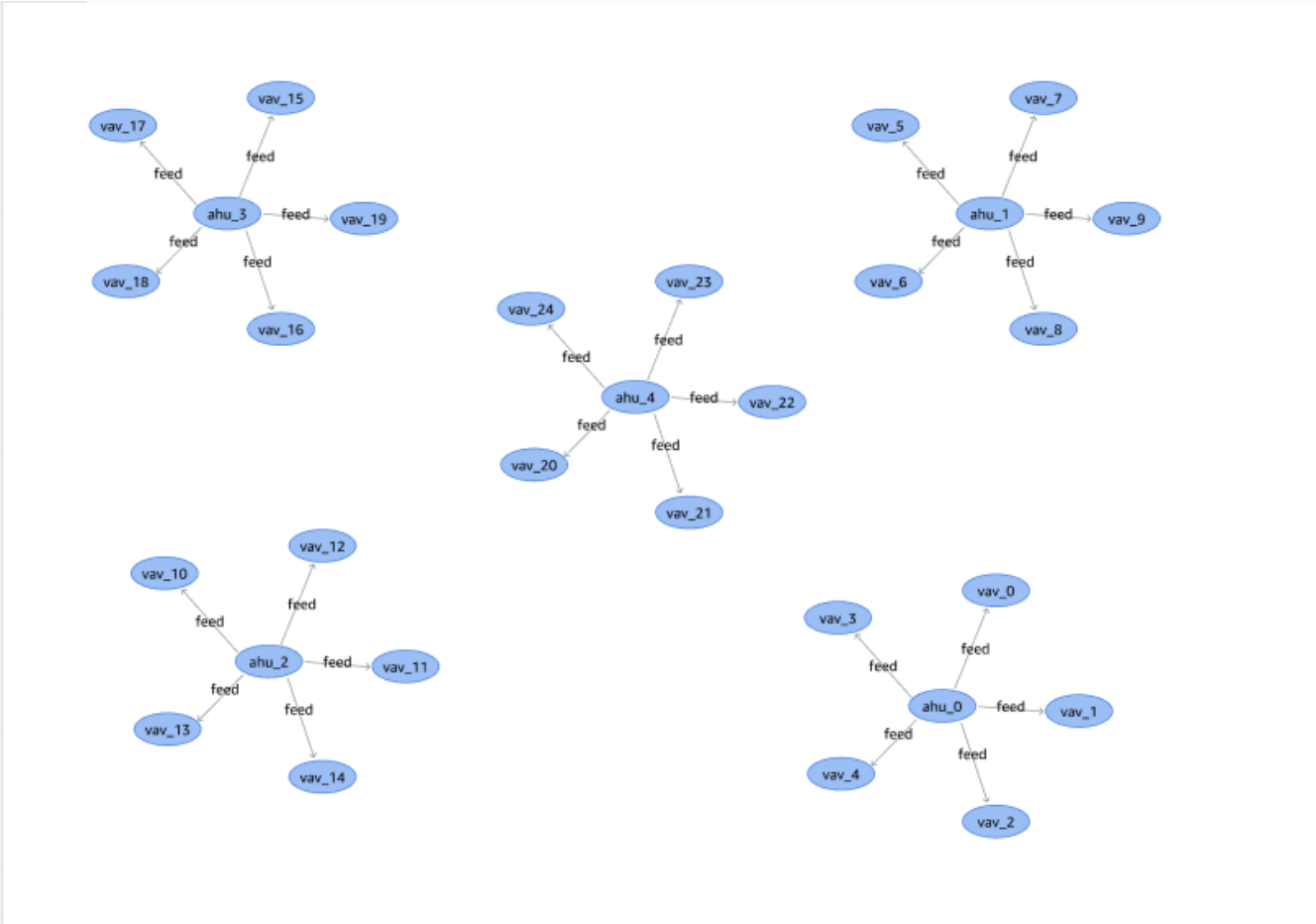
In dem gezeigten Beispiel sucht die Anfrage nach Entitäten, die `vav_%` in ihrem Namen enthalten, und organisiert diese Entitäten dann anhand der `feed` Beziehung zwischen ihnen. Dabei wird der folgende Code verwendet.

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

> **Note**
>
> Die Knowledge Graph-Syntax verwendet PartiQL. Informationen zu dieser Syntax finden Sie unterAWS IoT TwinMaker Knowledge Graph, zusätzliche Ressourcen.

3. Wählen Sie Abfrage ausführen, um die von Ihnen erstellte Anforderung auszuführen.

   Auf der Grundlage Ihrer Anfrage wird ein Diagramm generiert.



   Das oben gezeigte Beispieldiagramm basiert auf dem Abfragebeispiel in Schritt 2.

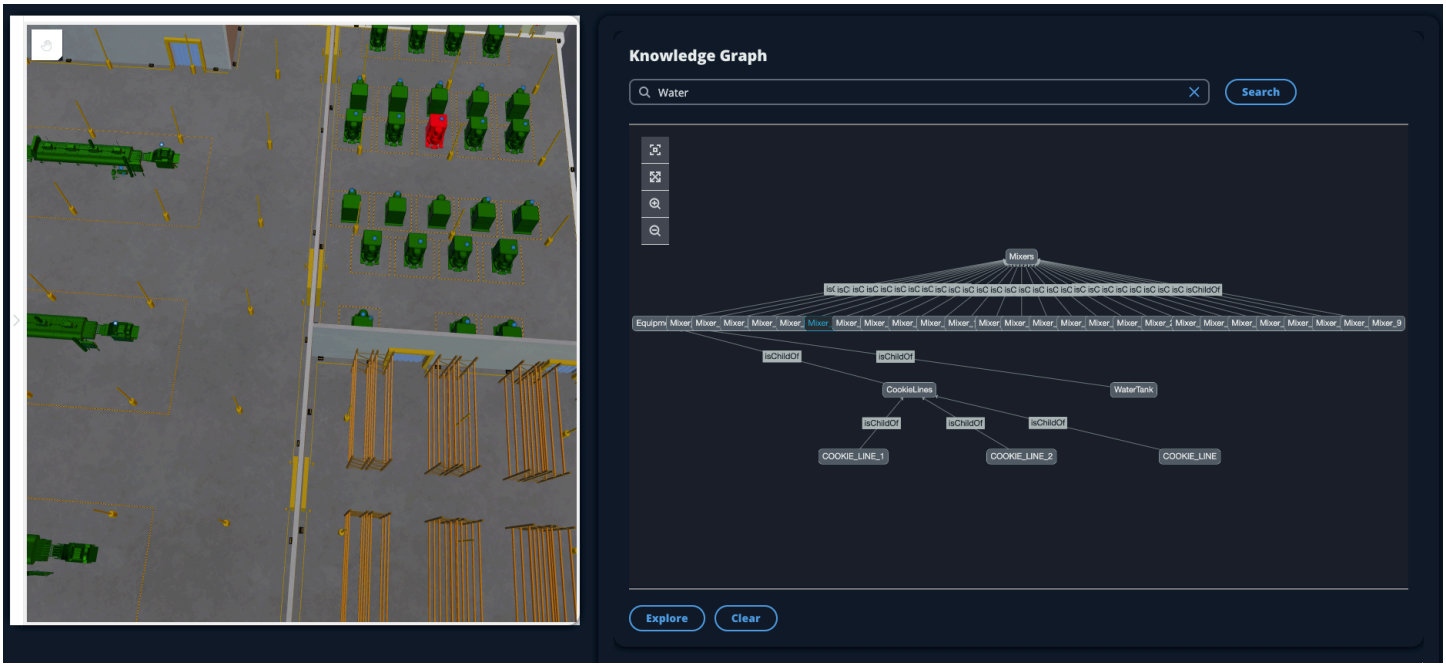
4. Die Ergebnisse der Abfrage werden auch in einer Liste dargestellt. Wählen Sie Ergebnisse aus, um die Abfrageergebnisse in einer Liste anzuzeigen.

5. Wählen Sie optional Exportieren als, um die Abfrageergebnisse im JSON- oder CSV-Format zu exportieren.

Dies deckt die grundlegende Verwendung von Knowledge Graph in der Konsole ab. Weitere Informationen und Beispiele zur Veranschaulichung der Knowledge Graph-Syntax finden Sie unterAWS IoT TwinMaker Knowledge Graph, zusätzliche Ressourcen.

Integration von Knowledge Graph-Szenen

Sie können AWS IoT App-Kit-Komponenten verwenden, um eine Webanwendung zu erstellen, die den Knowledge Graph in Ihre AWS IoT TwinMaker Szenen integriert. Auf diese Weise können Sie Diagramme erstellen, die auf den 3D-Knoten (den 3D-Modellen, die Ihre Ausrüstung oder Systeme repräsentieren) basieren, die in Ihrer Szene vorhanden sind. Um eine Anwendung zu erstellen, die 3D-Knoten aus Ihrer Szene grafisch darstellt, binden Sie die 3D-Knoten zunächst an Entitäten in Ihrem Workspace. Mit diesem Mapping werden die Beziehungen zwischen den in Ihrer Szene vorhandenen 3D-Modellen und den Entitäten in Ihrem Workspace AWS IoT TwinMaker grafisch dargestellt. Anschließend können Sie eine Webanwendung erstellen, 3D-Modelle mit Ihrer Szene auswählen und deren Beziehungen zu anderen Entitäten in einem Grafikformat untersuchen.



Ein Beispiel für eine funktionierende Webanwendung, die die AWS IoT App-Kit-Komponenten verwendet, um Diagramme in einer AWS IoT TwinMaker Szene zu generieren, finden Sie in der AWS IoT TwinMaker Beispiel-React-App auf Github.

AWS IoT TwinMaker Voraussetzungen für Szenengraphen

Bevor Sie eine Web-App erstellen, die AWS IoT TwinMaker Knowledge Graph in Ihren Szenen verwendet, müssen Sie die folgenden Voraussetzungen erfüllen:

- Erstellen Sie einen AWS IoT TwinMaker Arbeitsbereich. Sie können einen Workspace in der AWS IoT TwinMaker Konsole erstellen.
- Machen Sie sich mit dem AWS IoT TwinMaker Entity-Component-System und der Erstellung von Entitäten vertraut. Weitere Informationen finden Sie unter Erstellen Sie Ihre erste Entität.
- Erstellen Sie eine mit 3D-Modellen gefüllte AWS IoT TwinMaker Szene.
- Machen Sie sich mit AWS IoT TwinMaker den Komponenten des AWS IoT App-Kits vertraut. Weitere Informationen zu den AWS IoT TwinMaker Komponenten finden Sie unterErstellen Sie eine benutzerdefinierte Webanwendung mithilfe von AWS IoT TwinMaker UI-Komponenten.
- Machen Sie sich mit den Konzepten und Schlüsselbegriffen von Knowledge-Graphen vertraut. Siehe AWS IoT TwinMaker Kernkonzepte des Wissensgraphen.

> **Note**
>
> Um den AWS IoT TwinMaker Knowledge Graph und alle zugehörigen Funktionen nutzen zu können, müssen Sie entweder den Standard - oder den gestaffelten Paketpreis nutzen. Weitere Informationen zur AWS IoT TwinMaker Preisgestaltung finden Sie unterAWS IoT TwinMaker Preismodus wechseln.

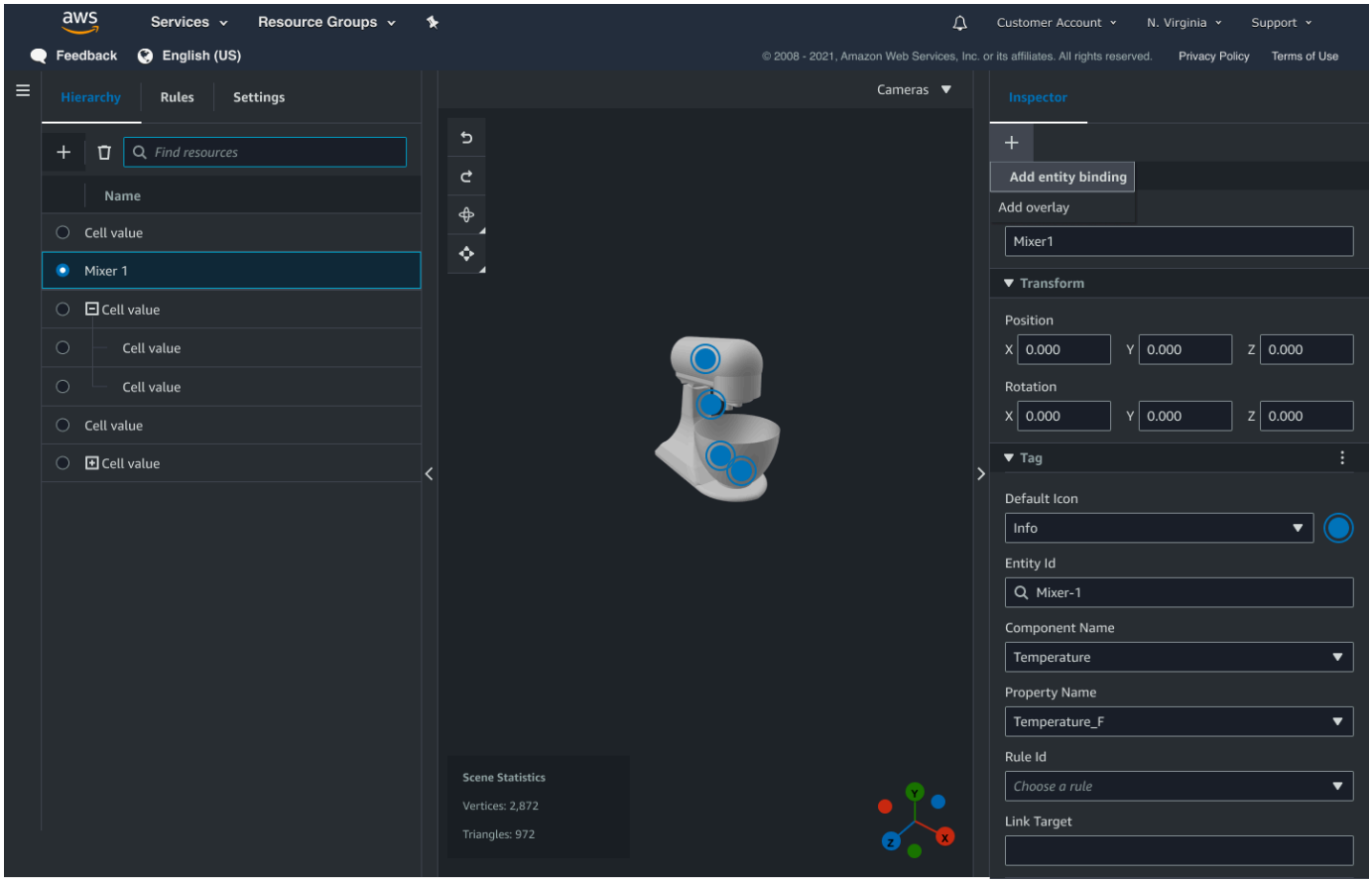
Binden Sie 3D-Knoten in Ihrer Szene

Bevor Sie eine Web-App erstellen, die Knowledge Graph in Ihre Szene integriert, binden Sie die in Ihrer Szene vorhandenen 3D-Modelle, die als 3D-Knoten bezeichnet werden, an die zugehörige Workspace-Entität. Wenn Sie beispielsweise ein Modell einer Mixer-Ausrüstung in einer Szene haben und eine entsprechende Entität aufgerufen wird`mixer_0`, erstellen Sie eine Datenbindung zwischen dem Modell des Mixers und der Entität, die den Mixer darstellt, sodass das Modell und die Entität grafisch dargestellt werden können.

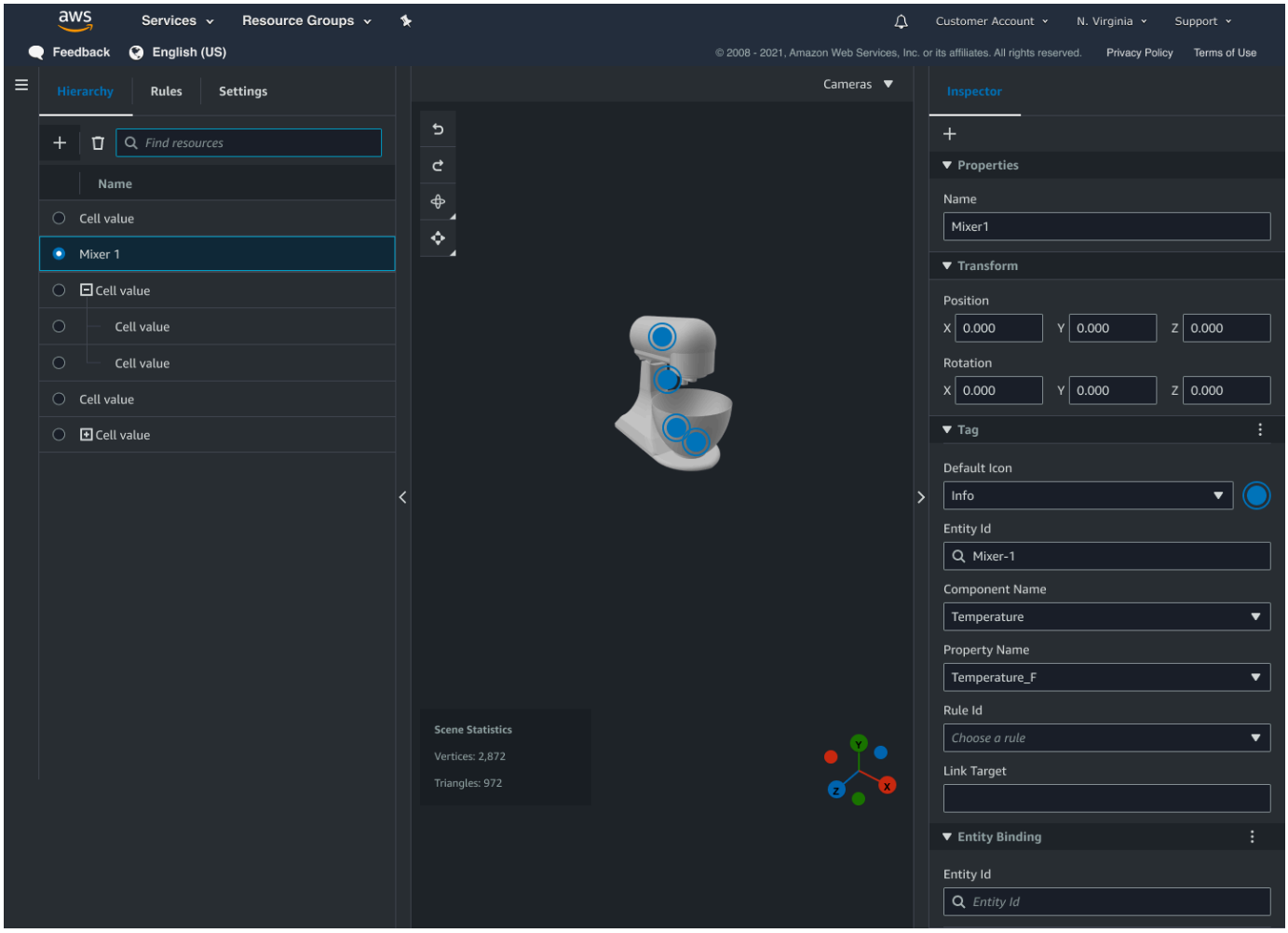
Um eine Datenbindungsaktion durchzuführen

1. Melden Sie sich bei der AWS IoT TwinMaker -Konsole an.

2. Öffnen Sie Ihren Workspace und wählen Sie eine Szene mit den 3D-Knoten aus, die Sie binden möchten.
3. Wählen Sie im Scene Composer einen Knoten (3D-Modell) aus. Wenn Sie einen Knoten auswählen, wird ein Inspektorfenster auf der rechten Seite des Bildschirms geöffnet.
4. Navigieren Sie im Inspektorfenster zum oberen Rand des Bedienfelds und wählen Sie die Schaltfläche + aus. Wählen Sie dann die Option Entitätsbindung hinzufügen. Dadurch wird ein Drop-down-Menü geöffnet, in dem Sie eine Entität auswählen können, die an Ihren aktuell ausgewählten Knoten gebunden werden soll.



5. Wählen Sie im Dropdownmenü Datenbindung die Entitäts-ID aus, die Sie dem 3D-Modell zuordnen möchten. Wählen Sie in den Feldern Komponentenname und Eigenschaftsname die Komponenten und Eigenschaften aus, die Sie binden möchten.



Sobald Sie eine Auswahl für die Felder Entitäts-ID, Komponentenname und Eigenschaftsname getroffen haben, ist die Bindung abgeschlossen.

6. Wiederholen Sie diesen Vorgang für alle Modelle und Entitäten, die Sie grafisch darstellen möchten.

> **Note**
>
> Derselbe Datenbindungsvorgang kann für Ihre Szenen-Tags ausgeführt werden. Wählen Sie einfach ein Tag anstelle einer Entität aus und folgen Sie demselben Verfahren, um das Tag an einen Knoten zu binden.

Erstellen Sie eine Webanwendung

Nachdem Sie Ihre Entitäten gebunden haben, verwenden Sie die AWS IoT App-Kit-Bibliothek, um eine Web-App mit einem Knowledge-Graph-Widget zu erstellen, mit dem Sie Ihre Szene anzeigen und die Beziehungen zwischen Ihren Szenenknoten und Entitäten untersuchen können.

Verwenden Sie die folgenden Ressourcen, um Ihre eigene App zu erstellen:

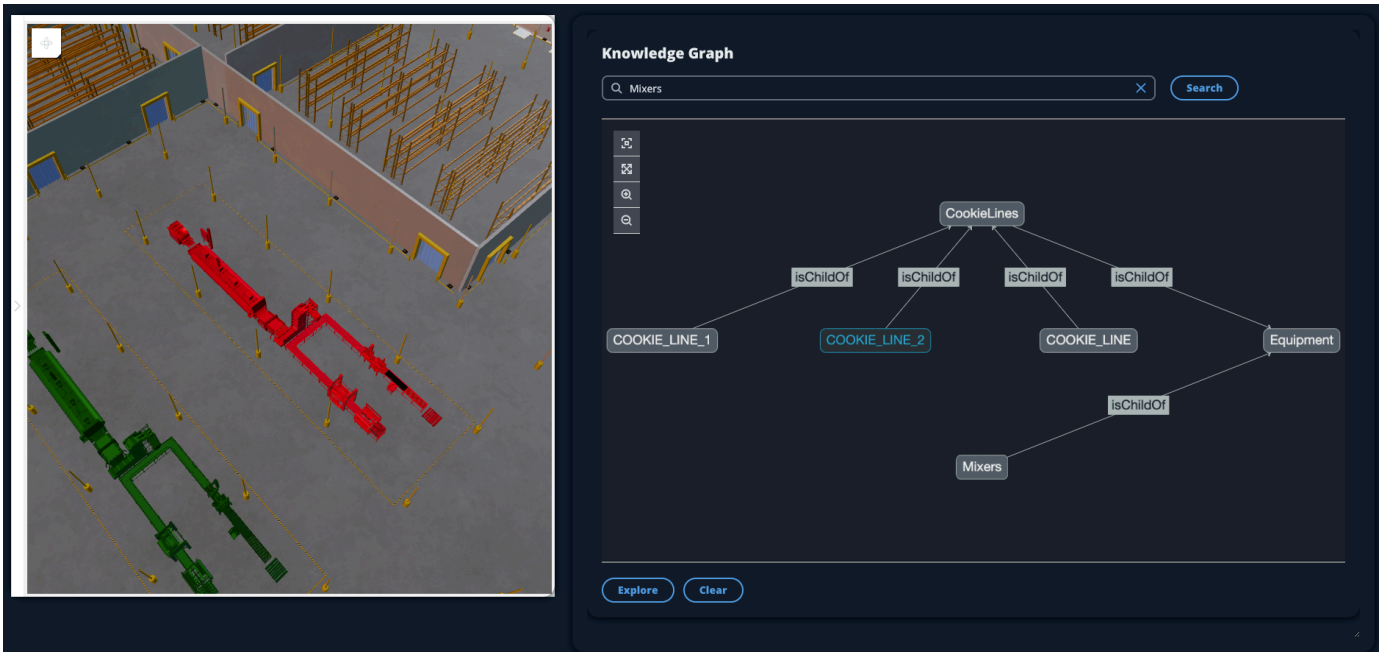
- Die Github-Readme-Dokumentation zum AWS IoT TwinMaker Beispiel für die React-App.
- Die AWS IoT TwinMaker Beispielquelle für die React-App auf Github.
- Die Dokumentation „Erste Schritte“ für das AWS IoT App-Kit.
- Die Dokumentation zur Video Player-Komponente für das AWS IoT App-Kit.
- Die Dokumentation zur Scene Viewer-Komponente für das AWS IoT App-Kit.

Das folgende Verfahren veranschaulicht die Funktionalität der Scene Viewer-Komponente in einer Web-App.

> **Note**
>
> Dieses Verfahren basiert auf der Implementierung der AWS IoT App Kit Scene Viewer-Komponente in der AWS IoT TwinMaker React-Beispiel-App.

1. Öffnen Sie die Scene Viewer-Komponente der AWS IoT TwinMaker React-Beispiel-App. Geben Sie im Suchfeld einen Entitätsnamen oder einen Teil des Entitätsnamens ein (bei der Suche wird Groß- und Kleinschreibung beachtet) und wählen Sie dann die Schaltfläche Suchen aus. Wenn ein Modell an die Entitäts-ID gebunden ist, wird das Modell in der Szene hervorgehoben und ein Knoten der Entität wird im Scene Viewer-Fenster angezeigt.

2. Um ein Diagramm aller Beziehungen zu erstellen, wählen Sie einen Knoten im Scene Viewer-Widget aus und klicken Sie auf die Schaltfläche „Erkunden“.



3. Drücken Sie die Taste Löschen, um Ihre aktuelle Grafikauswahl zu löschen und von vorne zu beginnen.

Wie benutzt man AWS IoT TwinMaker Knowledge Graph mit Grafana

In diesem Abschnitt erfahren Sie, wie Sie Ihrem AWS IoT TwinMaker Grafana-Dashboard einen Abfrage-Editor hinzufügen, um Abfragen auszuführen und anzuzeigen.

AWS IoT TwinMaker Voraussetzungen für den Abfrage-Editor

Bevor Sie den AWS IoT TwinMaker Knowledge Graph in Grafana verwenden, müssen Sie die folgenden Voraussetzungen erfüllen:

- Erstellen Sie einen AWS IoT TwinMaker Arbeitsbereich. Sie können einen Workspace in der AWS IoT TwinMaker Konsole erstellen.
- AWS IoT TwinMaker Für die Verwendung mit Grafana konfigurieren. Detaillierte Anweisungen finden Sie unter AWS IoT TwinMaker Grafana-Dashboard-Integration.

> **Note**
>
> Um den AWS IoT TwinMaker Knowledge Graph verwenden zu können, müssen Sie entweder den Standard - oder den gestaffelten Paketpreismodus verwenden. Weitere Informationen finden Sie unter AWS IoT TwinMaker Preismodus wechseln.

AWS IoT TwinMaker Berechtigungen für den Abfrageeditor

Um den AWS IoT TwinMaker Abfrage-Editor in Grafana verwenden zu können, benötigen Sie eine IAM-Rolle mit der Berechtigung für die Aktion. `iottwinmaker:ExecuteQuery` Fügen Sie diese Berechtigung zu Ihrer Workspace-Dashboard-Rolle hinzu, wie in diesem Beispiel gezeigt:

JSON

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

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

> **Note**
>
> Wenn Sie Ihre AWS IoT TwinMaker Grafana-Datenquelle konfigurieren, stellen Sie sicher, dass Sie die Rolle mit dieser Berechtigung für das ARN-Feld Rolle annehmen verwenden. Nachdem Sie es hinzugefügt haben, können Sie Ihren Workspace aus der Drop-down-Liste neben Workspace auswählen.

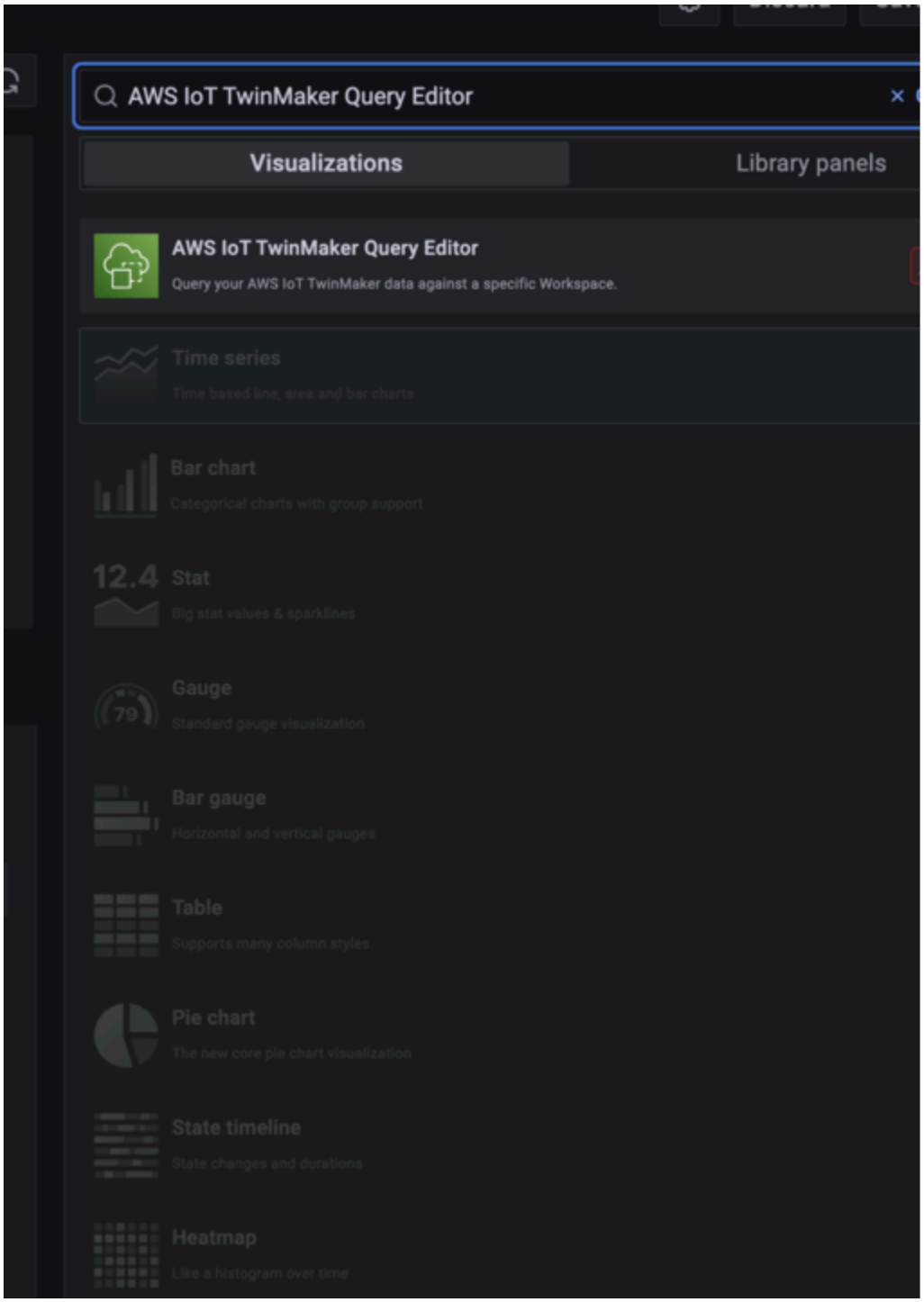
Weitere Informationen finden Sie unter Eine Dashboard-IAM-Rolle erstellen.

Richten Sie den AWS IoT TwinMaker Abfrage-Editor ein

Um ein neues Grafana-Dashboard-Panel für Knowledge Graph einzurichten

1. Öffnen Sie Ihr AWS IoT TwinMaker Grafana-Dashboard.

2. Erstellen Sie ein neues Dashboard-Panel. Ausführliche Schritte zum Erstellen eines Panels finden Sie unter Erstellen eines Dashboards in der Grafana-Dokumentation.
3. Wählen AWS IoT TwinMaker Sie aus der Liste der Visualisierungen den Abfrage-Editor aus.



4. Wählen Sie die Datenquelle aus, für die Abfragen ausgeführt werden sollen.
5. (Optional) Fügen Sie im dafür vorgesehenen Feld einen Namen für das neue Panel hinzu.

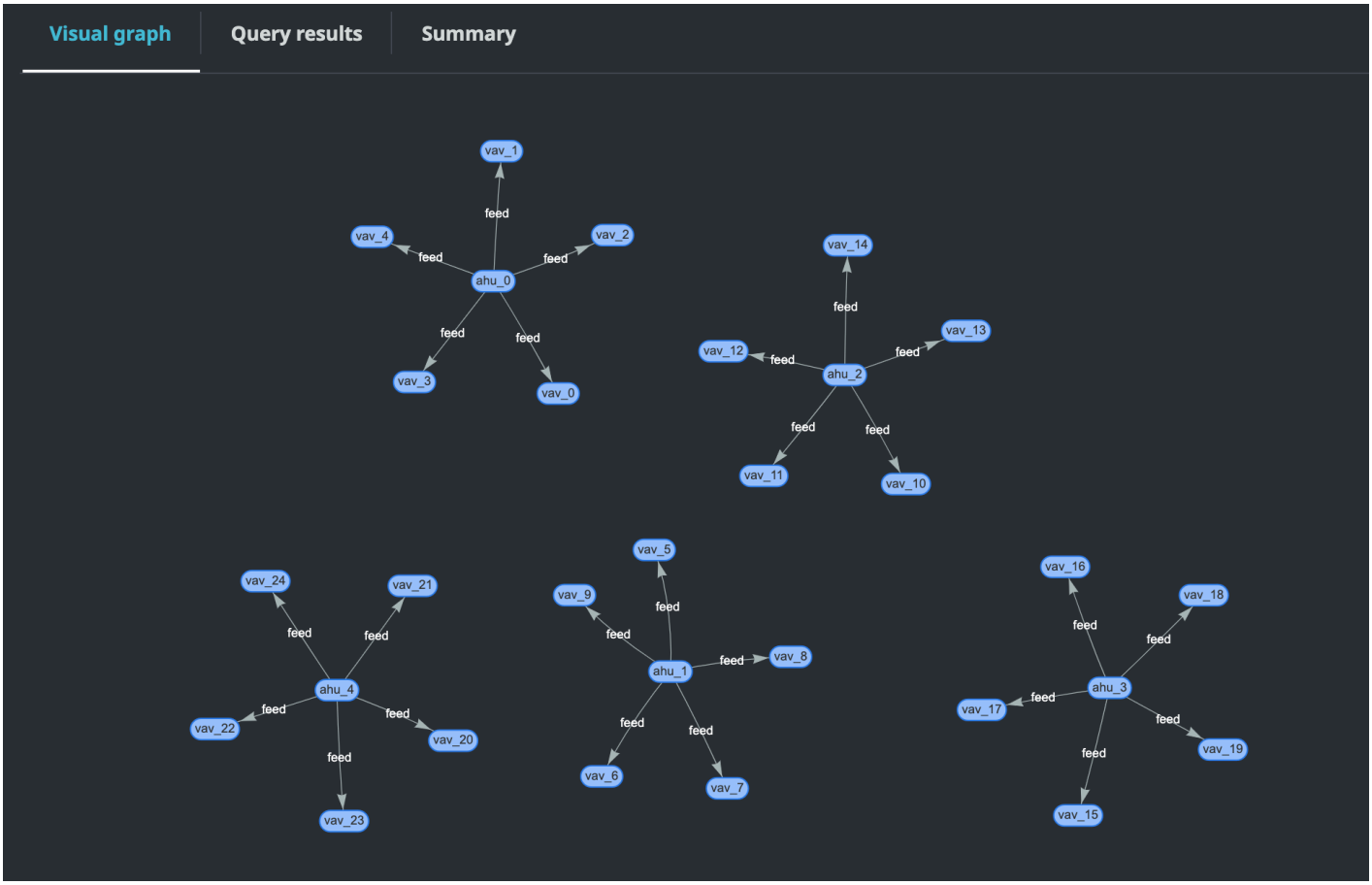
6. Wählen Sie Anwenden, um Ihr neues Panel zu speichern und zu bestätigen.

Das Knowledge-Graph-Panel funktioniert auf ähnliche Weise wie der Abfrage-Editor in der AWS IoT TwinMaker Konsole. Sie können Abfragen, die Sie im Panel stellen, ausführen, schreiben und löschen. Weitere Informationen zum Schreiben von Abfragen finden Sie unterAWS IoT TwinMaker Knowledge Graph, zusätzliche Ressourcen.

Wie benutzt man den AWS IoT TwinMaker Abfrage-Editor

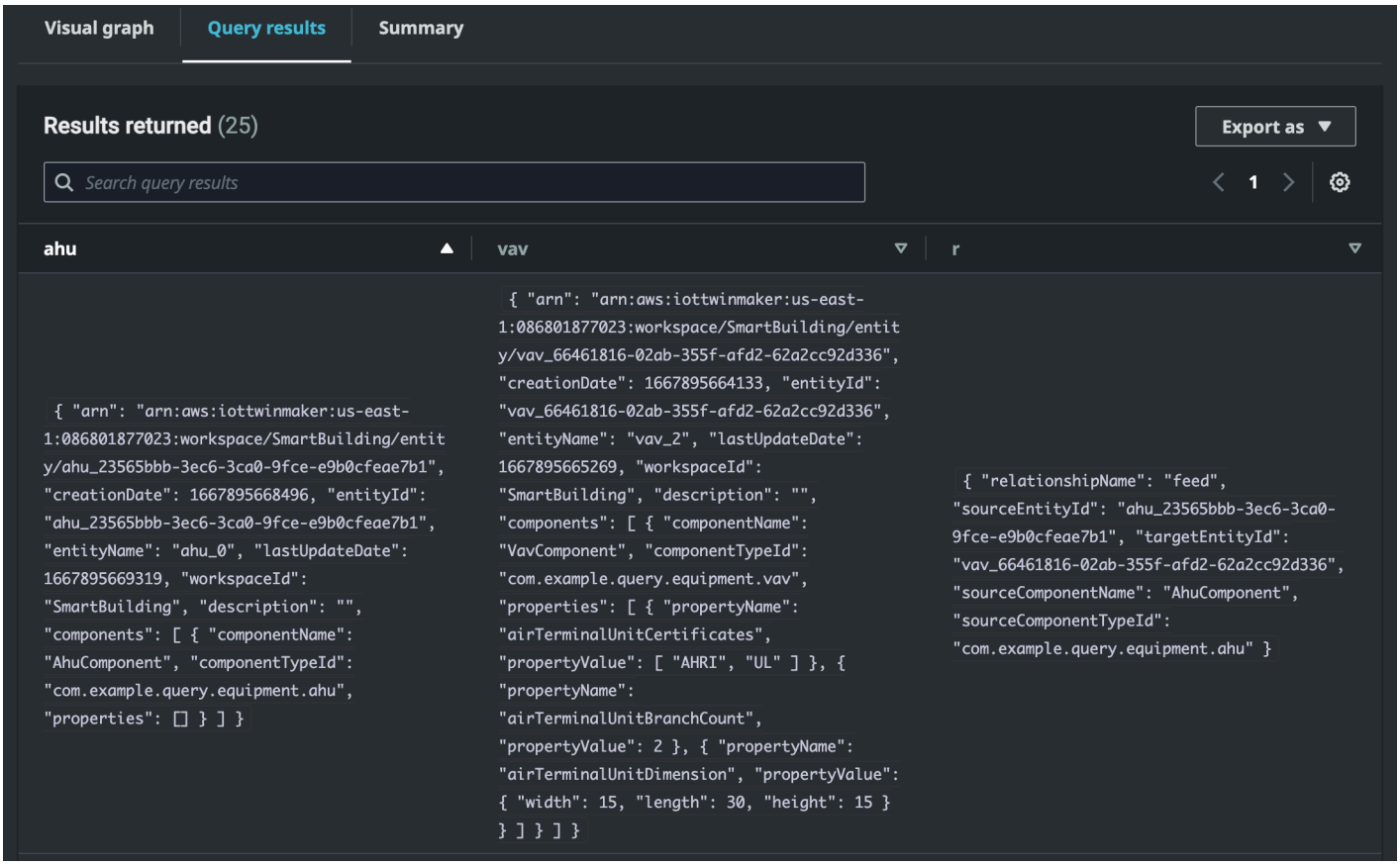
Die Ergebnisse Ihrer Abfragen werden auf drei Arten angezeigt, wie in den folgenden Bildern dargestellt: in einem Diagramm visualisiert, in einer Tabelle aufgeführt oder als Zusammenfassung der Ausführung dargestellt.

- Graphische Visualisierung:



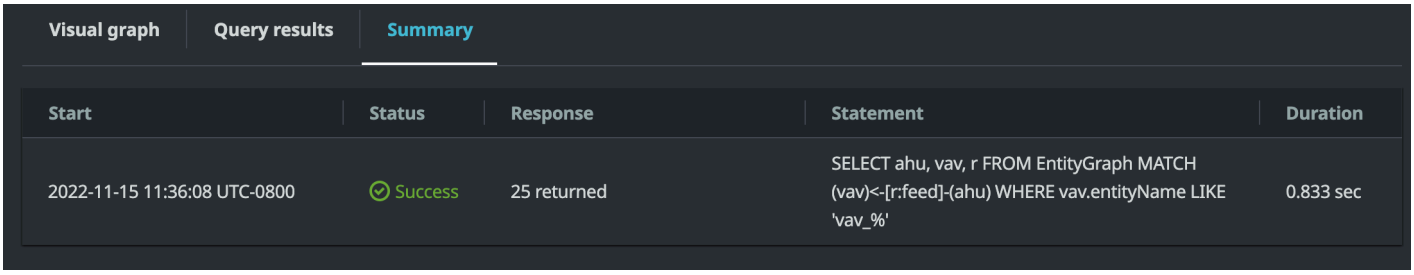
Das visuelle Diagramm zeigt nur Daten für Abfragen an, deren Ergebnis mindestens eine Beziehung enthält. In der Grafik werden Entitäten als Knoten und Beziehungen als gerichtete Kanten im Diagramm angezeigt.

- Tabellarische Daten:



Das tabellarische Datenformat zeigt die Daten für alle Abfragen an. Sie können die Tabelle nach bestimmten Ergebnissen oder Teilmengen der Ergebnisse durchsuchen. Die Daten können im JSON- oder CSV-Format exportiert werden.

- Zusammenfassung ausführen



In der Ausführungszusammenfassung werden die Abfrage und Metadaten zum Status der Abfrage angezeigt.

AWS IoT TwinMaker Knowledge Graph, zusätzliche Ressourcen

Dieser Abschnitt enthält grundlegende Beispiele für die PartiQL-Syntax, die zum Schreiben von Abfragen im Knowledge Graph verwendet wird, sowie Links zur PartiQL-Dokumentation, die Informationen zum Knowledge Graph-Datenmodell enthält.

- Dokumentation zum PartiQL-Graphdatenmodell
- Dokumentation zur PartiQL-Graphabfrage

Diese Reihe von Beispielen zeigt grundlegende Abfragen mit ihren Antworten. Verwenden Sie dies als Referenz, um Ihre eigenen Abfragen zu schreiben.

Grundlegende Abfragen

- Ruft alle Entitäten mit einem Filter ab

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

Diese Abfrage gibt alle Entitäten in einem Workspace mit dem Namen zurück`room_0`.

`FROM`Klausel: `EntityGraph` ist die Grafiksammlung, die alle Entitäten und ihre Beziehungen in einem Workspace enthält. Diese Sammlung wird automatisch auf der AWS IoT TwinMaker Grundlage der Entitäten in Ihrem Workspace erstellt und verwaltet.

`MATCH`Klausel: gibt ein Muster an, das einem Teil des Diagramms entspricht. In diesem Fall `(entity)` entspricht das Muster jedem Knoten im Diagramm und ist an die Entitätsvariable gebunden. Auf die `FROM` Klausel muss die `MATCH` Klausel folgen.

`WHERE`Klausel: gibt einen Filter für das `entityName` Feld des Knotens an, mit dem der Wert übereinstimmen muss`room_0`.

`SELECT`Klausel: spezifiziert die `entity` Variable, sodass der gesamte Entitätsknoten zurückgegeben wird.

Antwort:

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

Die `columnDescriptions` gibt Metadaten über die Spalte zurück, z. B. den Namen und den Typ. Der zurückgegebene Typ ist`NODE`. Dies zeigt an, dass der gesamte Knoten zurückgegeben wurde. Andere Werte für den Typ können sein`EDGE`, die auf eine Beziehung

hinweisen würdenVALUE, oder die auf einen Skalarwert hinweisen würden, z. B. eine Ganzzahl oder eine Zeichenfolge.

Das rows gibt eine Liste von Zeilen zurück. Da nur eine Entität gefunden wurde, rowData wird eine zurückgegeben, die alle Felder einer Entität enthält.

> **Note**
>
> Im Gegensatz zu SQL, wo Sie nur Skalarwerte zurückgeben können, können Sie ein Objekt (als JSON) mit PartiQL zurückgeben.

Jeder Knoten enthält alle Felder auf Entitätsebene wie entityId arn und components Felder auf Komponentenebene wie und sowie Felder auf Eigenschaftsebene properties wie componentTypeId undcomponentName, alles als verschachteltes JSON. propertyName propertyValue

- Ruft alle Beziehungen mit einem Filter ab:

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

Diese Abfrage gibt alle Beziehungen in einem Arbeitsbereich mit dem Beziehungsnamen zurückisLocationOf.

Die MATCH Klausel: gibt ein Muster an, das zwei Knoten (gekennzeichnet durch( )) entspricht, die durch eine gerichtete Kante (gekennzeichnet durch-[]->) miteinander verbunden und an eine Variable gebunden sind, die aufgerufen wirdrelationship.

Die WHERE Klausel: gibt einen Filter für das relationshipName Feld der Kante an, in dem sich der Wert befindetisLocationOf.

Die SELECT Klausel: gibt die Beziehungsvariable an, sodass der gesamte Kantenknoten zurückgegeben wird.

Antwort

```
{
    "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
    ]
}
```

Der Typ der Spalte in `columnDescriptions` ist ein`EDGE`.

Jedes `rowData` steht für eine Kante mit Feldern wie`relationshipName`. Dies entspricht dem Namen der Beziehungseigenschaft, der für die Entität definiert ist. Die`sourceEntityId`, `sourceComponentName` und `sourceComponentTypeId` geben Auskunft darüber, für welche Entität und Komponente die Beziehungseigenschaft definiert wurde. Das `targetEntityId` gibt an, auf welche Entität diese Beziehung zeigt.

- Ruft alle Entitäten mit einer bestimmten Beziehung zu einer bestimmten Entität ab

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

Diese Abfrage gibt alle Entitätsnamen aller Entitäten zurück, die eine `isLocationOf` Beziehung zu der `room_0` Entität haben.

Die `MATCH` Klausel: gibt ein Muster an, das zwei beliebigen Knoten (`e1`,`e2`) entspricht, die eine gerichtete Kante (`r`) haben.

Die `WHERE` Klausel: gibt einen Filter für den Beziehungsnamen und den Namen der Quellentität an.

Die `SELECT` Klausel: gibt das `entityName` Feld im `e2` Knoten zurück.

Antwort

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

In den ColumnDescriptions ist der Typ der Spalte, `VALUE` da es sich um eine `entityName` Zeichenfolge handelt.

Eine Entität,`floor_0`, wird zurückgegeben.

SPIEL

Die folgenden Muster werden in einer `MATCH` Klausel unterstützt:

- Entspricht dem Knoten 'b', der auf den Knoten 'a' zeigt:

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

- Ordnet Knoten 'a' zu, der auf Knoten 'b' zeigt:

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

  Es gibt keine Variable, die an eine Beziehung gebunden ist, vorausgesetzt, dass für die Beziehung kein Filter angegeben werden muss.

- Ordnen Sie Knoten 'a', der auf Knoten 'b' zeigt, und Knoten 'b', der auf Knoten 'a' zeigt, zu:

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

Dadurch werden zwei Treffer zurückgegeben: eine von 'a' nach 'b' und eine weitere von 'b' nach 'a', daher wird empfohlen, wo immer möglich gerichtete Kanten zu verwenden.

- Der Beziehungsname ist auch eine Bezeichnung für das Eigenschaftsdiagramm`EntityGraph`, sodass Sie den Beziehungsnamen einfach hinter einem Doppelpunkt (:) angeben können, anstatt `rel.relationshipName` in der `WHERE` Klausel einen Filter für anzugeben.

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

- Verkettung: Muster können so verkettet werden, dass sie in mehreren Beziehungen übereinstimmen.

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

- Variable Hop-Muster können sich auch über mehrere Knoten und Kanten erstrecken:

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

Diese Abfrage entspricht jedem Muster mit ausgehenden Kanten von Knoten 'a' innerhalb von 1 bis 5 Hops. Die zulässigen Quantifizierer sind:

`{m,n}`- zwischen m und n Wiederholungen

`{m,}`- m oder mehr Wiederholungen.

VON:

Ein Entitätsknoten kann verschachtelte Daten enthalten, z. B. Komponenten, die wiederum weitere verschachtelte Daten wie Eigenschaften enthalten. Auf diese kann zugegriffen werden, indem das Ergebnis des MATCH-Musters unverschachtelt wird.

```
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'
```

Greifen Sie auf verschachtelte Felder zu, indem Sie Punkte in eine Variable einfügen. `.` Ein Komma (,) wird verwendet, um Entitäten mit den darin enthaltenen Komponenten und dann den Eigenschaften innerhalb dieser Komponenten zu trennen (oder zu verbinden). `AS`wird verwendet,

um eine Variable an die nicht verschachtelten Variablen zu binden, sodass sie in den OR-Klauseln verwendet werden können. `WHERE SELECT` Diese Abfrage gibt alle Entitäten zurück, die eine mit einem Wert benannte Eigenschaft `meeting` in einer Komponente `roomFunction` mit der Komponententyp-ID enthalten `com.example.query.construction.room`

Um auf mehrere verschachtelte Felder eines Felds zuzugreifen, z. B. auf mehrere Komponenten in einer Entität, verwenden Sie die Komma-Notation, um eine Verknüpfung durchzuführen.

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

WÄHLEN SIE AUS:

- Einen Knoten zurückgeben:

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

- Gibt eine Kante zurück:

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

- Gibt einen Skalarwert zurück:

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

  Formatieren Sie den Namen des Ausgabefeldes, indem Sie ihn mit einem Alias versehen. `AS` Hier wird statt `propertyValue` als Spaltenname in der Antwort `roomfunction` zurückgegeben.

- Aliase zurückgeben:

  ```
  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%'
  ```

Es wird dringend empfohlen, Aliase zu verwenden, um explizit zu sein, die Lesbarkeit zu erhöhen und Mehrdeutigkeiten in Ihren Abfragen zu vermeiden.

WO:

- Die unterstützten logischen Operatoren sind `AND`, `NOT`, und `OR`.
- Die unterstützten Vergleichsoperatoren sind `<`, `<=`, `>`, `>=`, `=`, und `!=`.
- Verwenden Sie das `IN` Schlüsselwort, wenn Sie mehrere `OR` Bedingungen für dasselbe Feld angeben möchten.
- Filtern Sie nach einem Entitäts-, Komponenten- oder Eigenschaftsfeld:

```
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'
```

- Filtern Sie nach der `configuration` Eigenschaft. Hier `unit` ist der Schlüssel in der Konfigurationsübersicht und der `Celsius` Wert.

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

- Prüfen Sie, ob eine Karteneigenschaft einen bestimmten Schlüssel und Wert enthält:

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

- Prüft, ob eine Karteneigenschaft einen bestimmten Schlüssel enthält:

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

- Prüft, ob eine Listeneigenschaft einen bestimmten Wert enthält:

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

- Verwenden Sie die `lower()` Funktion für Vergleiche ohne Berücksichtigung der Groß- und Kleinschreibung. Standardmäßig wird bei allen Vergleichen zwischen Groß- und Kleinschreibung unterschieden.

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

WIE:

Nützlich, wenn Sie den genauen Wert für ein Feld nicht kennen und eine Volltextsuche für das angegebene Feld durchführen können. `%`steht für Null oder mehr.

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

- Infix-Suche: `%room%`
- Präfix-Suche: `room%`
- Suffix-Suche: `%room`
- Wenn Sie '%' in Ihren Werten haben, fügen Sie ein Escape-Zeichen ein `LIKE` und geben Sie das Escape-Zeichen mit `ESCAPE` an.

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

UNVERWECHSELBAR:

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

- Das `DISTINCT` Schlüsselwort entfernt Duplikate aus dem Endergebnis.

  `DISTINCT`wird bei komplexen Datentypen nicht unterstützt.

ZÄHLEN

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

- Das `COUNT` Schlüsselwort berechnet die Anzahl der Elemente in einem Abfrageergebnis.
- `COUNT`wird für verschachtelte komplexe Felder und Felder mit Grafikmustern nicht unterstützt.
- `COUNT`Aggregation wird bei `DISTINCT` verschachtelten Abfragen nicht unterstützt.

  Beispielsweise wird `COUNT(DISTINCT e.entityId)` nicht unterstützt.

PFAD

Die folgenden Musterprojektionen werden bei Abfragen mithilfe der Pfadprojektion unterstützt:

- Abfragen mit variablem Hop

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

Bei dieser Abfrage werden Knoten-Metadaten beliebiger Muster mit ausgehenden Kanten von Knoten A innerhalb von 1 bis 3 Hops abgeglichen und projiziert.

- Hop-Abfragen wurden behoben

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

Diese Abfrage gleicht Metadaten von Entitäten und eingehenden Kanten ab und projiziert sie auf b.

- Ungerichtete Abfragen

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

Diese Abfrage vergleicht und projiziert die Metadaten von Knoten in 1-Hop-Mustern, die a und c über b verbinden.

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

Diese PATH Abfrageantwort besteht nur aus Metadaten, die alle Knoten und Kanten der einzelnen Knoten path/pattern zwischen a und c über b identifizieren.

LIMIT und OFFSET:

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

`LIMIT`gibt die Anzahl der Ergebnisse an, die in der Abfrage zurückgegeben werden sollen, und `OFFSET` gibt die Anzahl der zu überspringenden Ergebnisse an.

LIMIT und MaxResults:

Das folgende Beispiel zeigt eine Abfrage, die insgesamt 500 Ergebnisse zurückgibt, aber nur 50 gleichzeitig pro API-Aufruf anzeigt. Dieses Muster kann verwendet werden, wenn Sie die Anzahl der angezeigten Ergebnisse einschränken müssen, z. B. wenn Sie nur 50 Ergebnisse in einer Benutzeroberfläche anzeigen können.

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

- Das `LIMIT` Schlüsselwort wirkt sich auf die Abfrage aus und begrenzt die resultierenden Zeilen. Wenn Sie die Anzahl der pro API-Aufruf zurückgegebenen Ergebnisse kontrollieren möchten, ohne die Gesamtzahl der zurückgegebenen Ergebnisse zu begrenzen, verwenden Sie`LIMIT`.
- `max-results`ist ein optionaler Parameter für die ExecuteQuery API-Aktion. `max-results`gilt nur für die API und dafür, wie Ergebnisse im Rahmen der obigen Abfrage gelesen werden.

  Durch die Verwendung `max-results` in einer Abfrage können Sie die Anzahl der angezeigten Ergebnisse reduzieren, ohne die tatsächliche Anzahl der zurückgegebenen Ergebnisse zu begrenzen.

Die folgende Abfrage durchläuft die nächste Ergebnisseite. Diese Abfrage verwendet den `ExecuteQuery` API-Aufruf, um die Zeilen 51-100 zurückzugeben, wobei die nächste Ergebnisseite durch das angegeben wird `next-token` — in diesem Fall lautet das Token:. `"H7kyGmvK376L"`

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

- Die `next-token` Zeichenfolge gibt die nächste Ergebnisseite an. Weitere Informationen finden Sie in der ExecuteQueryAPI-Aktion.

AWS IoT TwinMaker Die Knowledge-Graph-Abfrage hat die folgenden Beschränkungen:

Name des Grenzwerts	Kontingent	Einstellbar
Zeitüberschreitung für die Abfrageausführung	10 Sekunden	Nein
Maximale Anzahl von Hops	10	Ja
Maximale Anzahl von Self `JOIN` s	20	Ja
Maximale Anzahl projizierter Felder	20	Ja
Maximale Anzahl bedingter Ausdrücke (`AND`,`OR`,`NOT`)	10	Ja
Maximale Länge eines `LIKE` Ausdrucksmusters (einschli eßlich Platzhaltern und Escapes)	20	Ja
Maximale Anzahl von Elementen, die in einer Klausel angegeben werden können `IN`	10	Ja
Maximalwert für `OFFSET`	3000	Ja
Maximalwert für `LIMIT`	3000	Ja
Höchstwert für Traversie rungen (+) `OFFSET` `LIMIT`	3000	Ja

Asset-Synchronisierung mit AWS IoT SiteWise

AWS IoT TwinMaker unterstützt die Asset-Synchronisierung (Asset-Synchronisierung) für Ihre AWS IoT SiteWise Assets und Asset-Modelle. Unter Verwendung des AWS IoT SiteWise Komponententyps verwendet die Asset-Synchronisierung vorhandene AWS IoT SiteWise Assets und Asset-Modelle und konvertiert diese Ressourcen in AWS IoT TwinMaker Entitäten, Komponenten und Komponententypen. In den folgenden Abschnitten erfahren Sie, wie Sie die Asset-Synchronisierung konfigurieren und welche AWS IoT SiteWise Assets und Asset-Modelle mit Ihrem AWS IoT TwinMaker Workspace synchronisiert werden können.

Themen

- Verwenden Sie die Asset-Synchronisierung mit AWS IoT SiteWise
- Unterschiede zwischen benutzerdefinierten und Standardarbeitsbereichen
- Ressourcen, die synchronisiert wurden von AWS IoT SiteWise
- Analysieren Sie den Synchronisierungsstatus und die Fehler
- Löschen Sie einen Synchronisierungsauftrag
- Beschränkungen der Asset-Synchronisierung

Verwenden Sie die Asset-Synchronisierung mit AWS IoT SiteWise

In diesem Thema erfahren Sie, wie Sie die AWS IoT SiteWise Asset-Synchronisierung aktivieren und konfigurieren. Folgen Sie den entsprechenden Verfahren, je nachdem, welchen Workspace-Typ Sie verwenden.

> **Important**
>
> Informationen the section called “Unterschiede zwischen benutzerdefinierten und Standardarbeitsbereichen” zu den Unterschieden zwischen dem benutzerdefinierten und dem Standardarbeitsbereich finden Sie unter.

Themen

- Verwenden eines benutzerdefinierten Arbeitsbereichs
- Verwenden des Io TSite WiseDefaultWorkspace

Verwenden eines benutzerdefinierten Arbeitsbereichs

Überprüfe diese Voraussetzungen, bevor du die Asset-Synchronisierung aktivierst.

Voraussetzungen

Vor der Verwendung AWS IoT SiteWise muss Folgendes abgeschlossen sein:

- Sie haben einen AWS IoT TwinMaker Arbeitsbereich.
- Sie haben Vermögenswerte und Vermögensmodelle dabei AWS IoT SiteWise. Weitere Informationen finden Sie unter Asset-Modelle erstellen.
- Eine bestehende IAM-Rolle mit Leseberechtigungen für die folgenden AWS IoT SiteWise Aktionen:
  - `ListAssets`
  - `ListAssetModels`
  - `DescribeAsset`
  - `DescribeAssetModel`
- Die IAM-Rolle muss über die folgenden Schreibberechtigungen verfügen für: AWS IoT TwinMaker
  - `CreateEntity`
  - `UpdateEntity`
  - `DeleteEntity`
  - `CreateComponentType`
  - `UpdateComponentType`
  - `DeleteComponentType`
  - `ListEntities`
  - `GetEntity`
  - `ListComponentTypes`

  Verwenden Sie die folgende IAM-Rolle als Vorlage für die erforderliche Rolle:

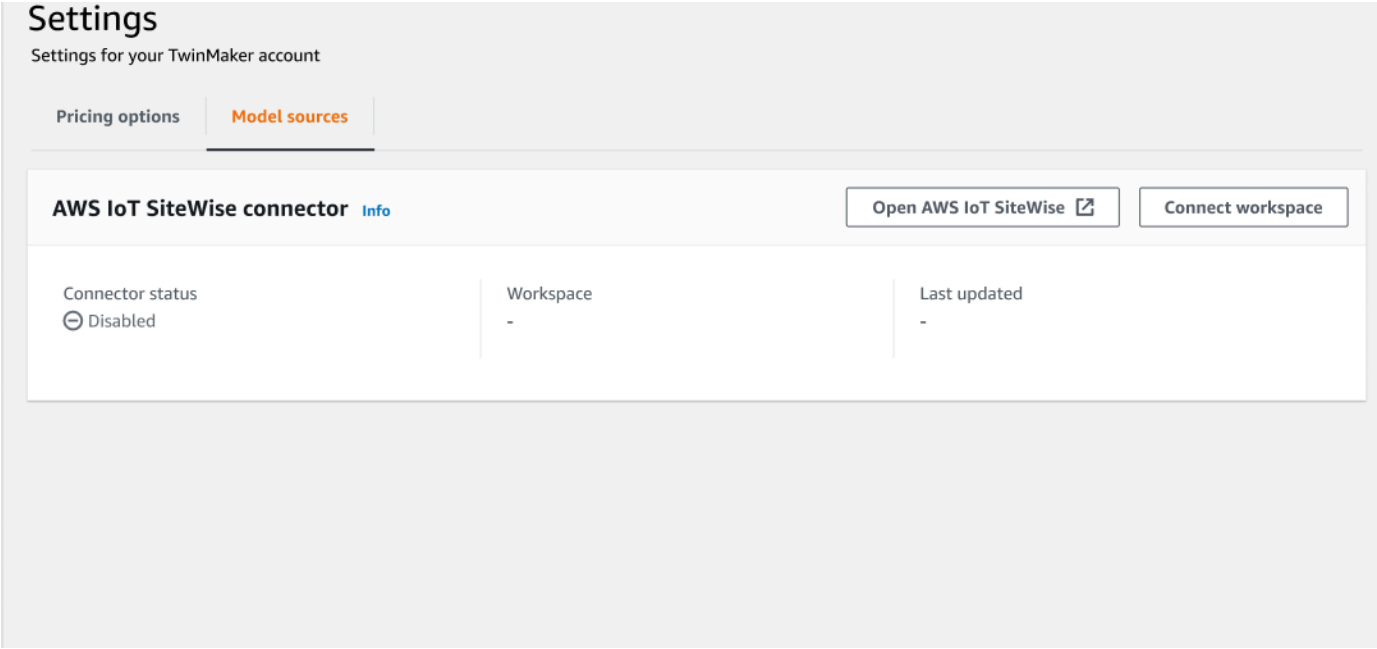
```
// trust relationships
 {
    {
        "Version": "2012-10-17",
        "Statement": [
```

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

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

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

            ]
        }
    ]
}
```

Gehen Sie wie folgt vor, um die AWS IoT SiteWise Asset-Synchronisierung zu aktivieren und zu konfigurieren.

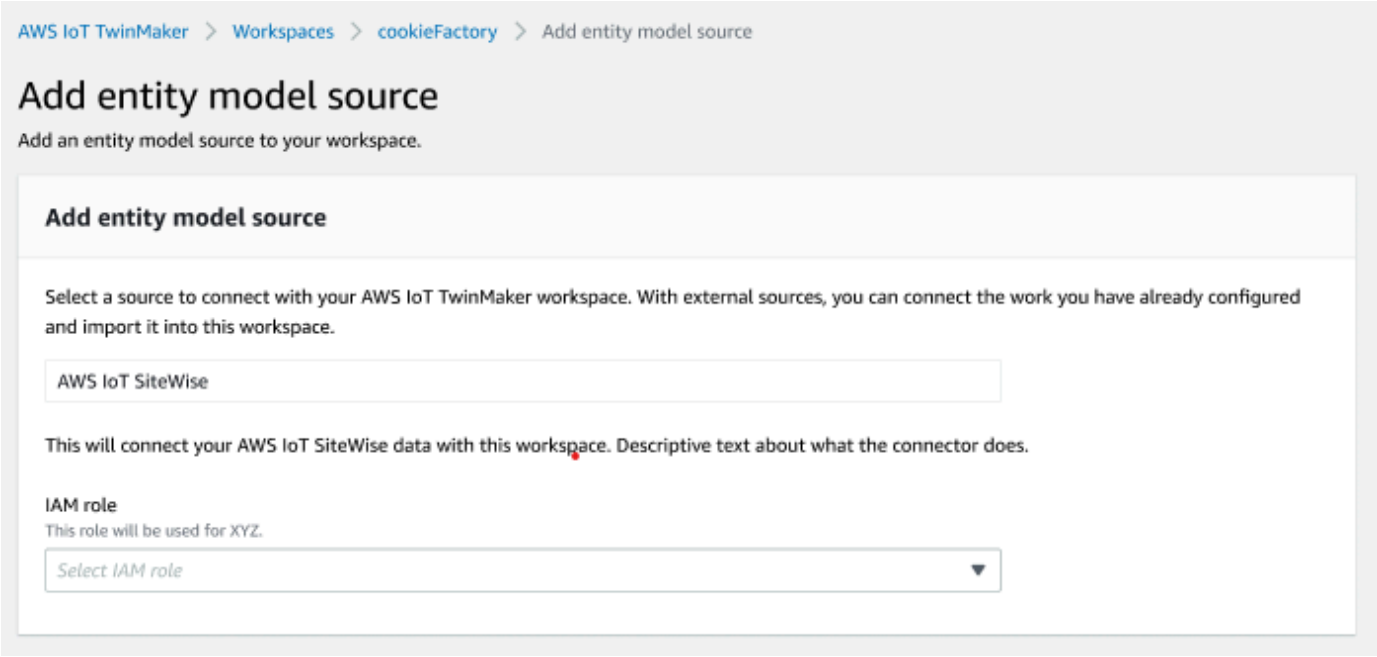
1. Navigieren Sie in der AWS IoT TwinMaker Konsole zur Seite Einstellungen.
2. Öffnen Sie die Registerkarte Modellquellen.



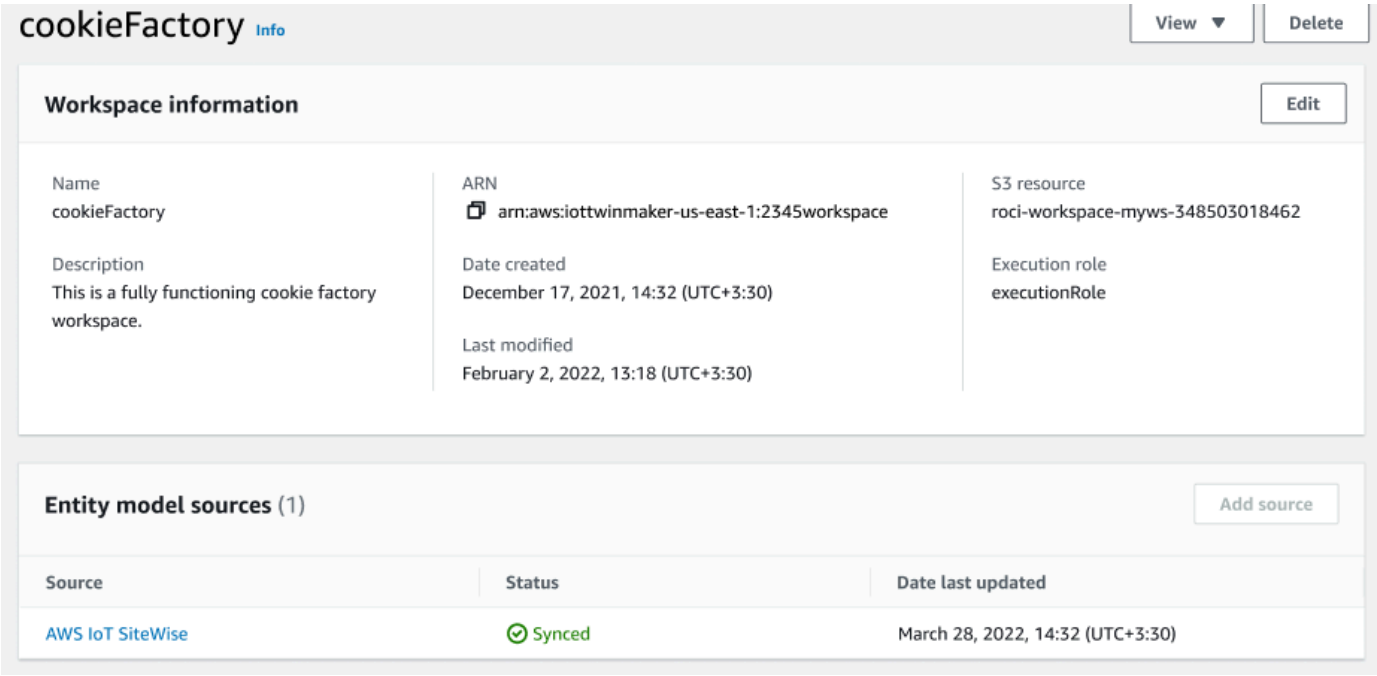
3. Wähle Workspace Connect, um deinen AWS IoT TwinMaker Workspace mit deinen AWS IoT SiteWise Assets zu verknüpfen.

> **Note**
>
> Du kannst die Asset-Synchronisierung nur mit einem einzigen AWS IoT TwinMaker Workspace verwenden. Sie müssen die Synchronisierung von einem Workspace trennen und eine Verbindung zu einem anderen Workspace herstellen, wenn Sie in einem anderen Workspace synchronisieren möchten.

4. Navigiere als Nächstes zu dem Workspace, in dem du die Asset-Synchronisierung verwenden möchtest.
5. Wählen Sie Add sources (Quellen hinzufügen). Dadurch wird die Quellseite Entitätsmodell hinzufügen geöffnet.



6. Vergewissern Sie sich auf der Seite „Quelle für Entitätsmodell hinzufügen“, dass das Quellfeld angezeigt wird AWS IoT SiteWise. Wählen Sie die IAM-Rolle aus, die Sie als Voraussetzung für die IAM-Rolle erstellt haben.
7. Sie haben jetzt die AWS IoT SiteWise Asset-Synchronisierung aktiviert. Oben auf der ausgewählten Workspace-Seite sollte ein Bestätigungsbanner erscheinen, das bestätigt, dass die Asset-Synchronisierung aktiv ist. Außerdem sollte jetzt im Abschnitt Entitätsmodellquellen eine Synchronisierungsquelle aufgeführt sein.



Verwenden des Io TSite WiseDefaultWorkspace

Wenn Sie sich für die AWS IoT SiteWiseAWS IoT TwinMaker Integration entscheiden, `IoTSiteWiseDefaultWorkspace` wird ein Standard-Workspace mit dem Namen erstellt und automatisch synchronisiert mit AWS IoT SiteWise.

Sie können die AWS IoT TwinMaker `CreateWorkspace` API auch verwenden, um einen Workspace mit dem Namen `IoTSiteWiseDefaultWorkspace` zu erstellen.

Voraussetzungen

Stellen Sie vor dem Erstellen sicher`IoTSiteWiseDefaultWorkspace`, dass Sie Folgendes getan haben:

- Erstellen Sie eine AWS IoT TwinMaker serviceverknüpfte Rolle. Weitere Informationen finden Sie unter Verwenden von serviceverknüpften Rollen für AWS IoT TwinMaker.
- Öffnen Sie unter https://console.aws.amazon.com/iam/ die IAM-Konsole.

  Überprüfen Sie die Rolle oder den Benutzer und vergewissern Sie sich, dass er oder sie dazu `iotsitewise:EnableSiteWiseIntegration` berechtigt ist.

  Falls erforderlich, fügen Sie der Rolle oder dem Benutzer die folgenden Berechtigungen hinzu:

  JSON

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

Unterschiede zwischen benutzerdefinierten und Standardarbeitsbereichen

> **Important**
>
> Neue AWS IoT SiteWise Funktionen wie `CompositionModel`, sind nur in `IoTSiteWiseDefaultWorkspace` verfügbar. Wir empfehlen Ihnen, einen Standardarbeitsbereich anstelle eines benutzerdefinierten Arbeitsbereichs zu verwenden.

Bei der `IoTSiteWiseDefaultWorkspace` Verwendung von gibt es einige bemerkenswerte Unterschiede zur Verwendung eines benutzerdefinierten Arbeitsbereichs mit Asset-Synchronisierung.

- Wenn Sie einen Standard-Workspace erstellen, sind der Amazon S3 S3-Standort und die IAM-Rolle optional.

  > **Note**
  >
  > Sie können verwenden`UpdateWorkspace`, um den Amazon S3 S3-Standort und die IAM-Rolle anzugeben.

- `IoTSiteWiseDefaultWorkspace`Es gibt kein Limit für die Ressourcenanzahl, mit der AWS IoT SiteWise Ressourcen synchronisiert werden können AWS IoT TwinMaker.
- Wenn Sie Ressourcen von synchronisieren AWS IoT SiteWise, `SyncSource` werden sie es sein`SITEWISE_MANAGED`. Dies beinhaltet `Entities` und`ComponentTypes`.
- Neue AWS IoT SiteWise Funktionen, wie sie nur in der verfügbar `CompositionModel` sind`IoTSiteWiseDefaultWorkspace`.

Es gibt einige spezifische Einschränkungen für: `IoTSiteWiseDefaultWorkspace`

- Der Standardarbeitsbereich kann nicht gelöscht werden.
- Um Ressourcen zu löschen, müssen Sie zuerst die AWS IoT SiteWise Ressourcen löschen. Anschließend AWS IoT TwinMaker werden die entsprechenden Ressourcen in gelöscht.

Ressourcen, die synchronisiert wurden von AWS IoT SiteWise

In diesem Thema wird aufgeführt, welche Assets du mit deinem AWS IoT TwinMaker Workspace synchronisieren kannst. AWS IoT SiteWise

> **Important**
>
> Informationen Unterschiede zwischen benutzerdefinierten und Standardarbeitsbereichen zu den Unterschieden zwischen dem benutzerdefinierten und dem Standardarbeitsbereich findest du unter.

Benutzerdefinierte und Standardarbeitsbereiche

Die folgenden Ressourcen sind synchronisiert und sowohl in benutzerdefinierten als auch in Standardarbeitsbereichen verfügbar:

Asset-Modelle

AWS IoT TwinMaker erstellt einen neuen Komponententyp für jedes Asset-Modell in AWS IoT SiteWise.

- Die Komponente `TypeId` für das Asset-Modell verwendet eines der folgenden Muster:
  - Benutzerdefinierter Arbeitsbereich - `iotsitewise.assetmodel:`*`assetModelId`*
  - Standardarbeitsbereich - *`assetModelId`*
- Jede Eigenschaft im Asset-Modell ist eine neue Eigenschaft im Komponententyp mit einem der folgenden Benennungsmuster:
  - Benutzerdefinierter Arbeitsbereich - `Property_`*`propertyId`*
  - Standardarbeitsbereich - *`propertyId`*

  Der Eigenschaftsname in AWS IoT SiteWise wird wie `displayName` in der Eigenschaftsdefinition gespeichert.
- Jede Hierarchie im Asset-Modell ist eine neue Eigenschaft vom Typ `LIST` und `nestedType` gehört `RELATIONSHIP` zum Komponententyp. Die Hierarchie wird der Eigenschaft mit einem Namen zugeordnet, dem eines der folgenden Elemente vorangestellt ist:
  - Benutzerdefinierter Arbeitsbereich - *`Hierarchy_hierarchyId`*
  - Standardarbeitsbereich - *`hierarchyId`*

Anlage

AWS IoT TwinMaker erstellt eine neue Entität für jedes Asset in AWS IoT SiteWise.

- Das `entityId` ist dasselbe wie das `assetId` in AWS IoT SiteWise.
- Diese Entitäten haben eine einzige Komponente namens`sitewiseBase`, deren Komponententyp dem Asset-Modell für dieses Asset entspricht.
- Alle Überschreibungen auf Vermögensebene, wie z. B. die Festlegung eines Eigenschaftsalias oder einer Maßeinheit, werden in der Entität in AWS IoT TwinMaker wiedergegeben.

Nur Standard-Workspace

Die folgenden Elemente sind synchronisiert und nur im Standardarbeitsbereich verfügbar,`IoTSiteWiseDefaultWorkspace`.

AssetModelComponents

AWS IoT TwinMaker erstellt für jede `AssetModelComponents` Komponente einen neuen Komponententyp. AWS IoT SiteWise

- Die Komponente `TypeId` für das Asset-Modell verwendet das folgende Muster:`assetModelId`.
- Jede Eigenschaft im Asset-Modell ist eine neue Eigenschaft im Komponententyp mit dem Eigenschaftsnamen als`propertyId`. Der Eigenschaftsname in AWS IoT SiteWise wird wie `displayName` in der Eigenschaftsdefinition gespeichert.
- Jede Hierarchie im Asset-Modell ist eine neue Eigenschaft vom Typ `LIST` und `nestedType` gehört `RELATIONSHIP` zum Komponententyp. Die Hierarchie ist der Eigenschaft zugeordnet, der ein Name vorangestellt ist. `hierarchyId`

AssetModelCompositeModel

AWS IoT TwinMaker erstellt für jede `AssetModelCompositeModel` Komponente einen neuen Komponententyp. AWS IoT SiteWise

- Die Komponente `TypeId` für das Asset-Modell verwendet das folgende Muster:`assetModelId_assetModelCompositeModelId`.
- Jede Eigenschaft im Asset-Modell ist eine neue Eigenschaft im Komponententyp mit dem Eigenschaftsnamen als`propertyId`. Der Eigenschaftsname in AWS IoT SiteWise wird wie `displayName` in der Eigenschaftsdefinition gespeichert.

AssetCompositeModels

AWS IoT TwinMaker erstellt für jeden `AssetCompositeModel` Eingang eine neue Verbundkomponente AWS IoT SiteWise.

- Das `componentName` ist dasselbe wie das `assetModelCompositeModelId` in AWS IoT SiteWise.

Ressourcen wurden nicht synchronisiert

Die folgenden Ressourcen werden nicht synchronisiert:

Nicht synchronisierte Assets und Asset-Modelle

- Alarmmodelle werden als CompositeModels synchronisiert, aber die entsprechenden Daten im Asset, die sich auf Alarme beziehen, werden nicht synchronisiert.
- AWS IoT SiteWise Datenströme werden nicht synchronisiert. Nur Eigenschaften, die im Asset-Modell modelliert wurden, werden synchronisiert.
- Eigenschaftswerte für Attribute, Messungen, Transformationen, Aggregate und die Berechnung von Metadaten wie Formel und Fenster werden nicht synchronisiert. Nur die Metadaten zu den Eigenschaften, wie Alias, Maßeinheit und Datentyp, werden synchronisiert. Die Werte können mit der regulären AWS IoT TwinMaker Datenkonnektor-API, abgefragt werden. GetPropertyValueHistory

Verwenden Sie synchronisierte Entitäten und Komponententypen in AWS IoT TwinMaker

Sobald die Assets synchronisiert wurden AWS IoT SiteWise, können die synchronisierten Komponententypen nur noch eingelesen werden. AWS IoT TwinMaker Jede Aktualisierungs- oder Löschaktion muss in ausgeführt werden AWS IoT SiteWise, und diese Änderungen werden synchronisiert, AWS IoT TwinMaker sofern der SyncJob noch aktiv ist.

Die synchronisierten Entitäten und die AWS IoT SiteWise Basiskomponente sind ebenfalls schreibgeschützt. AWS IoT TwinMaker Sie können der synchronisierten Entität weitere nicht synchronisierte Komponenten hinzufügen, sofern keine Attribute auf Entitätsebene wie die Beschreibung oder aktualisiert werden. `entityName`

Für die Interaktion mit synchronisierten Entitäten gelten einige Einschränkungen. Sie können keine untergeordneten Entitäten unter einer synchronisierten Entität in der Hierarchie der synchronisierten

Entität erstellen. Darüber hinaus können Sie keine nicht synchronisierten Komponententypen erstellen, die von einem synchronisierten Komponententyp ausgehen.

> **Note**
>
> Zusätzliche Komponenten werden zusammen mit der Entität gelöscht, wenn das Asset in gelöscht wird AWS IoT SiteWise oder wenn Sie den Synchronisierungsjob löschen.

Sie können diese synchronisierten Entitäten in Grafana-Dashboards verwenden und sie wie normale Entitäten als Tags im Scene Composer hinzufügen. Sie können auch Knowledge-Graph-Abfragen für diese synchronisierten Entitäten stellen.

> **Note**
>
> Synchronisierte Entitäten ohne Änderung werden nicht in Rechnung gestellt. Ihnen werden jedoch Gebühren für diese Entitäten in Rechnung gestellt, wenn Änderungen vorgenommen wurden. AWS IoT TwinMaker Wenn Sie beispielsweise einer synchronisierten Entität eine nicht synchronisierte Komponente hinzufügen, wird diese Entität jetzt in Rechnung gestellt. AWS IoT TwinMaker Weitere Informationen finden Sie unter AWS IoT TwinMaker – Preise.

Analysieren Sie den Synchronisierungsstatus und die Fehler

Dieses Thema enthält Anleitungen zur Analyse von Synchronisierungsfehlern und -status.

> **Important**
>
> Informationen zu the section called “Unterschiede zwischen benutzerdefinierten und Standardarbeitsbereichen” den Unterschieden zwischen dem benutzerdefinierten und dem Standardarbeitsbereich finden Sie unter.

Synchronisieren Sie den Jobstatus

Ein Synchronisierungsauftrag hat je nach Status einen der folgenden Status.

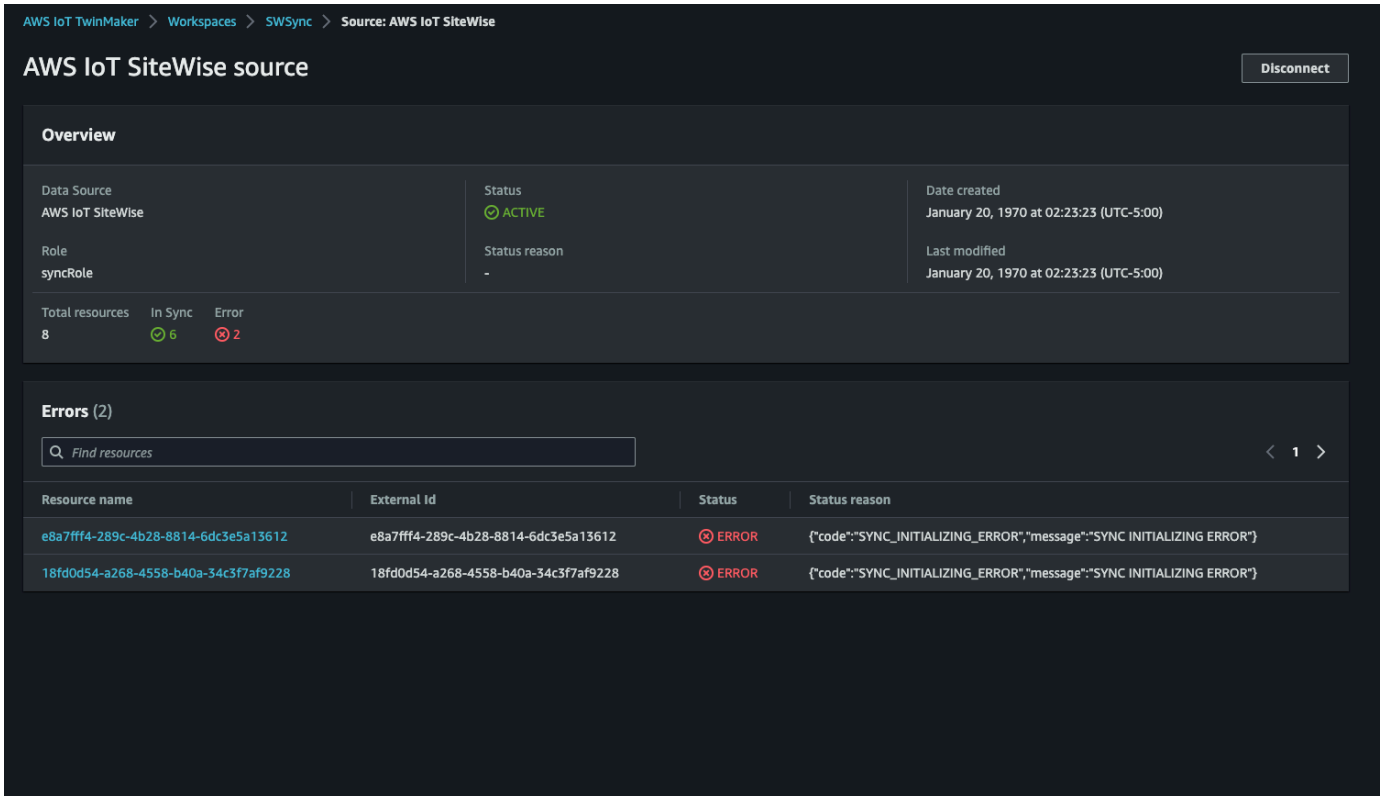
- Der `CREATING` Status des Synchronisierungsauftrags bedeutet, dass der Job nach Berechtigungen sucht und Daten von lädt AWS IoT SiteWise , um die Synchronisierung vorzubereiten.

- Der `INITIALIZING` Status des Synchronisierungsauftrags bedeutet, dass alle vorhandenen Ressourcen synchronisiert wurden. AWS IoT SiteWise AWS IoT TwinMaker Dieser Schritt kann länger dauern, wenn der Benutzer über eine große Anzahl von Assets und Asset-Modellen verfügt. AWS IoT SiteWise Sie können die Anzahl der synchronisierten Ressourcen überwachen, indem Sie den Synchronisierungsauftrag in der AWS IoT TwinMaker Konsole aktivieren oder die `ListSyncResources` API aufrufen.
- Der `ACTIVE` Status des Synchronisierungsauftrags bedeutet, dass der Initialisierungsschritt abgeschlossen ist. Der Job ist jetzt bereit, alle neuen Updates von AWS IoT SiteWise zu synchronisieren.
- Der `ERROR` Status des Synchronisierungsauftrags weist auf einen Fehler bei einem der vorherigen Status hin. Überprüfen Sie die Fehlermeldung. Möglicherweise liegt ein Problem mit der Einrichtung der IAM-Rolle vor. Wenn Sie eine neue IAM-Rolle verwenden möchten, löschen Sie den Synchronisierungsjob, bei dem der Fehler aufgetreten ist, und erstellen Sie einen neuen mit der neuen Rolle.

Synchronisierungsfehler werden auf der Modellquellenseite angezeigt, auf die Sie über die Tabelle Entitätsmodellquellen in Ihrem Workspace zugreifen können. Auf der Seite mit der Modellquelle wird eine Liste der Ressourcen angezeigt, die nicht synchronisiert werden konnten. Die meisten Fehler werden durch den Synchronisierungsauftrag automatisch wiederholt. Wenn für die Ressource jedoch eine Aktion erforderlich ist, bleibt sie im `ERROR` Status. Sie können eine Liste der Fehler auch mithilfe der ListSyncResourcesAPI abrufen.

Gehen Sie wie folgt vor, um alle aufgelisteten Fehler für die aktuelle Quelle anzuzeigen.

1. Navigieren Sie in der AWS IoT TwinMaker Konsole zu Ihrem Workspace.
2. Wählen Sie die AWS IoT SiteWise Quelle aus, die im Modal Entity Model Sources aufgeführt ist, um die Seite mit den Details zur Asset-Synchronisierung zu öffnen.



3. Wie im vorherigen Screenshot gezeigt, werden alle Ressourcen mit anhaltenden Fehlern in der Tabelle Fehler aufgeführt. Sie können diese Tabelle verwenden, um Fehler im Zusammenhang mit bestimmten Ressourcen aufzuspüren und zu beheben.

Zu den möglichen Fehlern gehören die folgenden:

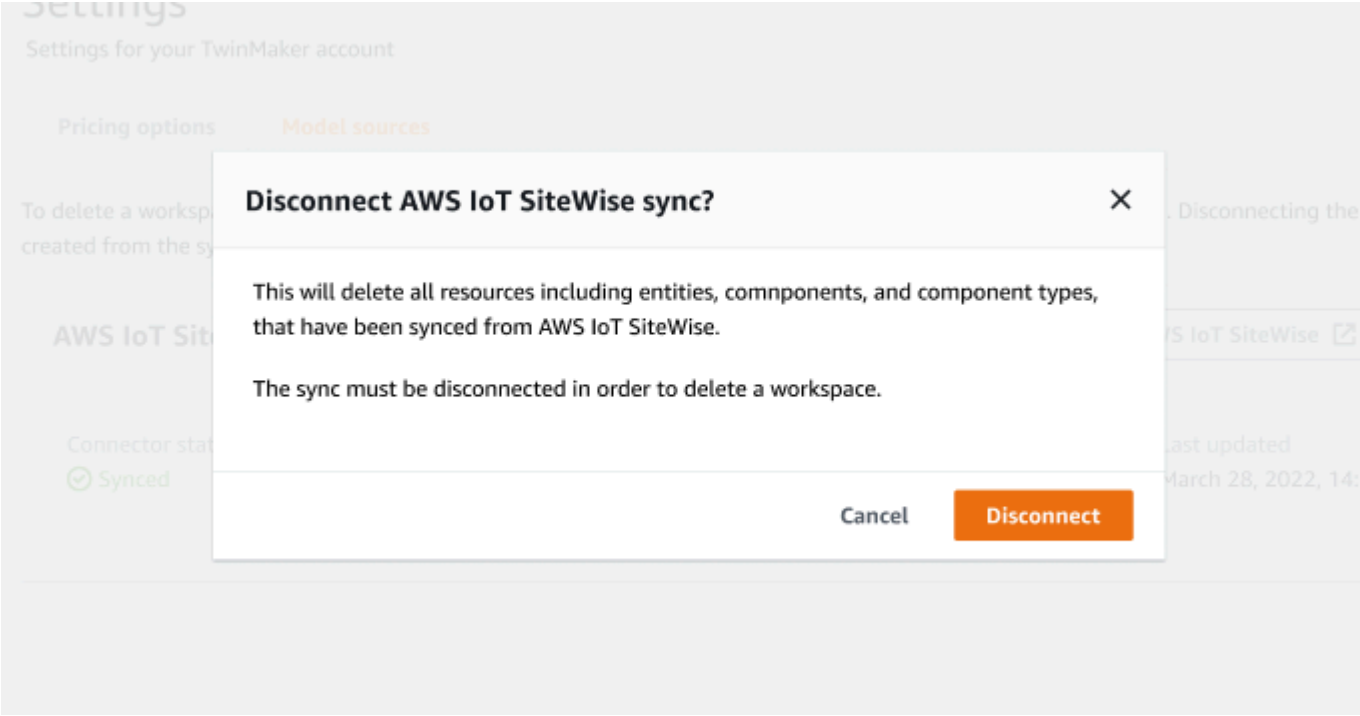
- AWS IoT SiteWise Unterstützt zwar doppelte Asset-Namen, aber AWS IoT TwinMaker nur auf der R00T Ebene, nicht unter derselben übergeordneten Entität. Wenn Sie zwei Assets mit demselben Namen unter einer übergeordneten Entität haben AWS IoT SiteWise, kann eines davon nicht synchronisiert werden. Um diesen Fehler zu beheben, löschen Sie entweder eines der Assets oder verschieben Sie eines unter ein anderes übergeordnetes Asset, AWS IoT SiteWise bevor Sie die Synchronisierung durchführen.
- Wenn Sie bereits über eine Entität mit derselben ID wie die AWS IoT SiteWise Asset-ID verfügen, kann dieses Asset erst synchronisiert werden, wenn Sie die vorhandene Entität löschen.

Löschen Sie einen Synchronisierungsauftrag

Gehen Sie wie folgt vor, um einen Synchronisierungsauftrag zu löschen.

> **Important**
>
> Informationen the section called “Unterschiede zwischen benutzerdefinierten und Standardarbeitsbereichen” zu den Unterschieden zwischen dem benutzerdefinierten und dem Standardarbeitsbereich finden Sie unter.

1. Navigieren Sie zur AWS IoT TwinMaker -Konsole.
2. Öffnen Sie den Arbeitsbereich, aus dem Sie den Synchronisierungsjob löschen möchten.
3. Wählen Sie unter Entitätsmodellquellen die AWS IoT SiteWise Quelle aus, um die Seite mit den Quelldetails zu öffnen.
4. Um den Synchronisierungsjob zu beenden, wählen Sie Disconnect. Bestätigen Sie Ihre Auswahl, um den Synchronisierungsauftrag vollständig zu löschen.



Sobald ein Synchronisierungsauftrag gelöscht wurde, können Sie den Synchronisierungsauftrag in demselben oder einem anderen Arbeitsbereich erneut erstellen.

Du kannst einen Workspace nicht löschen, wenn es Synchronisierungsjobs in diesem Workspace gibt. Lösche zuerst die Synchronisierungsaufträge, bevor du einen Workspace löschst.

Wenn beim Löschen des Synchronisierungsauftrags Fehler auftreten, verbleibt der Synchronisierungsauftrag im `DELETING` Status und wird automatisch erneut versucht. Sie können jetzt alle synchronisierten Entitäten oder Komponententypen manuell löschen, wenn beim Löschen einer Ressource ein Fehler auftritt.

> **Note**
>
> Alle Ressourcen, von denen aus synchronisiert wurden, AWS IoT SiteWise werden zuerst gelöscht, dann wird der Synchronisierungsauftrag selbst gelöscht.

Beschränkungen der Asset-Synchronisierung

> **Important**
>
> Informationen the section called “Unterschiede zwischen benutzerdefinierten und Standardarbeitsbereichen” zu den Unterschieden zwischen dem benutzerdefinierten und dem Standardarbeitsbereich finden Sie unter.

Da die AWS IoT SiteWise Kontingente höher sind als die AWS IoT TwinMaker Standardkontingente, erhöhen wir die folgenden Grenzwerte für Entitäten und Komponententypen, von denen aus synchronisiert wird. AWS IoT SiteWise

- 1000 synchronisierte Komponententypen in einem Workspace, da nur 1000 Objektmodelle aus synchronisiert werden können. AWS IoT SiteWise
- 100.000 synchronisierte Entitäten in einem Workspace, da nur 100.000 Assets synchronisiert werden können. AWS IoT SiteWise
- Maximal 2000 untergeordnete Entitäten pro übergeordneter Entität. Pro Anlage eines alleinerziehenden Elternteils werden 2000 Vermögenswerte von Kindern synchronisiert.

> **Note**
>
> Die GetEntityAPI gibt nur die ersten 50 untergeordneten Entitäten für eine Hierarchieeigenschaft zurück, aber Sie können die GetPropertyValueAPI verwenden, um die Liste aller untergeordneten Entitäten zu paginieren und abzurufen.

- 600 Eigenschaften pro synchronisierter Komponente von AWS IoT SiteWise, wodurch Asset-Modelle mit insgesamt 600 Eigenschaften und Hierarchien synchronisiert werden können.

> **Note**
>
> Diese Grenzwerte gelten nur für die synchronisierten Entitäten. Fordern Sie eine Erhöhung des Kontingents an, wenn Sie möchten, dass diese Grenzwerte für nicht synchronisierte Ressourcen erhöht werden.

AWS IoT TwinMaker Grafana-Dashboard-Integration

AWS IoT TwinMaker unterstützt die Grafana-Integration über ein Anwendungs-Plugin. Verwenden Sie Grafana-Version 10.4.0 und spätere Versionen, um mit Ihrer digitalen Zwillingsanwendung zu interagieren. Das AWS IoT TwinMaker Plugin bietet benutzerdefinierte Panels, Dashboard-Vorlagen und eine Datenquelle, um eine Verbindung zu Ihren digitalen Zwillingsdaten herzustellen.

Weitere Informationen zum Onboarding bei Grafana und zum Einrichten von Berechtigungen für Ihr Dashboard finden Sie in den folgenden Themen:

Themen

- CORS-Konfiguration für Grafana Scene Viewer
- Einrichtung Ihrer Grafana-Umgebung
- Eine Dashboard-IAM-Rolle erstellen
- Eine AWS IoT TwinMaker Videoplayer-Richtlinie erstellen

> **Note**
>
> Sie müssen die CORS-Konfiguration (Cross-Origin Resource Sharing) des Amazon S3 S3-Buckets ändern, damit die Grafana-Benutzeroberfläche Ressourcen aus dem Bucket laden kann. Die Anweisungen finden Sie unter CORS-Konfiguration für Grafana Scene Viewer.

Weitere Informationen zum AWS IoT TwinMaker Grafana-Plugin finden Sie in der AWS IoT TwinMaker App-Dokumentation.

Weitere Informationen zu den wichtigsten Komponenten des Grafana-Plugins finden Sie im Folgenden:

- AWS IoT TwinMaker Datenquelle
- Dashboard-Vorlagen
- Scene Viewer-Bedienfeld
- Bedienfeld „Video Player“

CORS-Konfiguration für Grafana Scene Viewer

Das AWS IoT TwinMaker Grafana-Plugin erfordert eine CORS-Konfiguration (Cross-Origin Resource Sharing), die es der Grafana-Benutzeroberfläche ermöglicht, Ressourcen aus dem Amazon S3 S3-Bucket zu laden. Ohne die CORS-Konfiguration erhalten Sie im Scene Viewer die Fehlermeldung „Load 3D Scene failed with Network Failure“, da die Grafana-Domäne nicht auf die Ressourcen im Amazon S3 S3-Bucket zugreifen kann.

Gehen Sie wie folgt vor, um Ihren Amazon S3 S3-Bucket mit CORS zu konfigurieren:

1. Melden Sie sich bei der IAM-Konsole an und öffnen Sie die Amazon S3 S3-Konsole.
2. Wählen Sie in der Bucket-Liste den Namen des Buckets aus, den Sie als Ressourcen-Bucket Ihres AWS IoT TwinMaker Workspace verwenden.
3. Wählen Sie Permissions (Berechtigungen).
4. Wähle im Abschnitt Quellenübergreifende gemeinsame Nutzung von Ressourcen die Option Bearbeiten aus, um den CORS-Editor zu öffnen.
5. Geben Sie im Textfeld des CORS-Konfigurationseditors die folgende JSON-CORS-Konfiguration ein oder kopieren Sie sie und fügen Sie sie ein, indem Sie die Grafana-Workspace-Domain durch Ihre Domain *GRAFANA-WORKSPACE-DOMAIN* ersetzen.

> **Note**
>
> Sie müssen das Sternchen am Anfang des * JSON-Elements beibehalten. `"AllowedOrigins":`

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

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

6. Wählen Sie Änderungen speichern aus, um die CORS-Konfiguration abzuschließen.

Weitere Informationen zu CORS mit Amazon S3 S3-Buckets finden Sie unter Verwenden von Cross-Origin Resource Sharing (CORS).

Einrichtung Ihrer Grafana-Umgebung

Sie können Amazon Managed Grafana für einen vollständig verwalteten Service verwenden oder eine Grafana-Umgebung einrichten, die Sie selbst verwalten. Mit Amazon Managed Grafana können Sie Open-Source-Grafana schnell für Ihre Bedürfnisse bereitstellen, betreiben und skalieren. Alternativ können Sie Ihre eigene Infrastruktur zur Verwaltung von Grafana-Servern einrichten.

Weitere Informationen zu beiden Grafana-Umgebungsoptionen finden Sie in den folgenden Themen:

- Amazon Managed Grafana
- Selbstverwaltetes Grafana

Amazon Managed Grafana

Amazon Managed Grafana bietet ein AWS IoT TwinMaker Plugin, AWS IoT TwinMaker mit dem Sie sich schnell in Grafana integrieren können. Da Amazon Managed Grafana Grafana-Server für Sie verwaltet, können Sie Ihre Daten visualisieren, ohne Hardware oder andere Grafana-Infrastruktur erstellen, verpacken oder bereitstellen zu müssen. Weitere Informationen zu Amazon Managed Grafana finden Sie unter Was ist Amazon Managed Grafana? .

> **Note**
>
> Amazon Managed Grafana unterstützt derzeit Version 1.3.1 des AWS IoT TwinMaker Grafana-Plug-ins.

Voraussetzungen für Amazon Managed Grafana

Für die Verwendung AWS IoT TwinMaker in einem Amazon Managed Grafana-Dashboard müssen Sie zunächst die folgenden Voraussetzungen erfüllen:

- Erstellen Sie einen AWS IoT TwinMaker Workspace. Weitere Informationen zum Erstellen von Workspaces finden Sie unter Erste Schritte mit AWS IoT TwinMaker.

> **Note**
>
> Wenn Sie zum ersten Mal einen Amazon Managed Grafana-Workspace in der AWS Management Console erstellen, AWS IoT TwinMaker ist nicht aufgeführt. Das Plugin ist jedoch bereits auf allen Workspaces installiert. Sie finden das AWS IoT TwinMaker Plugin in der Open-Source-Grafana-Plugin-Liste. Sie finden die AWS IoT TwinMaker Datenquelle, indem Sie auf der Seite Datenquellen die Option Datenquelle hinzufügen auswählen.

Wenn Sie einen Amazon Managed Grafana-Workspace erstellen, wird automatisch eine IAM-Rolle erstellt, um die Berechtigungen für die Grafana-Instance zu verwalten. Dies wird als Workspace-IAM-Rolle bezeichnet. Dies ist die Option des Authentifizierungsanbieters, mit der Sie alle AWS IoT TwinMaker Datenquellen für Grafana konfigurieren. Amazon Managed Grafana unterstützt das automatische Hinzufügen von Berechtigungen für nicht AWS IoT TwinMaker, daher müssen Sie diese Berechtigungen manuell einrichten. Weitere Informationen zum Einrichten manueller Berechtigungen finden Sie unterEine Dashboard-IAM-Rolle erstellen.

Selbstverwaltetes Grafana

Sie können wählen, ob Sie Ihre eigene Infrastruktur für den Betrieb von Grafana hosten möchten. Informationen zum lokalen Ausführen von Grafana auf Ihrem Computer finden Sie unter Grafana installieren. Das AWS IoT TwinMaker Plugin ist im öffentlichen Grafana-Katalog verfügbar. Informationen zur Installation dieses Plugins in Ihrer Grafana-Umgebung finden Sie unter AWS IoT TwinMaker App.

Wenn Sie Grafana lokal ausführen, können Sie Dashboards nicht einfach teilen oder mehreren Benutzern Zugriff gewähren. Eine skriptbasierte Schnellstartanleitung zum Teilen von Dashboards mit lokalem Grafana finden AWS IoT TwinMaker Sie unter Beispiel-Repository. Diese Ressource führt Sie durch das Hosten einer Grafana-Umgebung auf Cloud9 und Amazon EC2 auf einem öffentlichen Endpunkt.

Sie müssen festlegen, welchen Authentifizierungsanbieter Sie für die Konfiguration von TwinMaker Datenquellen verwenden möchten. Sie konfigurieren die Anmeldeinformationen für die Umgebung auf der Grundlage der standardmäßigen Anmeldeinformationenkette (siehe Verwenden der standardmäßigen Anmeldeinformationsanbieterkette). Die Standardanmeldedaten können die permanenten Anmeldeinformationen eines beliebigen Benutzers oder jeder Rolle sein. Wenn Sie beispielsweise Grafana auf Amazon ausführen EC2 , hat die Standard-Anmeldeinformationskette Zugriff auf die EC2 Amazon-Ausführungsrolle, die dann Ihr Authentifizierungsanbieter wäre. Der IAM Amazon Resource Name (ARN) des Authentifizierungsanbieters ist in den Schritten bis Eine Dashboard-IAM-Rolle erstellen erforderlich.

Eine Dashboard-IAM-Rolle erstellen

Mit AWS IoT TwinMaker können Sie den Datenzugriff auf Ihren Grafana-Dashboards steuern. Grafana-Dashboard-Benutzer sollten unterschiedliche Berechtigungsbereiche haben, um Daten anzuzeigen und in einigen Fällen Daten zu schreiben. Beispielsweise hat ein Alarmanbieter möglicherweise keine Erlaubnis, Videos anzusehen, während ein Administrator die Berechtigung für alle Ressourcen hat. Grafana definiert die Berechtigungen über Datenquellen, in denen Anmeldeinformationen und eine IAM-Rolle bereitgestellt werden. Die AWS IoT TwinMaker Datenquelle ruft AWS Anmeldeinformationen mit Berechtigungen für diese Rolle ab. Wenn keine IAM-Rolle bereitgestellt wird, verwendet Grafana den Umfang der Anmeldeinformationen, der nicht reduziert werden kann. AWS IoT TwinMaker

Um Ihre AWS IoT TwinMaker Dashboards in Grafana zu verwenden, erstellen Sie eine IAM-Rolle und fügen Richtlinien hinzu. Sie können die folgenden Vorlagen verwenden, um diese Richtlinien zu erstellen.

Eine IAM-Richtlinie erstellen

Erstellen Sie eine IAM-Richtlinie, die *YourWorkspaceId*`DashboardPolicy` in der IAM-Konsole aufgerufen wird. Diese Richtlinie gewährt Ihren Workspaces Zugriff auf den Amazon S3 S3-Bucket und die AWS IoT TwinMaker Ressourcen. Sie können sich auch dafür entscheiden,

AWS IoT Greengrass Edge Connector für Amazon Kinesis Video Streams zu verwenden, wofür Berechtigungen für die Kinesis Video Streams und für die Komponente konfigurierten AWS IoT SiteWise Assets erforderlich sind. Wählen Sie je nach Anwendungsfall eine der folgenden Richtlinienvorlagen aus.

1. Keine Richtlinie für Videoberechtigungen

   Wenn Sie das Grafana Video Player-Bedienfeld nicht verwenden möchten, erstellen Sie die Richtlinie mithilfe der folgenden Vorlage.

   JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject"
      ],
      "Resource": [
        "arn:aws:s3:::bucketName/*",
        "arn:aws:s3:::bucketName"
        ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "iottwinmaker:Get*",
        "iottwinmaker:List*"
      ],
      "Resource": [
        "arn:aws:iottwinmaker:region:accountId:workspace/workspaceId",
        "arn:aws:iottwinmaker:region:accountId:workspace/workspaceId/*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": "iottwinmaker:ListWorkspaces",
      "Resource": "*"
    }
  ]
}
```

Für jeden Workspace wird ein Amazon S3 S3-Bucket erstellt. Er enthält die 3D-Modelle und -Szenen, die auf einem Dashboard angezeigt werden können. Das SceneViewerPanel lädt Elemente aus diesem Bucket.

2. Eingeschränkte Richtlinie für Videoberechtigungen

   Um den Zugriff auf das Video Player-Bedienfeld in Grafana einzuschränken, gruppieren Sie Ihre AWS IoT Greengrass Edge Connector für Amazon Kinesis Video Streams Streams-Ressourcen nach Tags. Weitere Informationen zur Festlegung des Gültigkeitsbereichs für Ihre Videoressourcen finden Sie unter. Eine AWS IoT TwinMaker Videoplayer-Richtlinie erstellen

3. Alle Videoberechtigungen

   Wenn Sie Ihre Videos nicht gruppieren möchten, können Sie sie alle über den Grafana Video Player zugänglich machen. Jeder, der Zugriff auf einen Grafana-Workspace hat, kann Videos für jeden Stream in Ihrem Konto abspielen und hat nur Lesezugriff auf jedes AWS IoT SiteWise Asset. Dies schließt alle Ressourcen ein, die in der future geschaffen werden.

   Erstellen Sie die Richtlinie mit der folgenden Vorlage:

   JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject"
      ],
      "Resource": [
        "arn:aws:s3:::bucketName/*",
        "arn:aws:s3:::bucketName"
        ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "iottwinmaker:Get*",
        "iottwinmaker:List*"
      ],
      "Resource": [
        "arn:aws:iottwinmaker:region:accountId:workspace/workspaceId",
        "arn:aws:iottwinmaker:region:accountId:workspace/workspaceId/*"
```

```
      ]
    },
    {
      "Effect": "Allow",
      "Action": "iottwinmaker:ListWorkspaces",
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "kinesisvideo:GetDataEndpoint",
        "kinesisvideo:GetHLSStreamingSessionURL"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "iotsitewise:GetAssetPropertyValue",
        "iotsitewise:GetInterpolatedAssetPropertyValues"
      ],
      "Resource": "*"
    },
    {
       "Effect": "Allow",
       "Action": [
        "iotsitewise:BatchPutAssetPropertyValue"
      ],
      "Resource": "*",
      "Condition": {
        "StringLike": {
          "aws:ResourceTag/EdgeConnectorForKVS": "*workspaceId*"
        }
      }
    }
  ]
}
```

Diese Richtlinienvorlage bietet die folgenden Berechtigungen:

- Schreibgeschützter Zugriff auf einen S3-Bucket zum Laden einer Szene.
- Schreibgeschützter Zugriff auf AWS IoT TwinMaker für alle Entitäten und Komponenten in einem Workspace.

- Schreibgeschützter Zugriff, um alle Kinesis Video Streams Streams-Videos in Ihrem Konto zu streamen.
- Schreibgeschützter Zugriff auf die Immobilienwerthistorie aller AWS IoT SiteWise Vermögenswerte in Ihrem Konto.
- Erfassung von Daten in beliebige Immobilien eines AWS IoT SiteWise Vermögenswerts, die mit dem Schlüssel `EdgeConnectorForKVS` und dem Wert gekennzeichnet sind. `workspaceId`

Markieren Sie Ihr AWS IoT SiteWise Kamera-Asset und fordern Sie den Video-Upload von Edge an

Mit dem Video Player in Grafana können Benutzer manuell anfordern, dass Videos aus dem Edge-Cache in Kinesis Video Streams hochgeladen werden. Sie können diese Funktion für jedes AWS IoT SiteWise Asset aktivieren, das mit Ihrem AWS IoT Greengrass Edge Connector für Amazon Kinesis Video Streams verknüpft und mit dem Schlüssel `EdgeConnectorForKVS` gekennzeichnet ist.

Der Tag-Wert kann eine Liste von WorkspaceIDs sein, die durch eines der folgenden Zeichen getrennt sind:. `. : + = @ _ / -` Wenn Sie beispielsweise ein AWS IoT SiteWise Asset, das einem AWS IoT Greengrass Edge Connector für Amazon Kinesis Video Streams zugeordnet ist, in allen AWS IoT TwinMaker Arbeitsbereichen verwenden möchten, können Sie ein Tag verwenden, das diesem Muster folgt:. `WorkspaceA/WorkspaceB/WorkspaceC` Das Grafana-Plugin erzwingt, dass die AWS IoT TwinMaker WorkspaceID verwendet wird, um die Erfassung von Asset-Daten zu gruppieren AWS IoT SiteWise .

Fügen Sie Ihrer Dashboard-Richtlinie weitere Berechtigungen hinzu

Das AWS IoT TwinMaker Grafana-Plugin verwendet Ihren Authentifizierungsanbieter, um die von Ihnen erstellte Dashboard-Rolle aufzurufen AssumeRole . Intern schränkt das Plugin den höchsten Umfang der Berechtigungen ein, auf die Sie Zugriff haben, indem es beim Anruf eine Sitzungsrichtlinie verwendet. AssumeRole Weitere Informationen zu Sitzungsrichtlinien finden Sie unter Sitzungsrichtlinien.

Dies ist die maximal zulässige Richtlinie, die du in deiner Dashboard-Rolle für einen AWS IoT TwinMaker Workspace haben kannst:

JSON

```
{
```

```
"Version": "2012-10-17",
"Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "s3:GetObject"
    ],
    "Resource": [
      "arn:aws:s3:::bucketName/*",
      "arn:aws:s3:::bucketName"
      ]
  },
  {
    "Effect": "Allow",
    "Action": [
      "iottwinmaker:Get*",
      "iottwinmaker:List*"
    ],
    "Resource": [
      "arn:aws:iottwinmaker:region:accountId:workspace/workspaceId",
      "arn:aws:iottwinmaker:region:accountId:workspace/workspaceId/*"
    ]
  },
  {
    "Effect": "Allow",
    "Action": "iottwinmaker:ListWorkspaces",
    "Resource": "*"
  },
  {
    "Effect": "Allow",
    "Action": [
      "kinesisvideo:GetDataEndpoint",
      "kinesisvideo:GetHLSStreamingSessionURL"
    ],
    "Resource": "*"
  },
  {
    "Effect": "Allow",
    "Action": [
      "iotsitewise:GetAssetPropertyValue",
      "iotsitewise:GetInterpolatedAssetPropertyValues"
    ],
    "Resource": "*"
  },
```

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

Wenn du Aussagen hinzufügst, die `Allow` mehr Berechtigungen enthalten, funktionieren sie im AWS IoT TwinMaker Plugin nicht. Dies ist beabsichtigt, um sicherzustellen, dass das Plugin nur die minimal erforderlichen Berechtigungen verwendet.

Sie können die Berechtigungen jedoch weiter einschränken. Weitere Informationen finden Sie unter Eine AWS IoT TwinMaker Videoplayer-Richtlinie erstellen.

Die Grafana-Dashboard-IAM-Rolle erstellen

Erstellen Sie in der IAM-Konsole eine IAM-Rolle namens. *YourWorkspaceId*`DashboardRole` Hängen Sie das an die *YourWorkspaceId*`DashboardPolicy` Rolle an.

Um die Vertrauensrichtlinie der Dashboard-Rolle zu bearbeiten, müssen Sie dem Grafana-Authentifizierungsanbieter die Erlaubnis erteilen, die Dashboard-Rolle `AssumeRole` aufzurufen. Aktualisieren Sie die Vertrauensrichtlinie mit der folgenden Vorlage:

JSON

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

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

Weitere Informationen zum Erstellen einer Grafana-Umgebung und zum Finden Ihres Authentifizierungsanbieters finden Sie unterEinrichtung Ihrer Grafana-Umgebung.

Eine AWS IoT TwinMaker Videoplayer-Richtlinie erstellen

Im Folgenden finden Sie eine Richtlinienvorlage mit allen Videoberechtigungen, die Sie für das AWS IoT TwinMaker Plugin in Grafana benötigen:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject"
      ],
      "Resource": [
        "arn:aws:s3:::bucketName/*",
        "arn:aws:s3:::bucketName"
        ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "iottwinmaker:Get*",
        "iottwinmaker:List*"
      ],
      "Resource": [
        "arn:aws:iottwinmaker:region:accountId:workspace/workspaceId",
        "arn:aws:iottwinmaker:region:accountId:workspace/workspaceId/*"
      ]
    },
    {
```

```
      "Effect": "Allow",
      "Action": "iottwinmaker:ListWorkspaces",
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "kinesisvideo:GetDataEndpoint",
        "kinesisvideo:GetHLSStreamingSessionURL"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "iotsitewise:GetAssetPropertyValue",
        "iotsitewise:GetInterpolatedAssetPropertyValues"
      ],
      "Resource": "*"
    },
    {
       "Effect": "Allow",
       "Action": [
        "iotsitewise:BatchPutAssetPropertyValue"
      ],
      "Resource": "*",
      "Condition": {
        "StringLike": {
          "aws:ResourceTag/EdgeConnectorForKVS": "*workspaceId*"
        }
      }
    }
  ]
}
```

Weitere Informationen zur vollständigen Richtlinie finden Sie in der Vorlage für die Richtlinie „Alle Videoberechtigungen“ im Eine IAM-Richtlinie erstellen Thema.

Beschränken Sie den Zugriff auf Ihre Ressourcen

Das Video Player-Bedienfeld in Grafana ruft Kinesis Video Streams und IoT direkt auf, SiteWise um ein vollständiges Videowiedergabeerlebnis zu bieten. Um unbefugten Zugriff auf Ressourcen zu

verhindern, die nicht mit deinem AWS IoT TwinMaker Workspace verknüpft sind, füge Bedingungen zur IAM-Richtlinie für deine Workspace-Dashboard-Rolle hinzu.

Schränke GET-Berechtigungen nach unten

Sie können den Zugriff auf Ihre Amazon Kinesis Video Streams und AWS IoT SiteWise Assets einschränken, indem Sie Ressourcen taggen. Möglicherweise haben Sie Ihr AWS IoT SiteWise Kameraobjekt bereits anhand der AWS IoT TwinMaker WorkspaceID markiert, um die Funktion zur Anforderung von Video-Uploads zu aktivieren. Weitere Informationen finden Sie im Thema Video vom Edge hochladen. Sie können dasselbe Tag-Schlüssel-Wert-Paar verwenden, um den GET-Zugriff auf AWS IoT SiteWise Assets zu beschränken und Ihre Kinesis Video Streams auf die gleiche Weise zu taggen.

Sie können diese Bedingung dann zu den Anweisungen kinesisvideo und iotsitewise im folgenden Verzeichnis hinzufügen: *YourWorkspaceId*`DashboardPolicy`

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

Anwendungsfall aus der Praxis: Gruppieren von Kameras

In diesem Szenario haben Sie eine große Anzahl von Kameras, die den Prozess des Backens von Keksen in einer Fabrik überwachen. Plätzchenteig werden im Teigraum gebacken, Teig wird im Tiefkühlraum eingefroren und Kekse werden im Backraum gebacken. In jedem dieser Räume befinden sich Kameras, wobei verschiedene Bedienerteams jeden Prozess separat überwachen. Sie möchten, dass jede Gruppe von Operatoren für ihren jeweiligen Raum autorisiert ist. Beim Aufbau eines digitalen Zwillings für die Cookie-Fabrik wird ein einziger Arbeitsbereich verwendet, aber die Kameraberechtigungen müssen je nach Raum aufgeteilt werden.

Sie können diese Trennung von Rechten erreichen, indem Sie Kameragruppen anhand ihrer GroupingID taggen. In diesem Szenario lauten die GroupingIDs, und. BatterRoom FreezerRoom BakingRoom Die Kamera in jedem Raum ist mit Kinesis Video Streams verbunden und sollte über ein Tag verfügen mit: Key =`EdgeConnectorForKVS`, Value =`BatterRoom`. Der Wert kann eine Liste von Gruppierungen sein, die durch eines der folgenden Zeichen getrennt sind: `. : + = @ _ / -`

Verwenden Sie die folgenden *YourWorkspaceId*`DashboardPolicy` Richtlinienerklärungen, um das zu ändern:

```
...,
{
  "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*"
    }
  }
},
...
```

Diese Aussagen beschränken die Wiedergabe von Streaming-Videos und den Zugriff auf den AWS IoT SiteWise Eigenschaftsverlauf auf bestimmte Ressourcen in einer Gruppierung. Das *groupingId* wird durch Ihren Anwendungsfall definiert. In unserem Szenario wäre es die RoomID.

Erlaubnisbereich nach unten AWS IoT SiteWise BatchPutAssetPropertyValue

Wenn Sie diese Berechtigung erteilen, wird die Funktion zur Anforderung von Video-Uploads im Video Player aktiviert. Wenn Sie ein Video hochladen, können Sie einen Zeitraum angeben und die Anfrage einreichen, indem Sie im Bereich des Grafana-Dashboards auf Senden klicken.

Verwenden Sie die Standardrichtlinie, um iotsitewise: BatchPutAssetPropertyValue -Berechtigungen zu erteilen:

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

Mithilfe dieser Richtlinie können Benutzer jede beliebige Eigenschaft des BatchPutAssetPropertyValue AWS IoT SiteWise Kameraobjekts aufrufen. Sie können die Autorisierung für eine bestimmte propertyId einschränken, indem Sie sie in der Bedingung der Anweisung angeben.

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

Das Video Player-Bedienfeld in Grafana nimmt Daten in die Messeigenschaft mit dem Namen auf VideoUploadRequest, um das Hochladen von Videos aus dem Edge-Cache in Kinesis Video Streams zu initiieren. Suchen Sie die propertyId dieser Eigenschaft in der AWS IoT SiteWise Konsole. Verwenden Sie die folgende Datenschutzerklärung*YourWorkspaceId*`DashboardPolicy`, um das zu ändern:

```
...,
{
```

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

Diese Erklärung beschränkt das Aufnehmen von Daten auf eine bestimmte Eigenschaft Ihres markierten AWS IoT SiteWise Kameraobjekts. Weitere Informationen finden Sie unter So AWS IoT SiteWise funktioniert es mit IAM.

AWS IoT SiteWise Alarme mit AWS IoT TwinMaker Grafana-Dashboards verbinden

> **Note**
>
> Diese Funktion befindet sich in der öffentlichen Vorschauversion und kann sich ändern.

AWS IoT TwinMaker ist in der Lage, Alarme AWS IoT SiteWise und Ereignisse in AWS IoT TwinMaker Komponenten zu importieren. Auf diese Weise können Sie den Alarmstatus abfragen und Alarmschwellenwerte konfigurieren, ohne einen benutzerdefinierten Datenkonnektor für die AWS IoT SiteWise Datenmigration implementieren zu müssen. Sie können das AWS IoT TwinMaker Grafana-Plugin verwenden, um den Alarmstatus zu visualisieren und den Alarmschwellenwert in Grafana zu konfigurieren, ohne API-Aufrufe zu tätigen AWS IoT TwinMaker oder direkt mit AWS IoT SiteWise Alarmen zu interagieren.

> **Note**
>
> Hinweis zum Ende des Supports: Am 20. Mai 2026 AWS endet der Support für. AWS IoT Events Nach dem 20. Mai 2026 können Sie nicht mehr auf die AWS IoT Events Konsole oder AWS IoT Events die Ressourcen zugreifen. Weitere Informationen finden Sie unter AWS IoT Events Ende des Supports.

AWS IoT SiteWise Voraussetzungen für die Alarm-Konfiguration

Bevor Sie Alarme erstellen und in Ihr Grafana-Dashboard integrieren, stellen Sie sicher, dass Sie die folgenden Voraussetzungen überprüft haben:

- Machen Sie sich mit dem AWS IoT SiteWise Modell und dem Asset-System vertraut. Weitere Informationen finden Sie unter Erstellen von Asset-Modellen und Erstellen von Assets im AWS IoT SiteWise Benutzerhandbuch.
- Machen Sie sich mit den IoT-Events-Alarmmodellen vertraut und erfahren Sie, wie Sie sie an ein AWS IoT SiteWise Modell anhängen. Weitere Informationen finden Sie im AWS IoT SiteWise Benutzerhandbuch unter Definieren von AWS IoT Ereignisalarmen.

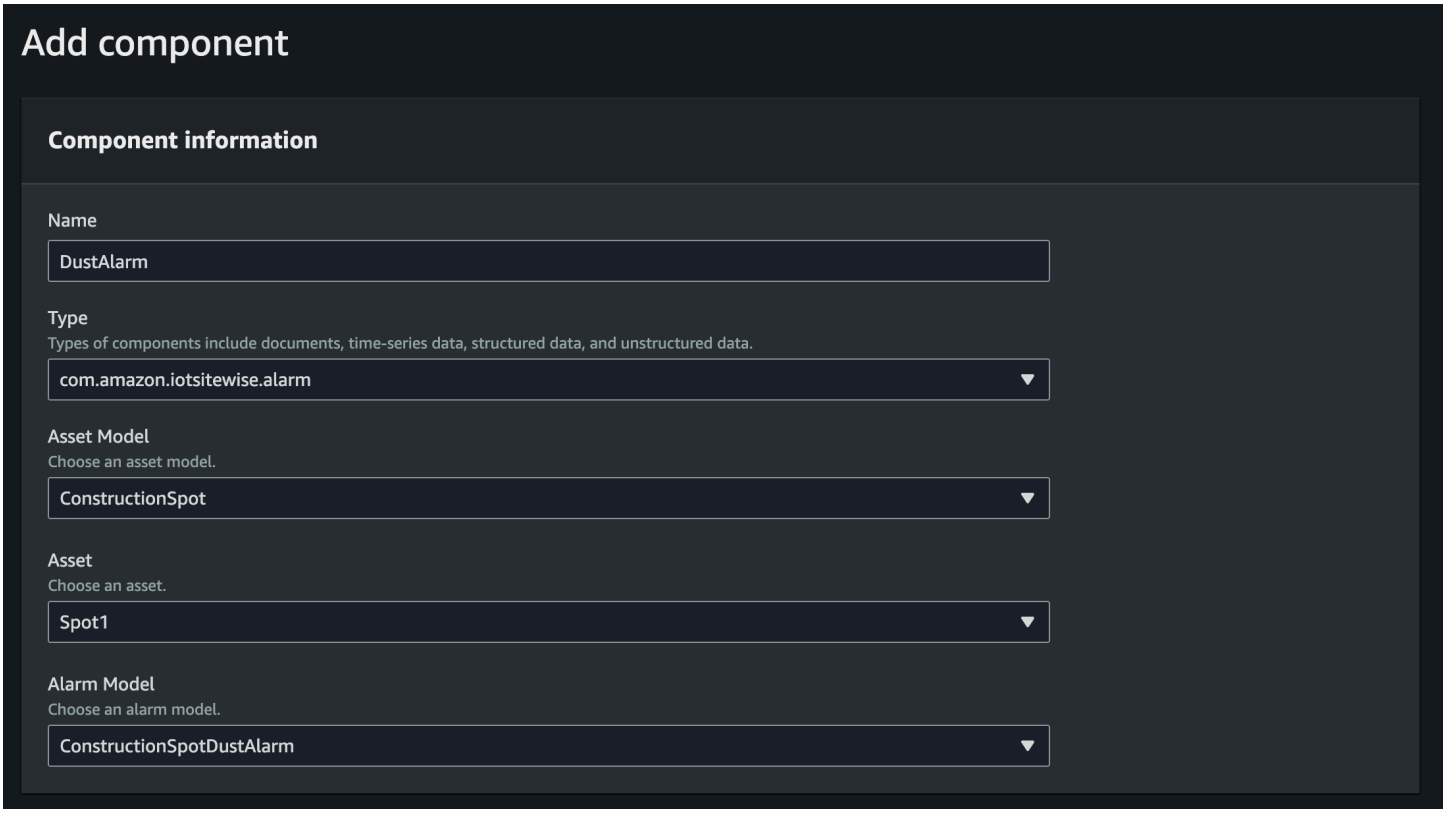
- Integrieren Sie AWS IoT TwinMaker sich in Grafana, damit Sie auf Ihre AWS IoT TwinMaker Ressourcen in Grafana zugreifen können. Weitere Informationen finden Sie unter AWS IoT TwinMaker Grafana-Dashboard-Integration.

Definieren Sie die AWS IoT SiteWise IAM-Rolle der Alarmkomponente

AWS IoT TwinMaker verwendet die Workspace-IAM-Rolle, um den Alarmschwellenwert in Grafana abzufragen und zu konfigurieren. Die folgenden Berechtigungen sind in der AWS IoT TwinMaker Workspace-Rolle erforderlich, um mit AWS IoT SiteWise Alarmen in Grafana interagieren zu können:

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

Erstellen Sie in der AWS IoT TwinMaker Konsole eine Entität, die Ihr AWS IoT SiteWise Asset darstellt. Stellen Sie sicher, dass Sie eine Komponente für diese Entität hinzufügen, die `com.amazon.iotsitewise.alarm` als Komponententyp verwendet wird, und wählen Sie die entsprechenden Asset- und Alarmmodelle aus.



Der obige Screenshot ist ein Beispiel für die Erstellung dieser Entität mit dem Typ. `com.amazon.iotsitewise.alarm`

Wenn Sie diese Komponente erstellen, AWS IoT TwinMaker werden automatisch die zugehörigen Alarm-Eigenschaften aus AWS IoT SiteWise und importiert AWS IoT Events. Sie können dieses Typmuster für Alarmkomponenten wiederholen, um Alarmkomponenten für alle in Ihrem Workspace benötigten Ressourcen zu erstellen.

Abfragen und Aktualisierungen über die AWS IoT TwinMaker API

Nachdem Sie die Alarmkomponenten erstellt haben, können Sie den Alarmstatus und den Schwellenwert über die AWS IoT TwinMaker API abfragen und die Alarmschwellenwerte aktualisieren.

Im Folgenden finden Sie ein Beispiel für eine Anfrage zur Abfrage des Alarmstatus:

```
aws iottwinmaker get-property-value-history --cli-input-json \
'{
    "workspaceId": "{workspaceId}",
    "entityId": "{entityId}",
```

```
    "componentName": "{componentName}",
    "selectedProperties": ["alarm_status"],
    "startTime": "{startTimeIsoString}",
    "endTime": "{endTimeIsoString}"
}'
```

Im Folgenden finden Sie ein Beispiel für eine Anfrage zur Abfrage des Alarmschwellenwerts.

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

Im Folgenden finden Sie ein Beispiel für eine Anfrage zur Aktualisierung des Alarmschwellenwerts:

```
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}"
                }
            ]
        }
    ]
}'
```

Konfigurieren Sie Ihr Grafana-Dashboard für Alarme

Es muss eine zweite Dashboard-IAM-Rolle mit Schreibzugriff erstellt werden. Dies ist eine normale Rolle, aber mit der Berechtigung, dass die Aktion `iottwinmaker:BatchPutPropertyValues` zum TwinMaker Workspace-Arn hinzugefügt werden kann, wie im folgenden Beispiel.

JSON

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

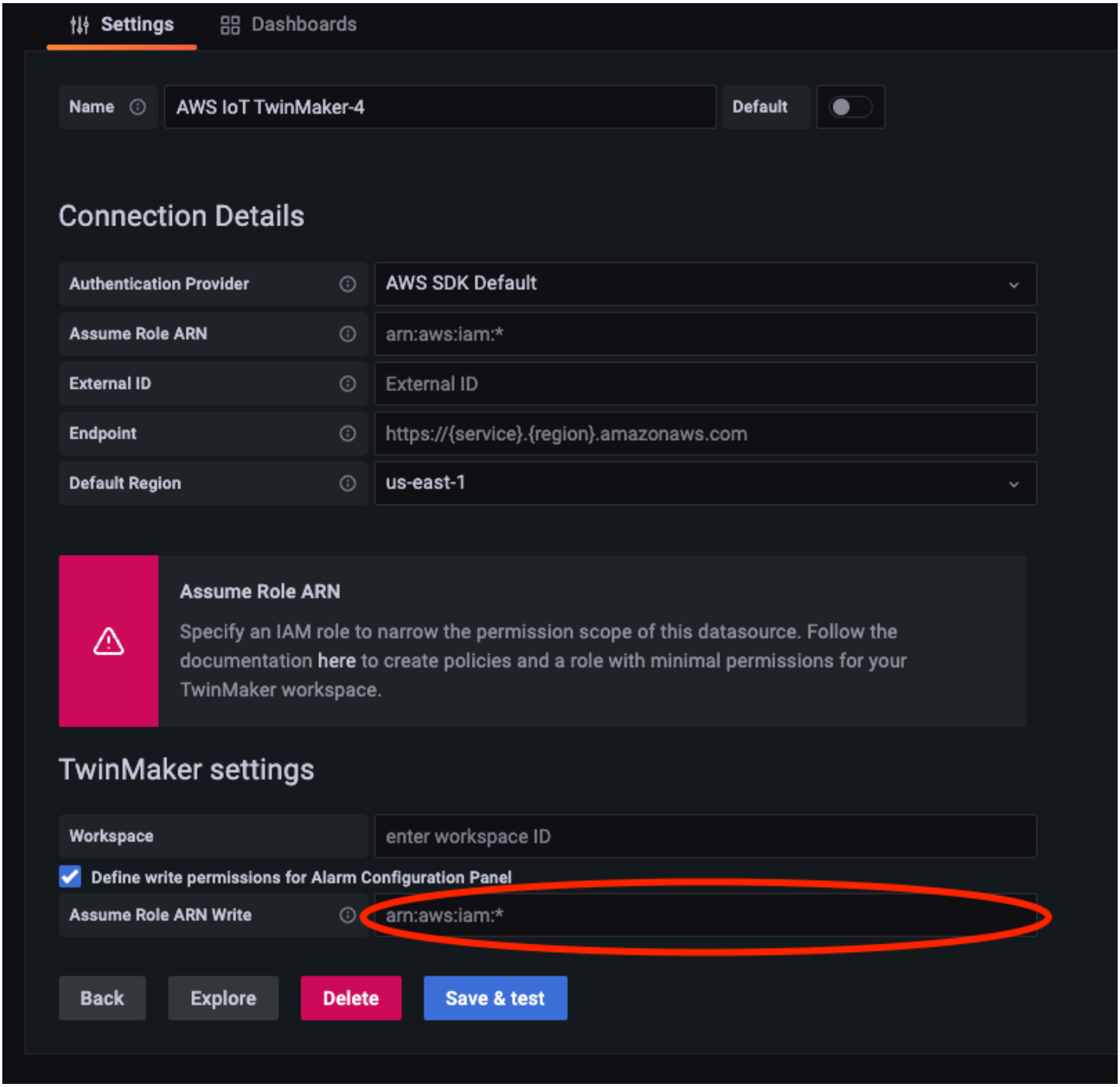
Alternativ können Sie stattdessen diese Aussage am Ende Ihrer IAM-Rolle hinzufügen:

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

```
        "{workspaceArn}/*"
    ]
}
```

Für die Datenquelle muss der Write-ARN mit der Dashboard-Write-Rolle festgelegt sein, die Sie erstellt haben.

Nachdem Sie Ihre IAM-Rolle geändert haben, melden Sie sich in Ihrem Grafana-Dashboard an, um die aktualisierte Rolle arn anzunehmen. Aktivieren Sie das Kontrollkästchen Schreibberechtigungen für das Alarm Configuration Panel definieren und kopieren Sie den ARN für die Rolle Write.



Verwenden Sie das Grafana-Dashboard zur Alarmvisualisierung

Gehen Sie wie folgt vor, um Ihrem Dashboard ein Alarmkonfigurationsfeld hinzuzufügen und es zu konfigurieren:

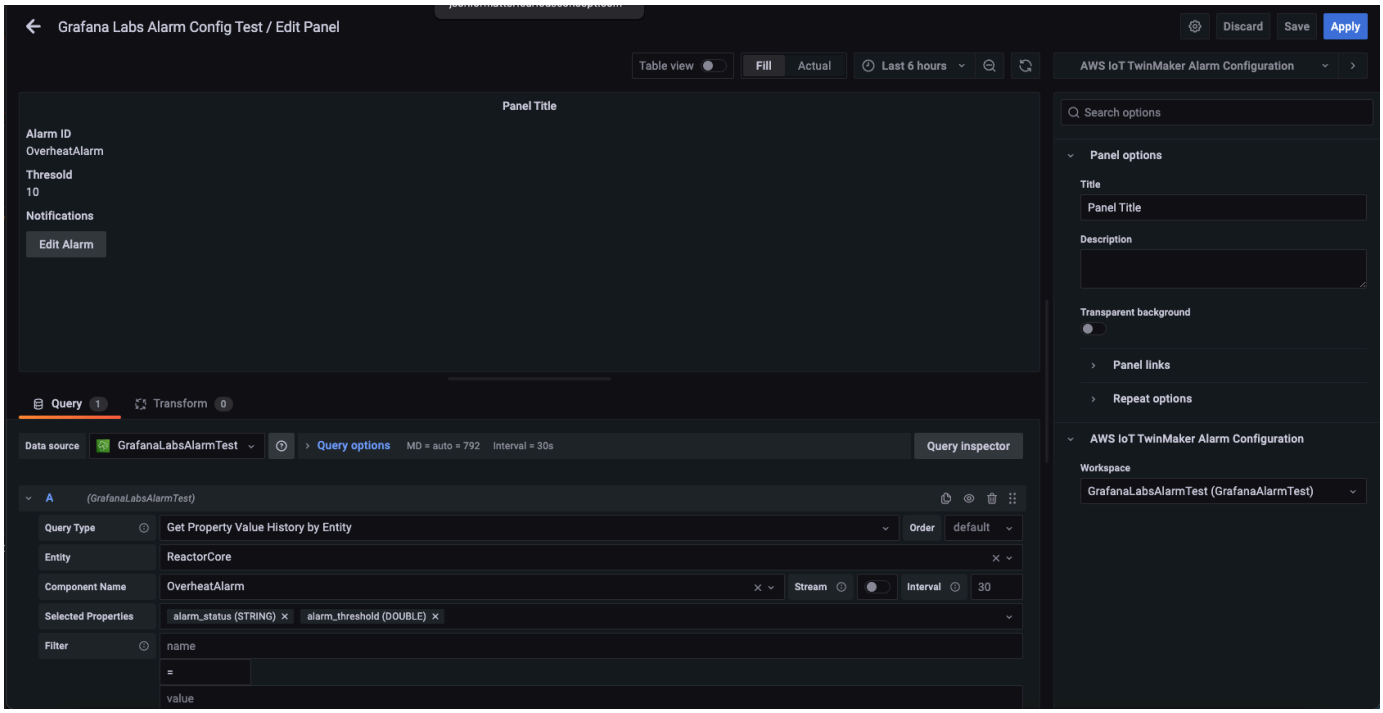
1. Wählen Sie den Arbeitsbereich in den Panel-Optionen aus.

2. Stellen Sie Ihre Datenquelle in der Abfragekonfiguration ein.
3. Verwenden Sie den folgenden Abfragetyp:. `Get Property Value History by Entity`
4. Wählen Sie eine Entität oder Entitätsvariable aus, der Sie einen Alarm hinzufügen möchten.
5. Nachdem Sie die Entität ausgewählt haben, wählen Sie eine Komponente oder Komponentenvariable aus, auf die Sie eine Eigenschaft anwenden möchten.
6. Wählen Sie für die Eigenschaft: `alarm_status` und`alarm_threshold`.

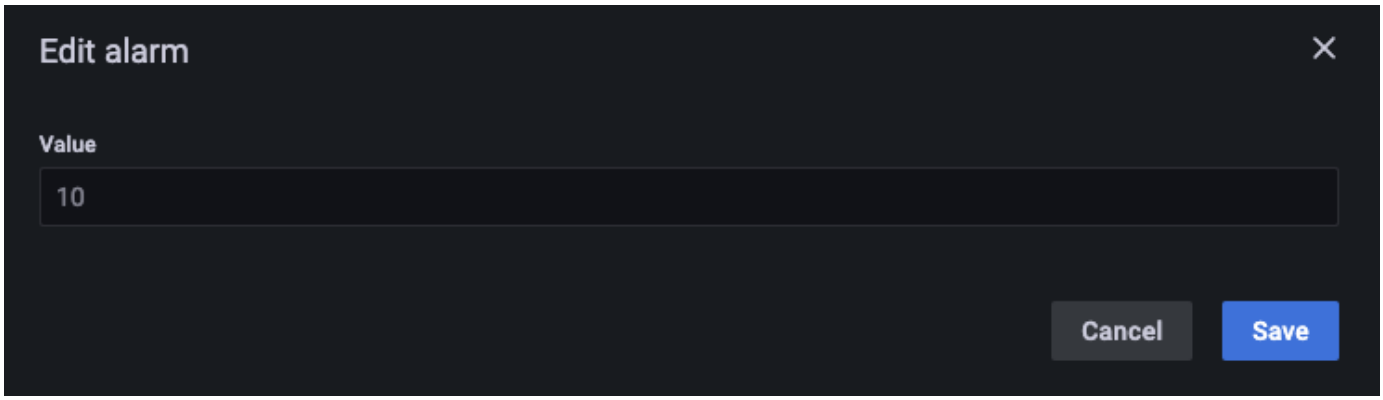
   Wenn die Verbindung hergestellt ist, sollten Sie die ID für die Alarm-ID und den aktuellen Schwellenwert sehen.

   > **Note**
   >
   > In der öffentlichen Vorschau werden keine Benachrichtigungen angezeigt. Sie sollten Ihren Alarmstatus und den Schwellenwert überprüfen, um sicherzustellen, dass die Eigenschaften korrekt angewendet wurden.

7. Es sollte die standardmäßige Abfragereihenfolge „Aufsteigend“ verwendet werden, damit der neueste Wert angezeigt wird.
8. Der Filterbereich der Abfrage kann leer gelassen werden. Eine vollständige Konfiguration ist unten abgebildet:



9. Mit der Schaltfläche „Alarm bearbeiten“ können Sie einen Dialog aufrufen, in dem Sie den aktuellen Alarmschwellenwert ändern können.
10. Wählen Sie Speichern, um den neuen Schwellenwert festzulegen.



> **Note**
>
> Dieses Fenster sollte nur mit einem Live-Zeitbereich verwendet werden, der die Gegenwart einschließt. Wenn Sie es für Zeitbereiche verwenden, die in der Vergangenheit enden und beginnen, werden möglicherweise unerwartete Werte angezeigt, wenn Alarmschwellenwerte immer als aktueller Schwellenwert bearbeitet werden.

AWS IoT TwinMaker Matterport-Integration

Matterport bietet eine Vielzahl von Aufnahmeoptionen, um reale Umgebungen zu scannen und immersive 3D-Modelle zu erstellen, die auch als digitale Zwillinge von Matterport bekannt sind. Diese Modelle werden Matterport-Räume genannt. AWS IoT TwinMaker unterstützt die Matterport-Integration, sodass Sie Ihre digitalen Matterport-Zwillinge in Ihre Szenen importieren können. AWS IoT TwinMaker Durch die Kopplung von Matterport Digital Twins mit AWS IoT TwinMaker können Sie Ihr digitales Zwillingssystem in einer virtuellen Umgebung visualisieren und überwachen.

Weitere Informationen zur Verwendung von Matterport finden Sie in der Dokumentation von Matterport auf der Matterport-Seite.AWS IoT TwinMaker

Themen zur Integration

- Überblick über die Integration
- Voraussetzungen für die Matterport-Integration
- Generieren und notieren Sie Ihre Matterport-Anmeldeinformationen
- Speichern Sie Ihre Matterport-Anmeldeinformationen in AWS Secrets Manager
- Importiert Matterport-Räume in Szenen AWS IoT TwinMaker
- Verwenden Sie Matterport-Bereiche in Ihrem AWS IoT TwinMaker Grafana-Dashboard

- Verwenden Sie Matterport-Bereiche in Ihrer Webanwendung AWS IoT TwinMaker

Überblick über die Integration

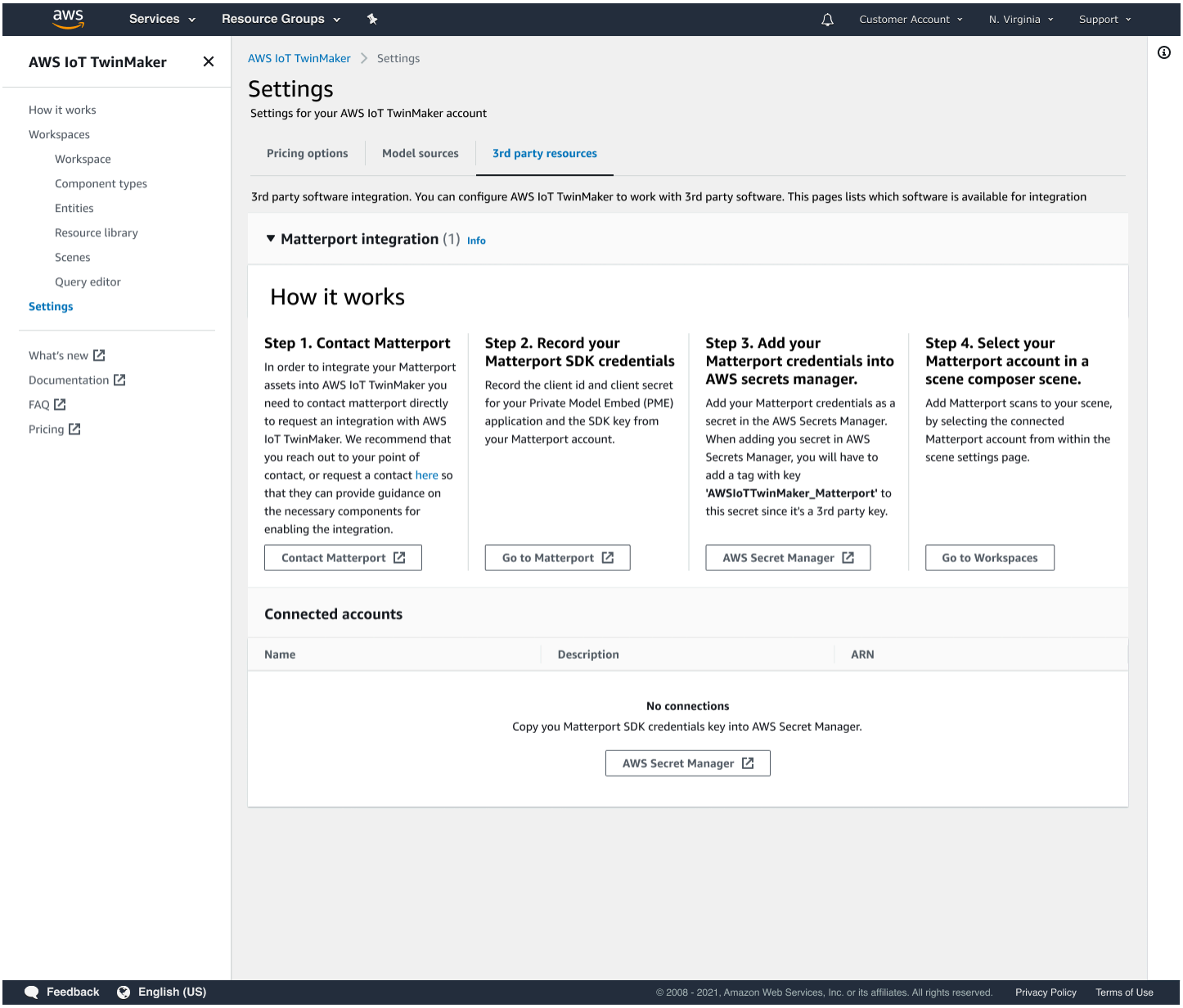
Diese Integration ermöglicht Ihnen Folgendes:

- Verwenden Sie Ihre Matterport-Tags und Leerzeichen im AWS IoT TwinMaker App-Kit.
- Sehen Sie sich Ihre importierten Matterport-Daten in Ihrem AWS IoT TwinMaker Grafana-Dashboard an. Weitere Informationen zur Verwendung von AWS IoT TwinMaker Grafana finden Sie in der Dokumentation zur Grafana-Dashboard-Integration.
- Importieren Sie Ihre Matterport-Räume in Ihre Szenen. AWS IoT TwinMaker
- Wählen Sie Ihre Matterport-Tags aus und importieren Sie sie, die Sie an Daten in Ihrer Szene binden möchten. AWS IoT TwinMaker
- Zeigen Sie automatisch Ihren Matterport-Bereich und Ihre Tag-Änderungen in Ihrer AWS IoT TwinMaker Szene an und genehmigen Sie, welche synchronisiert werden sollen.

Der Integrationsprozess besteht aus 3 kritischen Schritten.

1. Generieren und notieren Sie Ihre Matterport-Anmeldeinformationen
2. Speichern Sie Ihre Matterport-Anmeldeinformationen in AWS Secrets Manager
3. Importiert Matterport-Räume in Szenen AWS IoT TwinMaker

Sie starten Ihre Integration in der AWS IoT TwinMaker Konsole. Öffnen Sie auf der Einstellungsseite der Konsole unter Ressourcen von Drittanbietern die Matterport-Integration, um zwischen den verschiedenen Ressourcen zu navigieren, die für die Integration erforderlich sind.



Voraussetzungen für die Matterport-Integration

Bevor Sie Matterport mit integrieren, stellen AWS IoT TwinMaker Sie bitte sicher, dass Sie die folgenden Voraussetzungen erfüllen:

- Sie haben ein Matterport-Konto auf Unternehmensebene und die für die Integration erforderlichen Matterport-Produkte erworben. AWS IoT TwinMaker
- Sie haben AWS IoT TwinMaker einen Workspace. Weitere Informationen finden Sie unter Erste Schritte mit AWS IoT TwinMaker.

- Sie haben Ihre AWS IoT TwinMaker Workspace-Rolle aktualisiert. Weitere Informationen zum Erstellen einer Workspace-Rolle finden Sie unter Eine Servicerolle für erstellen und verwalten AWS IoT TwinMaker.

  Füge deiner Workspace-Rolle Folgendes hinzu:

  ```
  {
    "Effect": "Allow",
    "Action": "secretsmanager:GetSecretValue",
    "Resource": [
     "AWS Secrets Manager secret ARN"
    ]
  }
  ```

- Sie müssen sich an Matterport wenden, um die für die Aktivierung der Integration erforderliche Lizenzierung zu konfigurieren. Matterport wird auch ein Private Model Embed (PME) für die Integration aktivieren.

  Wenn Sie bereits einen Matterport Account Manager haben, wenden Sie sich direkt an diesen.

  Gehen Sie wie folgt vor, um Matterport zu kontaktieren und eine Integration zu beantragen, falls Sie keinen Ansprechpartner bei Matterport haben:

  1. Öffnen Sie die Matterport-Land-Seite. AWS IoT TwinMaker
  2. Drücken Sie die Schaltfläche Kontaktieren Sie uns, um das Kontaktformular zu öffnen.
  3. Füllen Sie die erforderlichen Informationen auf dem Formular aus.
  4. Wenn Sie bereit sind, wählen Sie SAY HELLO, um Ihre Anfrage an Matterport zu senden.

  Sobald Sie die Integration angefordert haben, können Sie die erforderlichen Matterport SDK- und PME-Anmeldeinformationen (Private Model Embed) generieren, die für die Fortsetzung des Integrationsprozesses erforderlich sind.

  > **Note**
  >
  > Dies kann bedeuten, dass für den Kauf neuer Produkte oder Dienstleistungen eine Gebühr anfällt.

Generieren und notieren Sie Ihre Matterport-Anmeldeinformationen

Um Matterport zu integrieren AWS IoT TwinMaker, müssen Sie Matterport-Anmeldeinformationen angeben AWS Secrets Manager . Verwenden Sie das folgende Verfahren, um die Matterport SDK-Anmeldeinformationen zu generieren.

1. Loggen Sie sich in Ihr Matterport-Konto ein.
2. Navigieren Sie zu Ihrer Kontoeinstellungsseite.
3. Wählen Sie auf der Einstellungsseite die Option Entwicklertools aus.
4. Gehen Sie auf der Seite mit den Entwicklertools zum Abschnitt SDK-Schlüsselverwaltung.
5. Wählen Sie im Abschnitt SDK-Schlüsselverwaltung die Option zum Hinzufügen eines neuen SDK-Schlüssels aus.
6. Sobald Sie den Matterport SDK-Schlüssel haben, fügen Sie Domains zum Schlüssel für AWS IoT TwinMaker und Ihren Grafana hinzu. Wenn Sie das AWS IoT TwinMaker App-Kit verwenden, stellen Sie sicher, dass Sie auch Ihre benutzerdefinierte Domain hinzufügen.
7. Suchen Sie als Nächstes den Abschnitt Anwendungsintegrationsmanagement. Dort sollte Ihre PME-Anwendung aufgeführt sein. Notieren Sie sich die folgenden Informationen:

   - Die Client-ID
   - Das geheime Kundengeheimnis

   > **Note**
   >
   > Da Ihnen das Kundengeheimnis nur einmal vorgelegt wird, empfehlen wir Ihnen dringend, Ihr Kundengeheimnis aufzuzeichnen. Sie müssen Ihr Client Secret in der AWS Secrets Manager Konsole angeben, um mit der Matterport-Integration fortzufahren.

   Diese Anmeldeinformationen werden automatisch erstellt, wenn Sie die erforderlichen Komponenten gekauft haben und das PME für Ihr Konto von Matterport aktiviert wurde. Wenn diese Anmeldeinformationen nicht angezeigt werden, wenden Sie sich an Matterport. Informationen zur Kontaktanfrage finden Sie im Matterport und im Kontaktformular. AWS IoT TwinMaker

Weitere Informationen zu den Matterport SDK-Anmeldeinformationen finden Sie in der offiziellen SDK-Dokumentation von Matterport im Überblick über die SDK-Dokumente.

Speichern Sie Ihre Matterport-Anmeldeinformationen in AWS Secrets Manager

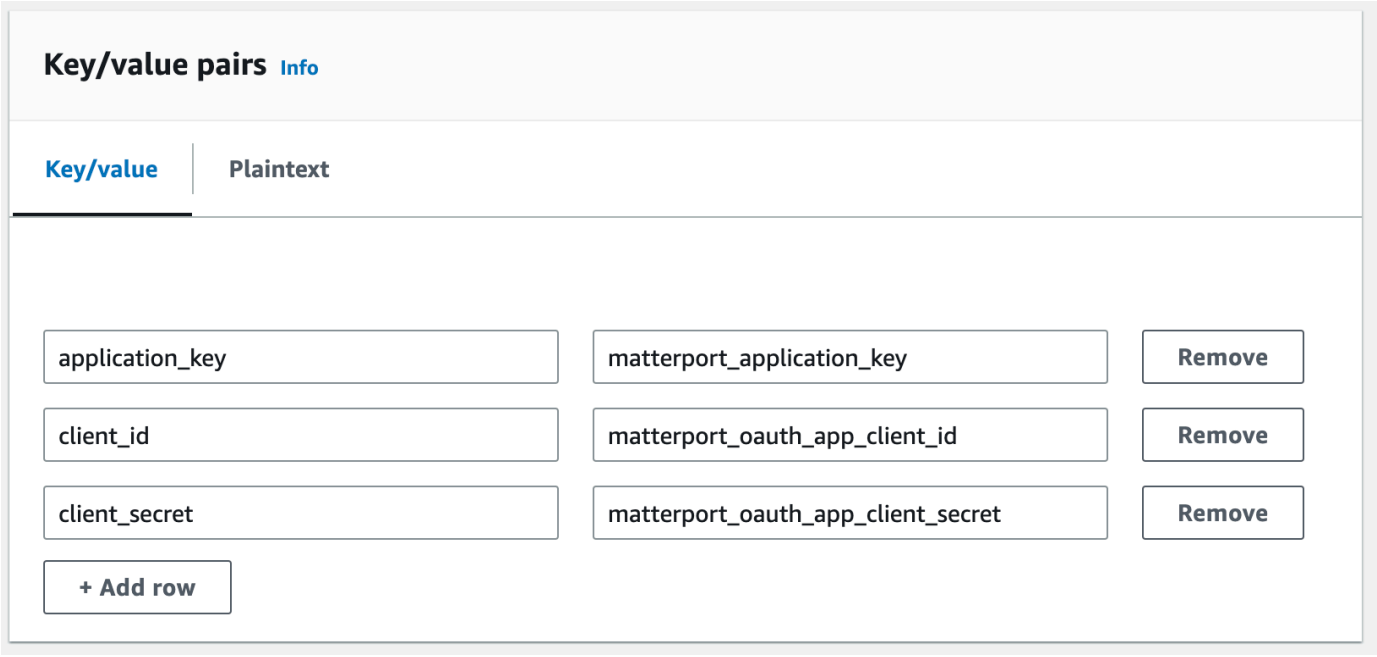
Verwenden Sie das folgende Verfahren, um Ihre Matterport-Anmeldeinformationen in zu speichern. AWS Secrets Manager

> **Note**
>
> Sie benötigen den Client Secret, der anhand der Prozedur im Generieren und notieren Sie Ihre Matterport-Anmeldeinformationen Thema erstellt wurde, um mit der Matterport-Integration fortzufahren.

1. Loggen Sie sich in die AWS Secrets Manager Konsole ein.
2. Navigieren Sie zur Seite Secrets und wählen Sie Neues Secret speichern aus.
3. Wählen Sie für den Geheimtyp die Option Anderer Geheimtyp aus.
4. Fügen Sie im Abschnitt Schlüssel/Wert-Paare die folgenden Schlüssel-Wert-Paare mit Ihren Matterport-Anmeldeinformationen als Werten hinzu:

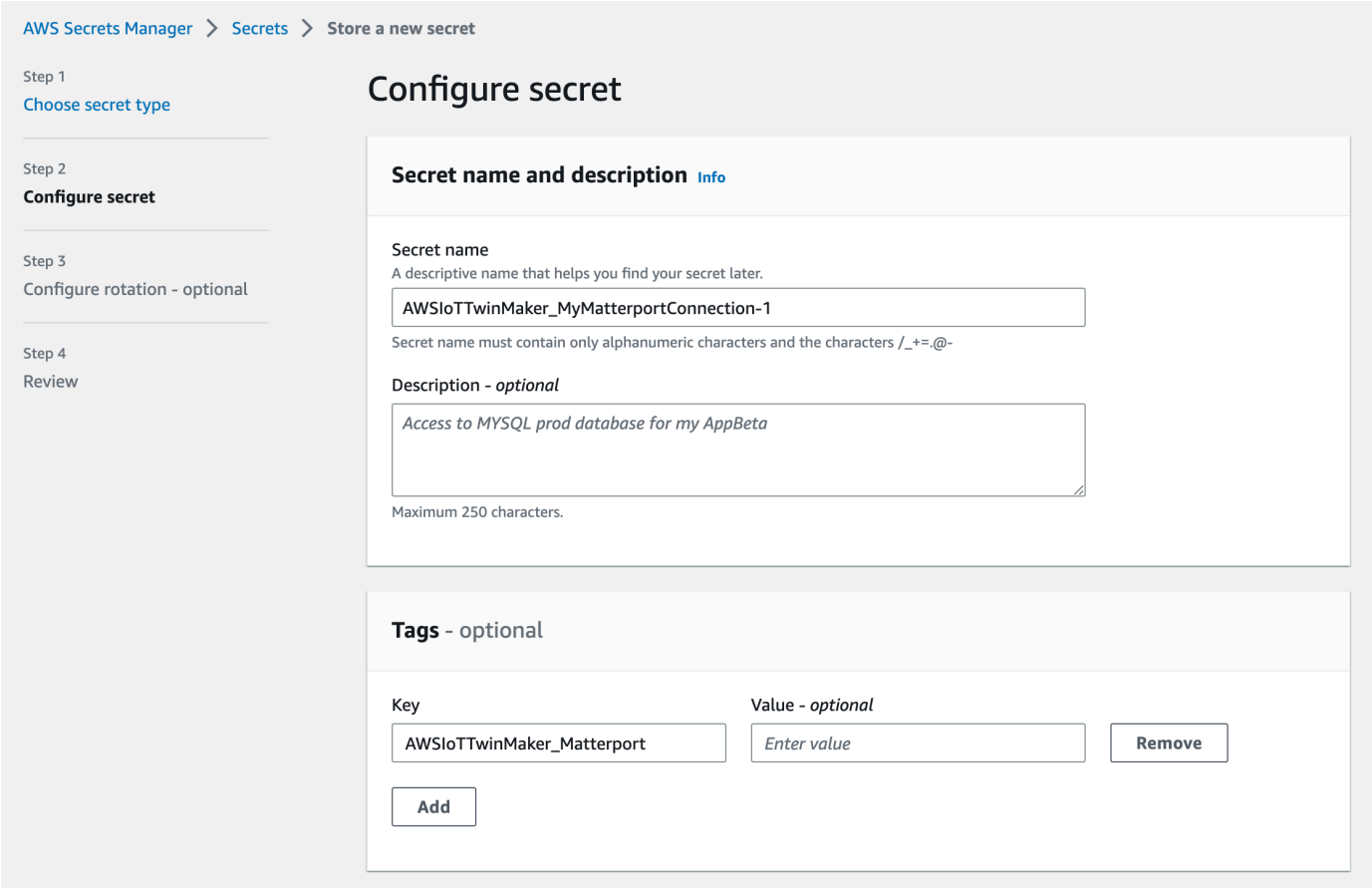
   - Erstellen Sie ein Schlüssel-Wert-Paar mit Key: und Value:. **`application_key`** *`<your Matterport credentials>`*
   - Erstellen Sie ein Schlüssel-Wert-Paar mit Key: und Value:**`client_id`**. *`<your Matterport credentials>`*
   - Erstellen Sie ein Schlüssel-Wert-Paar mit Key: und Value:**`client_secret`**. *`<your Matterport credentials>`*

   Wenn Sie fertig sind, sollten Sie eine Konfiguration haben, die dem folgenden Beispiel ähnelt:



5. Für den Verschlüsselungsschlüssel können Sie den Standard-Verschlüsselungsschlüssel ausgewählt lassen. `aws/secretsmanager`
6. Wählen Sie Weiter, um zur Seite „Geheimes Passwort konfigurieren“ zu gelangen.
7. Füllen Sie das Feld für den geheimen Namen und die Beschreibung aus.
8. Fügen Sie diesem Geheimnis im Abschnitt Tags ein Tag hinzu.

   Wenn Sie das Tag erstellen, weisen Sie den Schlüssel `AWSIoTTwinMaker_Matterport` wie im folgenden Screenshot gezeigt zu:



> **Note**
>
> Sie müssen ein Tag hinzufügen. Beim Hinzufügen von Geheimnissen von Drittanbietern sind Tags erforderlich AWS Secrets Manager, obwohl Tags als optional aufgeführt sind.

Das Feld Value (Wert) ist optional. Nachdem Sie einen Schlüssel eingegeben haben, können Sie Hinzufügen auswählen, um mit dem nächsten Schritt fortzufahren.

9. Wählen Sie Weiter, um zur Seite „Rotation konfigurieren“ zu gelangen. Das Einrichten einer geheimen Rotation ist optional. Wenn Sie das Hinzufügen Ihres Geheimnisses beenden möchten und keine Rotation benötigen, wählen Sie erneut Weiter. Weitere Informationen zur Rotation von Geheimnissen finden Sie unter AWS Secrets Manager Geheimnisse rotieren.
10. Bestätigen Sie Ihre geheime Konfiguration auf der Überprüfungsseite. Wenn du bereit bist, dein Geheimnis hinzuzufügen, wähle Store.

Weitere Informationen zur Verwendung AWS Secrets Manager finden Sie in der folgenden AWS Secrets Manager Dokumentation:

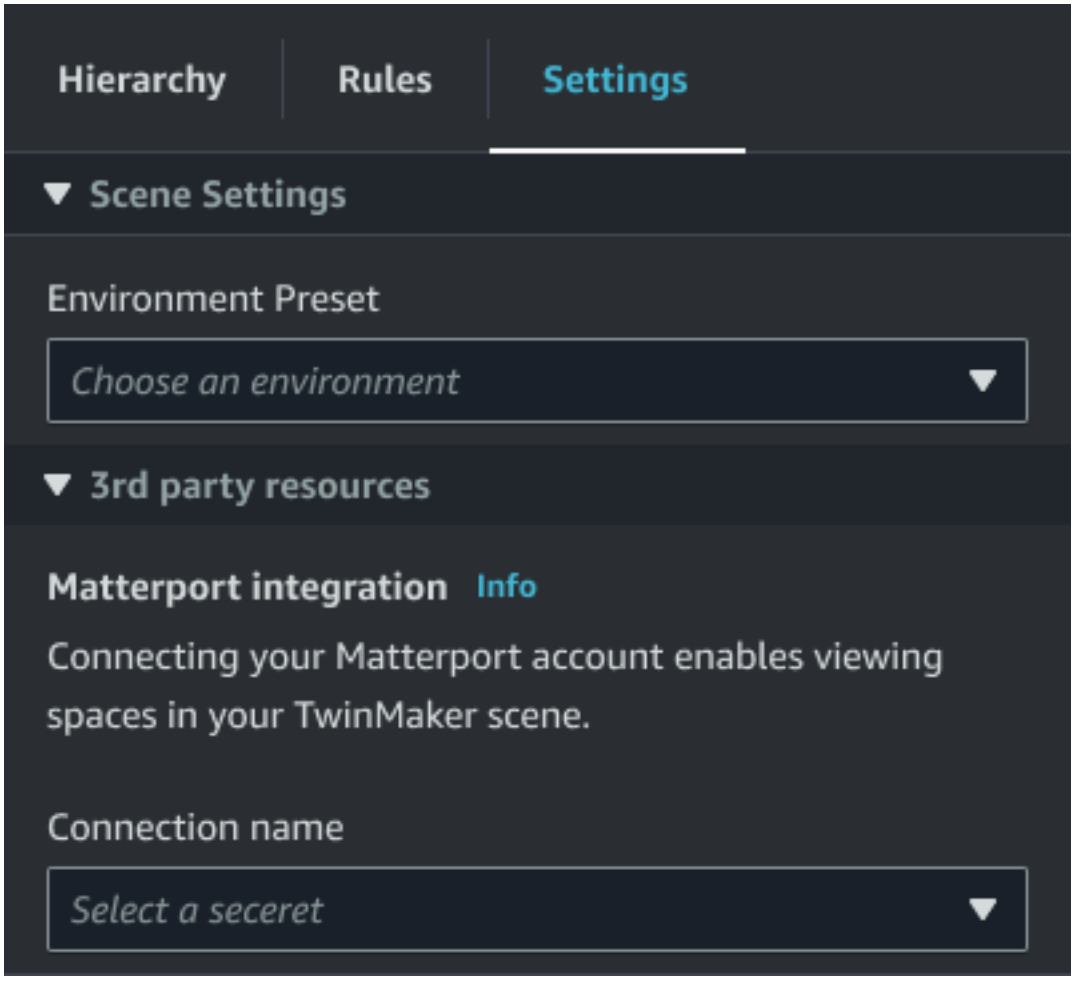
- Geheimnisse erstellen und verwalten mit AWS Secrets Manager
- Was ist AWS Secrets Manager?
- AWS Secrets Manager Geheimnisse rotieren

Jetzt sind Sie bereit, Ihre Matterport-Assets in AWS IoT TwinMaker Szenen zu importieren. Sehen Sie sich das Verfahren im folgenden Abschnitt an, Importiert Matterport-Räume in Szenen AWS IoT TwinMaker

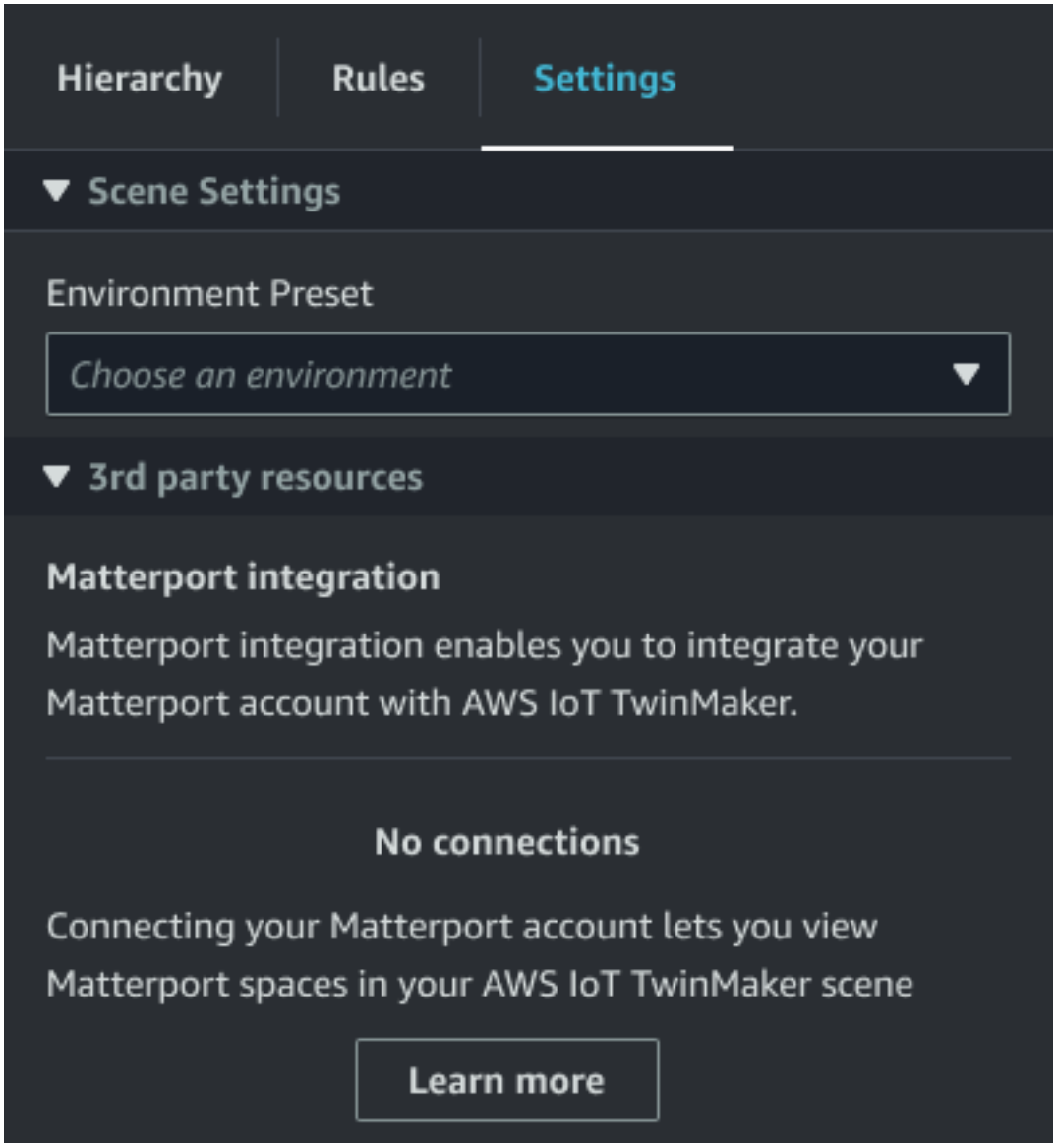
Importiert Matterport-Räume in Szenen AWS IoT TwinMaker

Fügen Sie Ihrer Szene Matterport-Scans hinzu, indem Sie das verbundene Matterport-Konto auf der Seite mit den Szeneneinstellungen auswählen. Gehen Sie wie folgt vor, um Ihre Matterport-Scans und -Tags zu importieren:

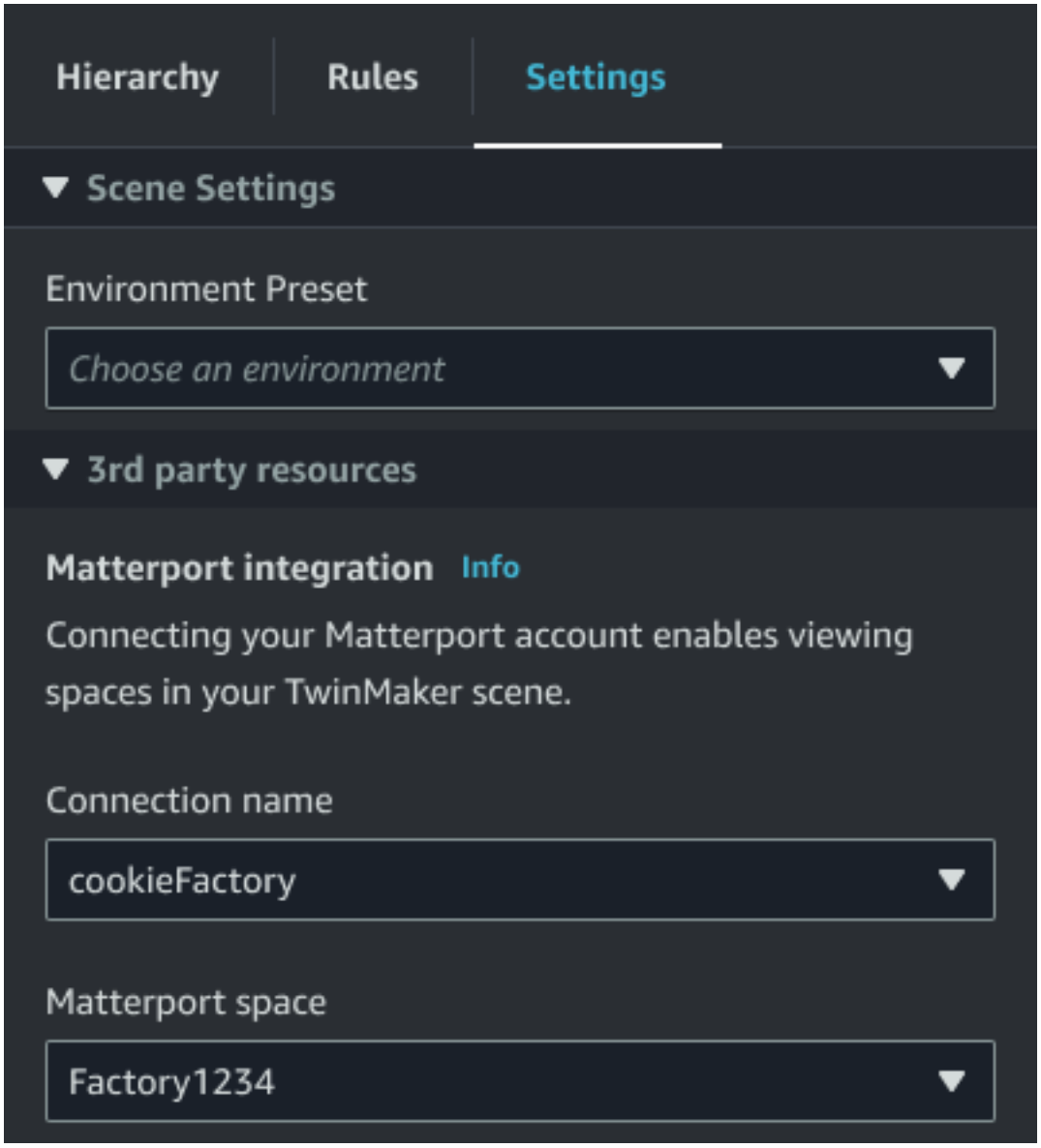
1. Melden Sie sich bei der AWS IoT TwinMaker -Konsole an.
2. Erstellen oder öffnen Sie eine bestehende AWS IoT TwinMaker Szene, in der Sie einen Matterport-Raum verwenden möchten.
3. Sobald die Szene geöffnet wurde, navigieren Sie zur Registerkarte Einstellungen.
4. Suchen Sie in den Einstellungen unter Ressourcen von Drittanbietern nach dem Verbindungsnamen und geben Sie den geheimen Schlüssel ein, den Sie in der Prozedur von erstellt habenSpeichern Sie Ihre Matterport-Anmeldeinformationen in AWS Secrets Manager.



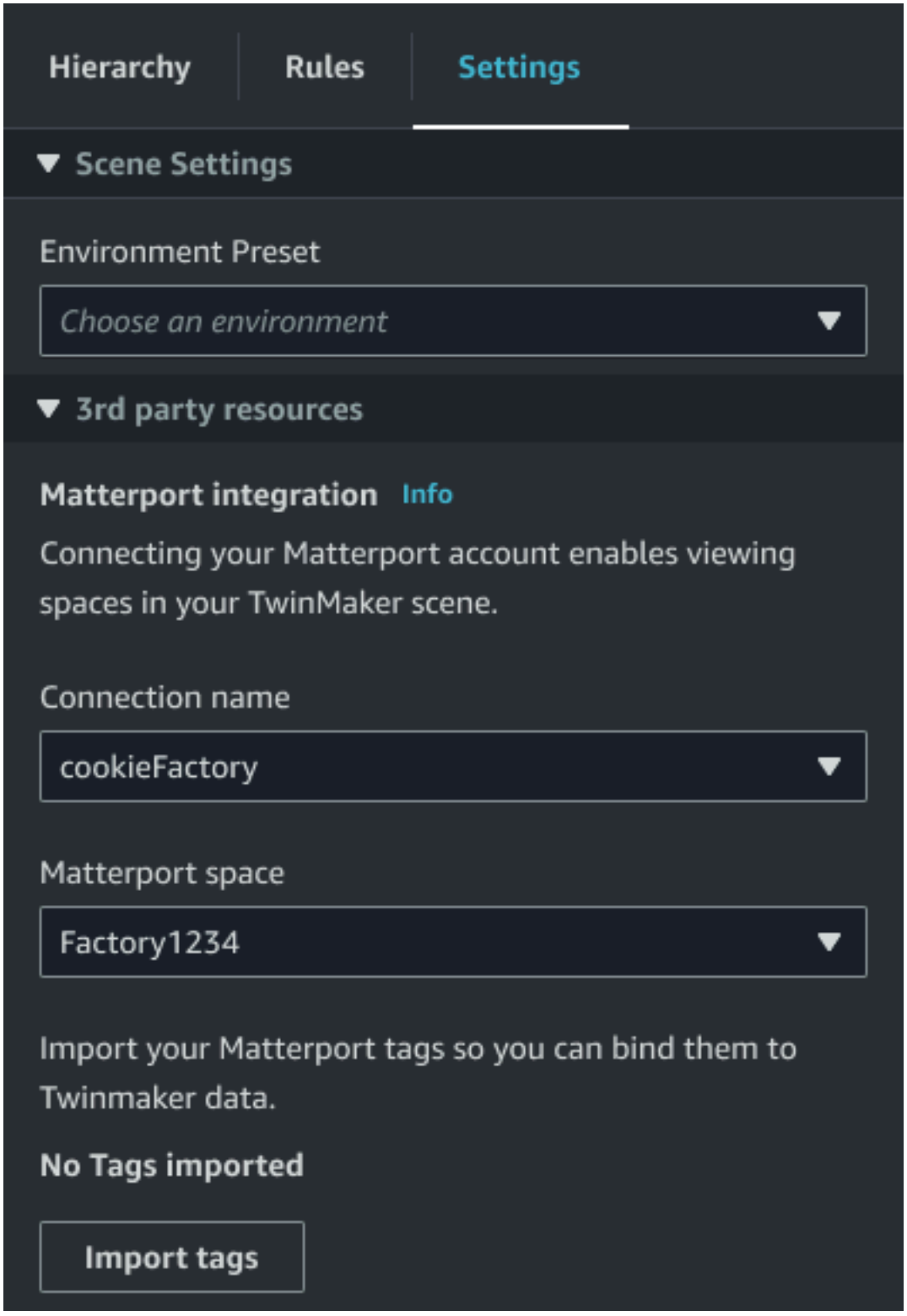
> **Note**
>
> Wenn Sie die Meldung „Keine Verbindungen“ sehen, navigieren Sie zur Seite mit den AWS IoT TwinMaker Konsoleneinstellungen, um mit der Matterport-Integration zu beginnen.



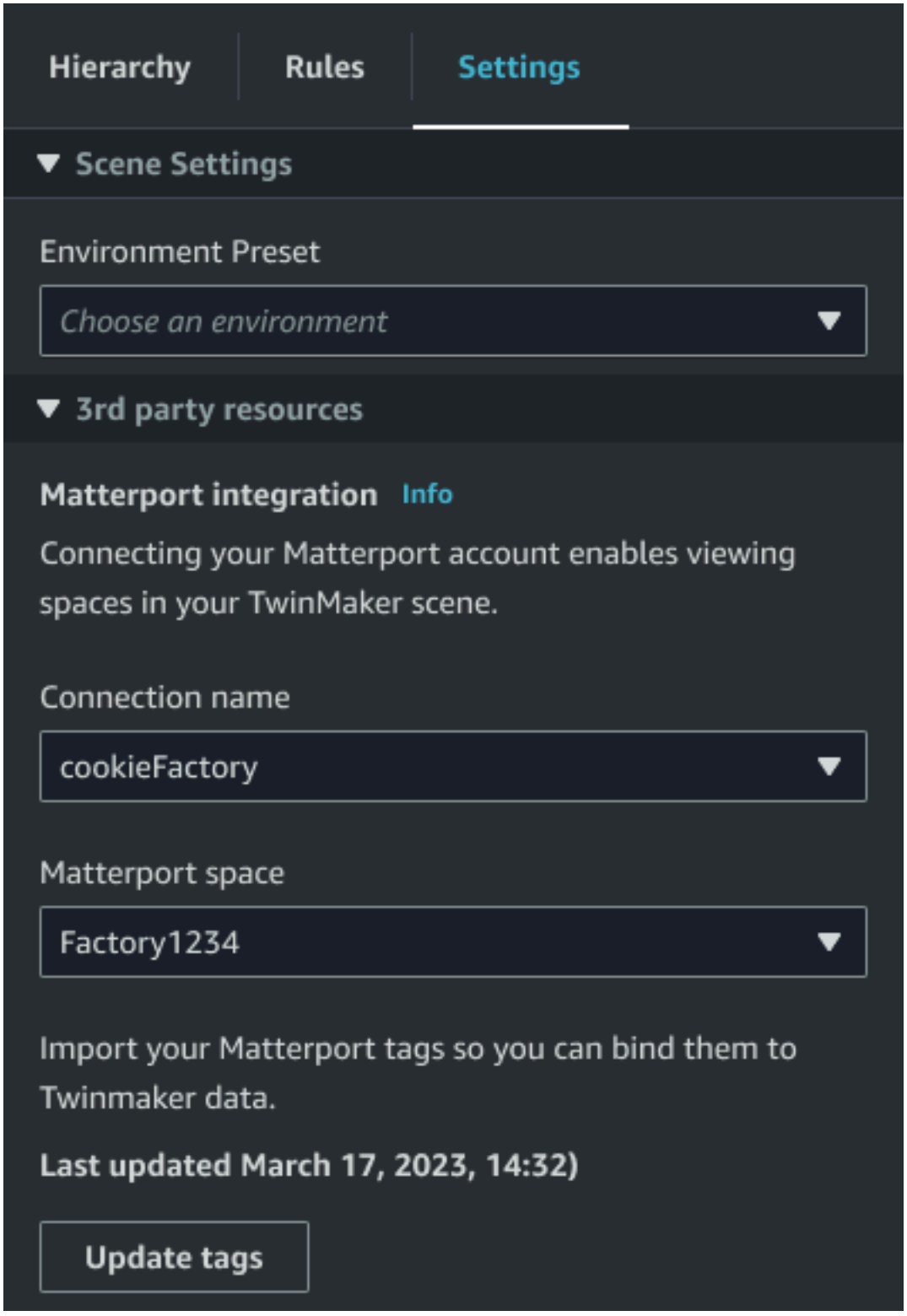
5. Wählen Sie als Nächstes den Matterport-Raum aus, den Sie in Ihrer Szene verwenden möchten, indem Sie ihn im Drop-down-Menü „Matterport-Raum“ auswählen.



6. Nachdem Sie einen Bereich ausgewählt haben, können Sie Ihre Matterport-Tags importieren und sie in AWS IoT TwinMaker Szenen-Tags konvertieren, indem Sie auf die Schaltfläche Tags importieren klicken.



Nachdem Sie Matterport-Tags importiert haben, wird die Schaltfläche durch die Schaltfläche Tags aktualisieren ersetzt. Sie können Ihre Matterport-Tags kontinuierlich aktualisieren, AWS IoT TwinMaker sodass sie immer die neuesten Änderungen in Ihrem Matterport-Konto widerspiegeln.



7. Sie haben die Integration AWS IoT TwinMaker mit Matterport erfolgreich abgeschlossen, und jetzt enthält Ihre AWS IoT TwinMaker Szene sowohl Ihren importierten Matterport-Bereich als

auch Ihre Tags. Sie können in dieser Szene wie in jeder anderen Szene arbeiten. AWS IoT TwinMaker

Weitere Informationen zum Arbeiten mit AWS IoT TwinMaker Szenen finden Sie unter AWS IoT TwinMaker Szenen erstellen und bearbeiten.

Verwenden Sie Matterport-Bereiche in Ihrem AWS IoT TwinMaker Grafana-Dashboard

Sobald Sie Ihren Matterport-Bereich in eine AWS IoT TwinMaker Szene importiert haben, können Sie diese Szene mit dem Matterport-Bereich in Ihrem Grafana-Dashboard anzeigen. Wenn Sie Grafana bereits mit konfiguriert haben AWS IoT TwinMaker, können Sie einfach das Grafana-Dashboard öffnen, um Ihre Szene mit dem importierten Matterport-Bereich anzusehen.

Wenn Sie noch nicht AWS IoT TwinMaker mit Grafana konfiguriert haben, schließen Sie zuerst den Grafana-Integrationsprozess ab. Bei der Integration AWS IoT TwinMaker mit Grafana haben Sie zwei Möglichkeiten. Sie können eine selbstverwaltete Grafana-Instance oder Amazon Managed Grafana verwenden.

In der folgenden Dokumentation erfahren Sie mehr über die Grafana-Optionen und den Integrationsprozess:

- AWS IoT TwinMaker Grafana-Dashboard-Integration.
- Amazon hat Grafana verwaltet.
- Selbstverwaltetes Grafana.

Verwenden Sie Matterport-Bereiche in Ihrer Webanwendung AWS IoT TwinMaker

Sobald Sie Ihren Matterport-Bereich in eine AWS IoT TwinMaker Szene importiert haben, können Sie diese Szene mit dem Matterport-Bereich in Ihrer App-Kit-Webanwendung anzeigen. AWS IoT

In der folgenden Dokumentation erfahren Sie mehr über die Verwendung des Anwendungskits: AWS IoT

- Weitere Informationen zur Verwendung AWS IoT TwinMaker mit dem AWS IoT App-Kit finden Sie unterErstellen Sie eine benutzerdefinierte Webanwendung mithilfe von AWS IoT TwinMaker UI-Komponenten.
- Weitere Informationen zur Verwendung des AWS IoT Anwendungskits finden Sie auf der Github-Seite des AWS IoT Anwendungskits.
- Anweisungen zum Starten einer neuen Webanwendung mithilfe des AWS IoT Anwendungskits finden Sie auf der offiziellen IoT App Kit-Dokumentationsseite.

AWS IoT TwinMaker Videointegration

Videokameras bieten eine gute Möglichkeit für die Simulation digitaler Zwillinge. Sie können AWS IoT TwinMaker damit den Standort und die physikalischen Bedingungen Ihrer Kamera simulieren. Erstellen Sie Entitäten AWS IoT TwinMaker für Ihre Kameras vor Ort und verwenden Sie Videokomponenten, um Live-Videos und Metadaten von Ihrer Site in Ihre AWS IoT TwinMaker Szene oder in ein Grafana-Dashboard zu streamen.

AWS IoT TwinMaker kann Videos von Edge-Geräten auf zwei Arten aufnehmen. Sie können Videos von Edge-Geräten mit dem Edge-Anschluss für Kinesis-Videostream streamen, oder Sie können Videos auf dem Edge-Gerät speichern und das Hochladen von Videos mit MQTT-Nachrichten starten. Verwenden Sie diese Komponente, um Videodaten von Ihren Geräten zur Verwendung mit Diensten zu streamen. AWS IoT Informationen zum Generieren der erforderlichen Ressourcen und zum Bereitstellen des Edge-Connectors für Kinesis Video Streams finden Sie unter Erste Schritte mit dem Edge-Connector für Kinesis-Videostream. GitHub Weitere Informationen zu der AWS IoT Greengrass Komponente finden Sie in der AWS IoT Greengrass Dokumentation zum Edge-Connector für Kinesis Video Streams.

Nachdem Sie die erforderlichen AWS IoT SiteWise Modelle erstellt und die Kinesis Video Streams Greengrass-Komponente konfiguriert haben, können Sie Videos am Edge zu Ihrer digitalen Zwillingsanwendung in der AWS IoT TwinMaker Konsole streamen oder aufnehmen. Sie können auch Livestreams und Metadaten von Ihren Geräten in einem Grafana-Dashboard anzeigen. Weitere Informationen zur Integration von Grafana und finden Sie AWS IoT TwinMaker unterAWS IoT TwinMaker Grafana-Dashboard-Integration.

Verwenden Sie den Edge-Anschluss für Kinesis-Videostream, um Videos einzuspielen AWS IoT TwinMaker

Mit dem Edge-Anschluss für Kinesis-Videostream können Sie Video und Daten an eine Entität in Ihrer AWS IoT TwinMaker Szene streamen. Dazu verwenden Sie eine Videokomponente. Gehen Sie wie folgt vor, um die Videokomponente zur Verwendung in Ihren Szenen zu erstellen.

Voraussetzungen

Bevor Sie die Videokomponente in Ihrer AWS IoT TwinMaker Szene erstellen, stellen Sie sicher, dass Sie die folgenden Voraussetzungen erfüllt haben.

- Die erforderlichen AWS IoT SiteWise Modelle und Ressourcen für den Edge-Anschluss für Kinesis-Videostream wurden erstellt. Weitere Informationen zum Erstellen der AWS IoT SiteWise Assets für den Connector finden Sie unter Erste Schritte mit dem Edge-Connector für Kinesis-Videostream.
- Der Kinesis Video Stream Edge Connector wurde auf Ihrem AWS IoT Greengrass Gerät bereitgestellt. Weitere Informationen zur Bereitstellung der Kinesis Video Stream Edge Connector-Komponente finden Sie in der Bereitstellungs-README-Datei.

Erstellen Sie Videokomponenten für Szenen AWS IoT TwinMaker

Gehen Sie wie folgt vor, um den Edge-Connector für die Kinesis-Videostream-Komponente für Ihre Szene zu erstellen.

1. Öffnen Sie in der AWS IoT TwinMaker Konsole die Szene, zu der Sie die Videokomponente hinzufügen möchten.
2. Nachdem die Szene geöffnet wurde, wählen Sie eine vorhandene Entität aus oder erstellen Sie die Entität, zu der Sie die Komponente hinzufügen möchten, und wählen Sie dann Komponente hinzufügen.
3. Geben Sie im Bereich Komponente hinzufügen einen Namen für die Komponente ein und wählen Sie als Typ die Option com.amazon.iotsitewise.connector.edgevideo aus.
4. Wählen Sie ein Asset-Modell aus, indem Sie den Namen des Kameramodells auswählen, das Sie erstellt haben. AWS IoT SiteWise Dieser Name sollte das folgende Format haben:`EdgeConnectorForKVSCameraModel-0abc`, wobei die Buchstaben- und Zahlenfolge am Ende Ihrem eigenen Asset-Namen entspricht.
5. Wählen Sie unter Asset die AWS IoT SiteWise Kamera-Assets aus, von denen Sie Videos streamen möchten. Es erscheint ein kleines Fenster mit einer Vorschau des aktuellen Videostreams.

   > **Note**
   >
   > Um Ihr Videostreaming zu testen, wählen Sie Test. Dieser Test sendet ein MQTT-Ereignis aus, um das Video-Live-Streaming zu initiieren. Warten Sie einen Moment, bis das Video im Player angezeigt wird.

6. Um die Videokomponente zu Ihrer Entität hinzuzufügen, wählen Sie Komponente hinzufügen.

Fügen Sie Video und Metadaten aus dem Kinesis-Videostream zu einem Grafana-Dashboard hinzu

Nachdem Sie eine Videokomponente für Ihre Entität in Ihrer AWS IoT TwinMaker Szene erstellt haben, können Sie das Videopanel in Grafana so konfigurieren, dass Live-Streams angezeigt werden. Stellen Sie sicher, dass Sie ordnungsgemäß in Grafana AWS IoT TwinMaker integriert sind. Weitere Informationen finden Sie unter AWS IoT TwinMaker Grafana-Dashboard-Integration.

> **Important**
>
> Um Videos in Ihrem Grafana-Dashboard anzusehen, müssen Sie sicherstellen, dass die Grafana-Datenquellen über die richtigen IAM-Berechtigungen verfügen. Informationen zum Erstellen der erforderlichen Rolle und Richtlinie finden Sie unter. Eine Dashboard-IAM-Rolle erstellen

Führen Sie die folgenden Schritte aus, um Kinesis Video Streams und Metadaten in Ihrem Grafana-Dashboard anzuzeigen.

1. Öffnen Sie das Dashboard AWS IoT TwinMaker .
2. Wählen Sie „Bereich hinzufügen“ und anschließend „Leeres Bedienfeld hinzufügen“.

   > **Note**
   >
   > Für Grafana v10.4 befindet sich der AWS IoT TwinMaker Videoplayer unter Widget. Wählen Sie Hinzufügen >> Widget.

3. Wählen Sie aus der Liste der Panels das AWS IoT TwinMaker Videoplayer-Bedienfeld aus.
4. Geben Sie im AWS IoT TwinMaker Video-Player-Bereich den Stream-Namen des sowie den Namen des Kinesis-Videostreams ein KinesisVideoStreamName, von dem Sie Video streamen möchten.

   > **Note**
   >
   > Um Metadaten in das Grafana-Videopanel zu streamen, müssen Sie zunächst eine Entität mit einer Video-Streaming-Komponente erstellt haben.

5. Optional: Um Metadaten von AWS IoT SiteWise Assets zum Videoplayer zu streamen, wählen Sie für Entity die AWS IoT TwinMaker Entität aus, die Sie in Ihrer AWS IoT TwinMaker Szene erstellt haben. Wählen Sie als Komponentennamen die Videokomponente aus, die Sie für die Entität in Ihrer AWS IoT TwinMaker Szene erstellt haben.

Verwendung der AWS IoT TwinMaker Flink-Bibliothek

AWS IoT TwinMaker bietet eine Flink-Bibliothek, mit der Sie Daten lesen und in externe Datenspeicher schreiben können, die in Ihren digitalen Zwillingen verwendet werden.

Sie verwenden die AWS IoT TwinMaker Flink-Bibliothek, indem Sie sie als benutzerdefinierten Konnektor in Managed Service für Apache Flink installieren und Flink-SQL-Abfragen in einem Zeppelin-Notizbuch in Managed Service für Apache Flink ausführen. Das Notebook kann zu einer kontinuierlich laufenden Stream-Verarbeitungsanwendung heraufgestuft werden. Die Bibliothek nutzt AWS IoT TwinMaker Komponenten, um Daten aus Ihrem Workspace abzurufen.

Die AWS IoT TwinMaker Flink-Bibliothek erfordert Folgendes.

Voraussetzungen

1. Ein vollständig ausgefüllter Workspace mit Szenen und Komponenten. Verwenden Sie die integrierten Komponententypen für Daten von AWS Diensten (AWS IoT SiteWise und Kinesis Video Streams). Erstellen Sie benutzerdefinierte Komponententypen für Daten aus Quellen von Drittanbietern. Weitere Informationen finden Sie unter ???.
2. Ein Verständnis von Studio-Notebooks mit Managed Service für Apache Flink für Apache Flink. Diese Notebooks basieren auf Apache Zeppelin und verwenden das Apache Flink-Framework. Weitere Informationen finden Sie unter Verwenden eines Studio-Notebooks mit Managed Service für Apache Flink für Apache Flink.

Anweisungen zur Verwendung der Bibliothek finden Sie im Benutzerhandbuch zur AWS IoT TwinMaker Flink-Bibliothek.

Anweisungen zur Einrichtung AWS IoT TwinMaker mit dem Schnellstart in den AWS IoT TwinMaker Beispielen finden Sie in der README-Datei für die Beispielanwendung Insights.

Einloggen und Überwachen AWS IoT TwinMaker

Die Überwachung ist ein wichtiger Bestandteil der Aufrechterhaltung der Zuverlässigkeit, Verfügbarkeit und Leistung Ihrer AWS IoT TwinMaker anderen AWS Lösungen. AWS IoT TwinMaker unterstützt die folgenden Überwachungstools, um den Service zu überwachen, zu melden, wenn etwas nicht stimmt, und gegebenenfalls automatische Maßnahmen zu ergreifen:

- Amazon CloudWatch überwacht in Echtzeit Ihre AWS Ressourcen und die Anwendungen, auf denen Sie laufen AWS. Sie können Kennzahlen sammeln und verfolgen, benutzerdefinierte Dashboards erstellen und Alarme einrichten, die Sie benachrichtigen oder Maßnahmen ergreifen, wenn eine bestimmte Metrik einen von Ihnen festgelegten Schwellenwert erreicht. Sie können beispielsweise die CPU-Auslastung oder andere Kennzahlen für Ihre EC2 Amazon-Instances CloudWatch verfolgen und bei Bedarf automatisch neue Instances starten. Weitere Informationen finden Sie im CloudWatch Amazon-Benutzerhandbuch.
- Amazon CloudWatch Logs überwacht, speichert und bietet Zugriff auf Ihre Protokolldateien über AWS IoT TwinMaker Gateways und andere Quellen. CloudTrail CloudWatch Logs kann Informationen in den Protokolldateien überwachen und Sie benachrichtigen, wenn bestimmte Schwellenwerte erreicht werden. Sie können Ihre Protokolldaten auch in einem sehr robusten Speicher archivieren. Weitere Informationen finden Sie im Amazon CloudWatch Logs-Benutzerhandbuch.
- AWS CloudTrailerfasst API-Aufrufe und zugehörige Ereignisse, die von oder im Namen Ihres AWS Kontos getätigt wurden, und übermittelt die Protokolldateien an einen von Ihnen angegebenen Amazon S3 S3-Bucket. Sie können feststellen, welche Benutzer und Konten angerufen wurden AWS, von welcher Quell-IP-Adresse aus die Anrufe getätigt wurden und wann die Aufrufe erfolgten. Weitere Informationen finden Sie im AWS CloudTrail -Benutzerhandbuch.

Themen

- Überwachung AWS IoT TwinMaker mit CloudWatch Amazon-Metriken
- AWS IoT TwinMaker API-Aufrufe protokollieren mit AWS CloudTrail

Überwachung AWS IoT TwinMaker mit CloudWatch Amazon-Metriken

Sie können dies mithilfe AWS IoT TwinMaker CloudWatch von überwachen. Dabei werden Rohdaten gesammelt und zu lesbaren Metriken verarbeitet, die nahezu in Echtzeit verfügbar sind. Diese Statistiken werden 15 Monate gespeichert, damit Sie auf Verlaufsinformationen zugreifen können und einen besseren Überblick darüber erhalten, wie Ihre Webanwendung oder der Service ausgeführt werden. Sie können auch Alarme einrichten, die auf bestimmte Grenzwerte achten und Benachrichtigungen senden oder Aktivitäten auslösen, wenn diese Grenzwerte erreicht werden. Weitere Informationen finden Sie im CloudWatch Amazon-Benutzerhandbuch.

AWS IoT TwinMaker veröffentlicht die in den folgenden Abschnitten aufgelisteten Metriken und Dimensionen im `AWS/IoTTwinMaker` Namespace.

> **Tip**
>
> AWS IoT TwinMaker veröffentlicht Metriken in einem Intervall von einer Minute. Wenn Sie diese Metriken in der CloudWatch Konsole grafisch anzeigen, empfehlen wir Ihnen, einen Zeitraum von 1 Minute zu wählen, um die höchste verfügbare Auflösung Ihrer Metrikdaten zu sehen.

Inhalt

- Metriken

Metriken

AWS IoT TwinMaker veröffentlicht die folgenden Metriken.

Metriken

Metrik	Beschreibung
`ComponentTypeCreationFailure`	Diese Metrik gibt an, ob die Erstellung des Komponententyps erfolgreich war. Die Metrik wird veröffentlicht, wenn sich ein Komponententyp im `CREATING` Status befindet.

Metrik	Beschreibung
	Dies passiert, wenn ein Komponententyp mit den erforderlichen Eigenschaften im Schemaini tialisierer erstellt wird und diese Eigenschaften mit Standardwerten instanziiert werden. Der Metrikwert kann entweder `0` für Erfolg oder für Misserfolg stehen. 1 Abmessungen: ComponentTypeId, Workspace Id. Einheiten: Anzahl
`ComponentTypeUpdateFailure`	Diese Metrik gibt an, ob das Komponententyp-Update erfolgreich war. Die Metrik wird veröffentlicht, wenn sich ein Komponententyp im `UPDATING` Status befindet. Dies passiert, wenn ein Komponententyp mit den erforderlichen Eigenschaften im Schemaini tialisierer aktualisiert wird und diese Eigenscha ften mit Standardwerten instanziiert werden. Der Metrikwert kann entweder `0` für Erfolg oder für Misserfolg stehen. 1 Abmessungen: ComponentTypeId, Workspace Id. Einheiten: Anzahl

Metrik	Beschreibung
`EntityCreationFailure`	Diese Metrik gibt an, ob die Entitätserstellung erfolgreich war. Die Metrik wird veröffentlicht, wenn sich eine Entität im `CREATING` Status befindet. Dies passiert, wenn eine Entität mit einer Komponente erstellt wird. Der Metrikwert kann entweder `0` für Erfolg oder 1 für Misserfolg stehen. Abmessungen: EntityName, EntityId, WorkspaceId. Einheiten: Anzahl
`EntityUpdateFailure`	Diese Metrik gibt an, ob die Entitätsaktualisierung erfolgreich war. Die Metrik wird veröffentlicht, wenn sich eine Entität im `UPDATING` Status befindet. Dies passiert, wenn eine Entität aktualisiert wird. Der Metrikwert kann entweder `0` für Erfolg oder 1 für Misserfolg stehen. Abmessungen: EntityName, EntityId, WorkspaceId. Einheiten: Anzahl

Metrik	Beschreibung
`EntityDeletionFailure`	Diese Metrik gibt an, ob das Löschen der Entität erfolgreich war. Die Metrik wird veröffentlicht, wenn sich eine Entität im `DELETING` Status befindet. Dies passiert, wenn eine Entität gelöscht wird. Der Metrikwert kann entweder `0` für Erfolg oder 1 für Misserfolg stehen. Abmessungen: EntityName, EntityId, WorkspaceId. Einheiten: Anzahl

> **Tip**
>
> Alle Metriken werden im `AWS/IoTTwinMaker` Namespace veröffentlicht.

AWS IoT TwinMaker API-Aufrufe protokollieren mit AWS CloudTrail

AWS IoT TwinMaker ist in einen Dienst integriert AWS CloudTrail, der eine Aufzeichnung der Aktionen bereitstellt, die von einem Benutzer, einer Rolle oder einem AWS Dienst in ausgeführt wurden AWS IoT TwinMaker. CloudTrail erfasst API-Aufrufe AWS IoT TwinMaker als Ereignisse. Zu den erfassten Aufrufen gehören Aufrufe von der AWS IoT TwinMaker Konsole und Codeaufrufen für die AWS IoT TwinMaker API-Operationen. Wenn Sie einen Trail erstellen, können Sie die kontinuierliche Bereitstellung von CloudTrail Ereignissen an einen Amazon S3 S3-Bucket aktivieren, einschließlich Ereignissen für AWS IoT TwinMaker. Wenn Sie keinen Trail konfigurieren, können Sie die neuesten Ereignisse trotzdem in der CloudTrail Konsole im Ereignisverlauf anzeigen. Anhand der von gesammelten Informationen können Sie die Anfrage ermitteln CloudTrail, an die die Anfrage gestellt wurde AWS IoT TwinMaker, die IP-Adresse, von der aus die Anfrage gestellt wurde, wer die Anfrage gestellt hat, wann sie gestellt wurde, und weitere Informationen.

Weitere Informationen zu CloudTrail finden Sie im AWS CloudTrail Benutzerhandbuch.

AWS IoT TwinMaker Informationen in CloudTrail

Wenn Sie Ihr AWS Konto erstellen, CloudTrail wird es automatisch aktiviert. CloudTrail zeichnet die Aktivitäten von Supportereignissen auf AWS IoT TwinMaker, die in zusammen mit anderen AWS Serviceereignissen im Ereignisverlauf auftreten. Sie können aktuelle Ereignisse in Ihrem AWS Konto anzeigen, suchen und herunterladen. Weitere Informationen finden Sie unter Ereignisse mit CloudTrail Ereignisverlauf anzeigen.

Für eine fortlaufende Aufzeichnung der Ereignisse in Ihrem AWS Konto, einschließlich der Ereignisse für AWS IoT TwinMaker, erstellen Sie einen Trail. Ein Trail ermöglicht CloudTrail die Übermittlung von Protokolldateien an einen Amazon S3 S3-Bucket. Wenn Sie einen Trail in der Konsole erstellen, gilt der Trail standardmäßig für alle AWS Regionen. CloudTrailprotokolliert Ereignisse aus allen Regionen der AWS Partition und übermittelt die Protokolldateien an den von Ihnen angegebenen Amazon S3 S3-Bucket. Darüber hinaus können Sie andere AWS Dienste konfigurieren, um die in den CloudTrail Protokollen gesammelten Ereignisdaten weiter zu analysieren und darauf zu reagieren. Weitere Informationen finden Sie hier:

- Übersicht zum Erstellen eines Trails
- CloudTrail unterstützte Dienste und Integrationen
- Konfiguration von Amazon SNS SNS-Benachrichtigungen für CloudTrail
- Empfangen von CloudTrail Protokolldateien aus mehreren Regionen und Empfangen von CloudTrail Protokolldateien von mehreren Konten

Die meisten AWS IoT TwinMaker Vorgänge werden von der AWS IoT TwinMaker API-Referenz protokolliert CloudTrail und sind in dieser dokumentiert.

Die folgenden Datenebenenoperationen werden nicht protokolliert von CloudTrail:

- GetPropertyValue
- GetPropertyValueHistory
- BatchPutPropertyValues

Jeder Ereignis- oder Protokolleintrag enthält Informationen zu dem Benutzer, der die Anforderung generiert hat. Die Identitätsinformationen unterstützen Sie bei der Ermittlung der folgenden Punkte:

- Ob die Anforderung mit Root- oder -Benutzeranmeldeinformationen ausgeführt wurde.

- Gibt an, ob die Anforderung mit temporären Sicherheitsanmeldeinformationen für eine Rolle oder einen Verbundbenutzer gesendet wurde.
- Ob die Anfrage von einem anderen AWS Dienst gestellt wurde.

Weitere Informationen finden Sie unter CloudTrail -Element userIdentity.

Sicherheit in AWS IoT TwinMaker

Cloud-Sicherheit AWS hat höchste Priorität. Als AWS Kunde profitieren Sie von Rechenzentren und Netzwerkarchitekturen, die darauf ausgelegt sind, die Anforderungen der sicherheitssensibelsten Unternehmen zu erfüllen.

Sicherheit ist eine gemeinsame AWS Verantwortung von Ihnen und Ihnen. Das Modell der geteilten Verantwortung beschreibt dies als Sicherheit der Cloud selbst und Sicherheit in der Cloud:

- Sicherheit der Cloud — AWS ist verantwortlich für den Schutz der Infrastruktur, auf der AWS Dienste in der ausgeführt AWS Cloud werden. AWS bietet Ihnen auch Dienste, die Sie sicher nutzen können. Externe Prüfer testen und verifizieren regelmäßig die Wirksamkeit unserer Sicherheitsmaßnahmen im Rahmen der AWS . Weitere Informationen zu den Compliance-Programmen, die für gelten AWS IoT TwinMaker, finden Sie unter AWS Services im Umfang nach Compliance-Programmen AWS .
- Sicherheit in der Cloud — Ihre Verantwortung richtet sich nach dem AWS Dienst, den Sie nutzen. Sie sind auch für andere Faktoren verantwortlich, etwa für die Vertraulichkeit Ihrer Daten, für die Anforderungen Ihres Unternehmens und für die geltenden Gesetze und Vorschriften.

Diese Dokumentation hilft Ihnen zu verstehen, wie Sie das Modell der gemeinsamen Verantwortung bei der Nutzung anwenden können AWS IoT TwinMaker. In den folgenden Themen erfahren Sie, wie Sie die Konfiguration vornehmen AWS IoT TwinMaker , um Ihre Sicherheits- und Compliance-Ziele zu erreichen. Sie erfahren auch, wie Sie andere AWS Dienste nutzen können, die Sie bei der Überwachung und Sicherung Ihrer AWS IoT TwinMaker Ressourcen unterstützen.

Themen

- Datenschutz in AWS IoT TwinMaker
- Identity and Access Management für AWS IoT TwinMaker
- AWS IoT TwinMaker und Schnittstellen-VPC-Endpunkte ()AWS PrivateLink
- Konformitätsprüfung für AWS IoT TwinMaker
- Resilienz in AWS IoT TwinMaker
- Sicherheit der Infrastruktur in AWS IoT TwinMaker

Datenschutz in AWS IoT TwinMaker

Das Modell der AWS gemeinsamen Verantwortung und geteilter Verantwortung gilt für den Datenschutz in AWS IoT TwinMaker. Wie in diesem Modell beschrieben, AWS ist verantwortlich für den Schutz der globalen Infrastruktur, auf der alle Systeme laufen AWS Cloud. Sie sind dafür verantwortlich, die Kontrolle über Ihre in dieser Infrastruktur gehosteten Inhalte zu behalten. Sie sind auch für die Sicherheitskonfiguration und die Verwaltungsaufgaben für die von Ihnen verwendeten AWS-Services verantwortlich. Weitere Informationen zum Datenschutz finden Sie unter Häufig gestellte Fragen zum Datenschutz. Informationen zum Datenschutz in Europa finden Sie im Blog-Beitrag AWS -Modell der geteilten Verantwortung und in der DSGVO im AWS -Sicherheitsblog.

Aus Datenschutzgründen empfehlen wir, dass Sie AWS-Konto Anmeldeinformationen schützen und einzelne Benutzer mit AWS IAM Identity Center oder AWS Identity and Access Management (IAM) einrichten. So erhält jeder Benutzer nur die Berechtigungen, die zum Durchführen seiner Aufgaben erforderlich sind. Außerdem empfehlen wir, die Daten mit folgenden Methoden schützen:

- Verwenden Sie für jedes Konto die Multi-Faktor-Authentifizierung (MFA).
- Wird verwendet SSL/TLS , um mit AWS Ressourcen zu kommunizieren. Wir benötigen TLS 1.2 und empfehlen TLS 1.3.
- Richten Sie die API und die Protokollierung von Benutzeraktivitäten mit ein AWS CloudTrail. Informationen zur Verwendung von CloudTrail Pfaden zur Erfassung von AWS Aktivitäten finden Sie unter Arbeiten mit CloudTrail Pfaden im AWS CloudTrail Benutzerhandbuch.
- Verwenden Sie AWS Verschlüsselungslösungen zusammen mit allen darin enthaltenen Standardsicherheitskontrollen AWS-Services.
- Verwenden Sie erweiterte verwaltete Sicherheitsservices wie Amazon Macie, die dabei helfen, in Amazon S3 gespeicherte persönliche Daten zu erkennen und zu schützen.
- Wenn Sie für den Zugriff AWS über eine Befehlszeilenschnittstelle oder eine API FIPS 140-3-validierte kryptografische Module benötigen, verwenden Sie einen FIPS-Endpunkt. Weitere Informationen über verfügbare FIPS-Endpunkte finden Sie unter Federal Information Processing Standard (FIPS) 140-3.

Wir empfehlen dringend, in Freitextfeldern, z. B. im Feld Name, keine vertraulichen oder sensiblen Informationen wie die E-Mail-Adressen Ihrer Kunden einzugeben. Dies gilt auch, wenn Sie mit der Konsole, der AWS IoT TwinMaker API oder auf andere AWS-Services Weise arbeiten oder diese verwenden. AWS CLI AWS SDKs Alle Daten, die Sie in Tags oder Freitextfelder eingeben, die für Namen verwendet werden, können für Abrechnungs- oder Diagnoseprotokolle verwendet

werden. Wenn Sie eine URL für einen externen Server bereitstellen, empfehlen wir dringend, keine Anmeldeinformationen zur Validierung Ihrer Anforderung an den betreffenden Server in die URL einzuschließen.

Verschlüsselung im Ruhezustand

AWS IoT TwinMaker speichert Ihre Workspace-Informationen in einem Amazon S3 S3-Bucket, den der Service für Sie erstellt, falls Sie dies wünschen. In dem Bucket, den der Service für Sie erstellt, ist die serverseitige Standardverschlüsselung aktiviert. Wenn Sie beim Erstellen eines neuen Workspace Ihren eigenen Amazon S3 S3-Bucket verwenden möchten, empfehlen wir Ihnen, die serverseitige Standardverschlüsselung zu aktivieren. Weitere Informationen zur Standardverschlüsselung in Amazon S3 finden Sie unter Einstellung des standardmäßigen serverseitigen Verschlüsselungsverhaltens für Amazon S3 S3-Buckets.

Verschlüsselung während der Übertragung

Alle an AWS IoT TwinMaker gesendeten Daten werden über eine TLS-Verbindung unter Verwendung des HTTPS-Protokolls gesendet, sodass sie während der Übertragung standardmäßig sicher sind.

> **Note**
>
> Wir empfehlen, HTTPS für Amazon S3 S3-Bucket-Adressen als Kontrolle zu verwenden, um bei der AWS IoT TwinMaker Interaktion mit einem Amazon S3 S3-Bucket die Verschlüsselung während der Übertragung zu erzwingen. Weitere Informationen zu Amazon S3 S3-Buckets finden Sie unter Amazon S3 S3-Buckets erstellen, konfigurieren und damit arbeiten.

Identity and Access Management für AWS IoT TwinMaker

AWS Identity and Access Management (IAM) hilft einem Administrator AWS-Service , den Zugriff auf Ressourcen sicher zu AWS kontrollieren. IAM-Administratoren kontrollieren, wer authentifiziert (angemeldet) und autorisiert werden kann (über Berechtigungen verfügt), um Ressourcen zu verwenden. AWS IoT TwinMaker IAM ist ein Programm AWS-Service , das Sie ohne zusätzliche Kosten nutzen können.

Themen

- Zielgruppe
- Authentifizierung mit Identitäten
- Verwalten des Zugriffs mit Richtlinien
- Wie AWS IoT TwinMaker funktioniert mit IAM
- Beispiele für identitätsbasierte Richtlinien für AWS IoT TwinMaker
- Problembehandlung bei AWS IoT TwinMaker Identität und Zugriff
- Verwenden von serviceverknüpften Rollen für AWS IoT TwinMaker
- AWS verwaltete Richtlinien für AWS IoT TwinMaker

Zielgruppe

Die Art und Weise, wie Sie AWS Identity and Access Management (IAM) verwenden, hängt von der Arbeit ab, in der Sie tätig sind. AWS IoT TwinMaker

Dienstbenutzer — Wenn Sie den AWS IoT TwinMaker Dienst für Ihre Arbeit verwenden, stellt Ihnen Ihr Administrator die erforderlichen Anmeldeinformationen und Berechtigungen zur Verfügung. Wenn Sie für Ihre Arbeit mehr AWS IoT TwinMaker Funktionen verwenden, benötigen Sie möglicherweise zusätzliche Berechtigungen. Wenn Sie die Funktionsweise der Zugriffskontrolle nachvollziehen, wissen Sie bereits, welche Berechtigungen Sie von Ihrem Administrator anfordern müssen. Unter Problembehandlung bei AWS IoT TwinMaker Identität und Zugriff finden Sie nützliche Informationen für den Fall, dass Sie keinen Zugriff auf eine Feature in AWS IoT TwinMaker haben.

Serviceadministrator — Wenn Sie in Ihrem Unternehmen für die AWS IoT TwinMaker Ressourcen verantwortlich sind, haben Sie wahrscheinlich vollen Zugriff auf AWS IoT TwinMaker. Es ist Ihre Aufgabe, zu bestimmen, auf welche AWS IoT TwinMaker Funktionen und Ressourcen Ihre Servicebenutzer zugreifen sollen. Anschließend müssen Sie Anforderungen an Ihren IAM-Administrator senden, um die Berechtigungen der Servicebenutzer zu ändern. Lesen Sie die Informationen auf dieser Seite, um die Grundkonzepte von IAM nachzuvollziehen. Weitere Informationen darüber, wie Ihr Unternehmen IAM nutzen kann AWS IoT TwinMaker, finden Sie unterWie AWS IoT TwinMaker funktioniert mit IAM.

IAM-Administrator: Wenn Sie als IAM-Administrator fungieren, sollten Sie Einzelheiten dazu kennen, wie Sie Richtlinien zur Verwaltung des Zugriffs auf AWS IoT TwinMaker verfassen können. Beispiele für AWS IoT TwinMaker identitätsbasierte Richtlinien, die Sie in IAM verwenden können, finden Sie unter. Beispiele für identitätsbasierte Richtlinien für AWS IoT TwinMaker

Authentifizierung mit Identitäten

Authentifizierung ist die Art und Weise, wie Sie sich AWS mit Ihren Identitätsdaten anmelden. Sie müssen als IAM-Benutzer authentifiziert (angemeldet AWS) sein oder eine IAM-Rolle annehmen. Root-Benutzer des AWS-Kontos

Sie können sich AWS als föderierte Identität anmelden, indem Sie Anmeldeinformationen verwenden, die über eine Identitätsquelle bereitgestellt wurden. AWS IAM Identity Center (IAM Identity Center) -Benutzer, die Single Sign-On-Authentifizierung Ihres Unternehmens und Ihre Google- oder Facebook-Anmeldeinformationen sind Beispiele für föderierte Identitäten. Wenn Sie sich als Verbundidentität anmelden, hat der Administrator vorher mithilfe von IAM-Rollen einen Identitätsverbund eingerichtet. Wenn Sie über den Verbund darauf zugreifen AWS , übernehmen Sie indirekt eine Rolle.

Je nachdem, welcher Benutzertyp Sie sind, können Sie sich beim AWS Management Console oder beim AWS Zugangsportal anmelden. Weitere Informationen zur Anmeldung finden Sie AWS unter So melden Sie sich bei Ihrem an AWS-Konto im AWS-Anmeldung Benutzerhandbuch.

Wenn Sie AWS programmgesteuert darauf zugreifen, AWS stellt es ein Software Development Kit (SDK) und eine Befehlszeilenschnittstelle (CLI) bereit, um Ihre Anfragen mithilfe Ihrer Anmeldeinformationen kryptografisch zu signieren. Wenn Sie keine AWS Tools verwenden, müssen Sie Anfragen selbst signieren. Weitere Informationen zur Verwendung der empfohlenen Methode für die Selbstsignierung von Anforderungen finden Sie unter AWS Signature Version 4 für API-Anforderungen im IAM-Benutzerhandbuch.

Unabhängig von der verwendeten Authentifizierungsmethode müssen Sie möglicherweise zusätzliche Sicherheitsinformationen bereitstellen. AWS Empfiehlt beispielsweise, die Multi-Faktor-Authentifizierung (MFA) zu verwenden, um die Sicherheit Ihres Kontos zu erhöhen. Weitere Informationen finden Sie unter Multi-Faktor-Authentifizierung im AWS IAM Identity Center - Benutzerhandbuch und AWS Multi-Faktor-Authentifizierung (MFA) in IAM im IAM-Benutzerhandbuch.

AWS-Konto Root-Benutzer

Wenn Sie einen erstellen AWS-Konto, beginnen Sie mit einer Anmeldeidentität, die vollständigen Zugriff auf alle AWS-Services Ressourcen im Konto hat. Diese Identität wird als AWS-Konto Root-Benutzer bezeichnet. Der Zugriff erfolgt, indem Sie sich mit der E-Mail-Adresse und dem Passwort anmelden, mit denen Sie das Konto erstellt haben. Wir raten ausdrücklich davon ab, den Root-Benutzer für Alltagsaufgaben zu verwenden. Schützen Sie Ihre Root-Benutzer-Anmeldeinformationen. Verwenden Sie diese nur, um die Aufgaben auszuführen, die nur der Root-

Benutzer ausführen kann. Eine vollständige Liste der Aufgaben, für die Sie sich als Root-Benutzer anmelden müssen, finden Sie unter Aufgaben, die Root-Benutzer-Anmeldeinformationen erfordern im IAM-Benutzerhandbuch.

Verbundidentität

Als bewährte Methode sollten menschliche Benutzer, einschließlich Benutzer, die Administratorzugriff benötigen, für den Zugriff AWS-Services mithilfe temporärer Anmeldeinformationen den Verbund mit einem Identitätsanbieter verwenden.

Eine föderierte Identität ist ein Benutzer aus Ihrem Unternehmensbenutzerverzeichnis, einem Web-Identitätsanbieter AWS Directory Service, dem Identity Center-Verzeichnis oder einem beliebigen Benutzer, der mithilfe AWS-Services von Anmeldeinformationen zugreift, die über eine Identitätsquelle bereitgestellt wurden. Wenn föderierte Identitäten darauf zugreifen AWS-Konten, übernehmen sie Rollen, und die Rollen stellen temporäre Anmeldeinformationen bereit.

Für die zentrale Zugriffsverwaltung empfehlen wir Ihnen, AWS IAM Identity Center zu verwenden. Sie können Benutzer und Gruppen in IAM Identity Center erstellen, oder Sie können eine Verbindung zu einer Gruppe von Benutzern und Gruppen in Ihrer eigenen Identitätsquelle herstellen und diese synchronisieren, um sie in all Ihren AWS-Konten Anwendungen zu verwenden. Informationen zu IAM Identity Center finden Sie unter Was ist IAM Identity Center? im AWS IAM Identity Center - Benutzerhandbuch.

IAM-Benutzer und -Gruppen

Ein IAM-Benutzer ist eine Identität innerhalb Ihres Unternehmens AWS-Konto , die über spezifische Berechtigungen für eine einzelne Person oder Anwendung verfügt. Wenn möglich, empfehlen wir, temporäre Anmeldeinformationen zu verwenden, anstatt IAM-Benutzer zu erstellen, die langfristige Anmeldeinformationen wie Passwörter und Zugriffsschlüssel haben. Bei speziellen Anwendungsfällen, die langfristige Anmeldeinformationen mit IAM-Benutzern erfordern, empfehlen wir jedoch, die Zugriffsschlüssel zu rotieren. Weitere Informationen finden Sie unter Regelmäßiges Rotieren von Zugriffsschlüsseln für Anwendungsfälle, die langfristige Anmeldeinformationen erfordern im IAM-Benutzerhandbuch.

Eine IAM-Gruppe ist eine Identität, die eine Sammlung von IAM-Benutzern angibt. Sie können sich nicht als Gruppe anmelden. Mithilfe von Gruppen können Sie Berechtigungen für mehrere Benutzer gleichzeitig angeben. Gruppen vereinfachen die Verwaltung von Berechtigungen, wenn es zahlreiche Benutzer gibt. Sie könnten beispielsweise eine Gruppe benennen IAMAdminsund dieser Gruppe Berechtigungen zur Verwaltung von IAM-Ressourcen erteilen.

Benutzer unterscheiden sich von Rollen. Ein Benutzer ist einer einzigen Person oder Anwendung eindeutig zugeordnet. Eine Rolle kann von allen Personen angenommen werden, die sie benötigen. Benutzer besitzen dauerhafte Anmeldeinformationen. Rollen stellen temporäre Anmeldeinformationen bereit. Weitere Informationen finden Sie unter Anwendungsfälle für IAM-Benutzer im IAM-Benutzerhandbuch.

IAM-Rollen

Eine IAM-Rolle ist eine Identität innerhalb von Ihnen AWS-Konto , die über bestimmte Berechtigungen verfügt. Sie ist einem IAM-Benutzer vergleichbar, jedoch nicht mit einer bestimmten Person verknüpft. Um vorübergehend eine IAM-Rolle in der zu übernehmen AWS Management Console, können Sie von einer Benutzer- zu einer IAM-Rolle (Konsole) wechseln. Sie können eine Rolle übernehmen, indem Sie eine AWS CLI oder AWS API-Operation aufrufen oder eine benutzerdefinierte URL verwenden. Weitere Informationen zu Methoden für die Verwendung von Rollen finden Sie unter Methoden für die Übernahme einer Rolle im IAM-Benutzerhandbuch.

IAM-Rollen mit temporären Anmeldeinformationen sind in folgenden Situationen hilfreich:

- Verbundbenutzerzugriff – Um einer Verbundidentität Berechtigungen zuzuweisen, erstellen Sie eine Rolle und definieren Berechtigungen für die Rolle. Wird eine Verbundidentität authentifiziert, so wird die Identität der Rolle zugeordnet und erhält die von der Rolle definierten Berechtigungen. Informationen zu Rollen für den Verbund finden Sie unter Erstellen von Rollen für externe Identitätsanbieter (Verbund) im IAM-Benutzerhandbuch. Wenn Sie IAM Identity Center verwenden, konfigurieren Sie einen Berechtigungssatz. Wenn Sie steuern möchten, worauf Ihre Identitäten nach der Authentifizierung zugreifen können, korreliert IAM Identity Center den Berechtigungssatz mit einer Rolle in IAM. Informationen zu Berechtigungssätzen finden Sie unter Berechtigungssätze im AWS IAM Identity Center -Benutzerhandbuch.
- Temporäre IAM-Benutzerberechtigungen – Ein IAM-Benutzer oder eine -Rolle kann eine IAM-Rolle übernehmen, um vorübergehend andere Berechtigungen für eine bestimmte Aufgabe zu erhalten.
- Kontoübergreifender Zugriff – Sie können eine IAM-Rolle verwenden, um einem vertrauenswürdigen Prinzipal in einem anderen Konto den Zugriff auf Ressourcen in Ihrem Konto zu ermöglichen. Rollen stellen die primäre Möglichkeit dar, um kontoübergreifendem Zugriff zu gewähren. Bei einigen können Sie AWS-Services jedoch eine Richtlinie direkt an eine Ressource anhängen (anstatt eine Rolle als Proxy zu verwenden). Informationen zu den Unterschieden zwischen Rollen und ressourcenbasierten Richtlinien für den kontoübergreifenden Zugriff finden Sie unter Kontoübergreifender Ressourcenzugriff in IAM im IAM-Benutzerhandbuch.

- Serviceübergreifender Zugriff — Einige AWS-Services verwenden Funktionen in anderen AWS-Services. Wenn Sie beispielsweise in einem Service einen Anruf tätigen, ist es üblich, dass dieser Service Anwendungen in Amazon ausführt EC2 oder Objekte in Amazon S3 speichert. Ein Dienst kann dies mit den Berechtigungen des aufrufenden Prinzipals mit einer Servicerolle oder mit einer serviceverknüpften Rolle tun.
  - Forward Access Sessions (FAS) — Wenn Sie einen IAM-Benutzer oder eine IAM-Rolle verwenden, um Aktionen auszuführen AWS, gelten Sie als Principal. Bei einigen Services könnte es Aktionen geben, die dann eine andere Aktion in einem anderen Service initiieren. FAS verwendet die Berechtigungen des Prinzipals, der einen aufruft AWS-Service, in Kombination mit der Anfrage, Anfragen an AWS-Service nachgelagerte Dienste zu stellen. FAS-Anfragen werden nur gestellt, wenn ein Dienst eine Anfrage erhält, für deren Abschluss Interaktionen mit anderen AWS-Services oder Ressourcen erforderlich sind. In diesem Fall müssen Sie über Berechtigungen zum Ausführen beider Aktionen verfügen. Einzelheiten zu den Richtlinien für FAS-Anfragen finden Sie unter Zugriffssitzungen weiterleiten.
  - Servicerolle – Eine Servicerolle ist eine IAM-Rolle, die ein Service übernimmt, um Aktionen in Ihrem Namen auszuführen. Ein IAM-Administrator kann eine Servicerolle innerhalb von IAM erstellen, ändern und löschen. Weitere Informationen finden Sie unter Erstellen einer Rolle zum Delegieren von Berechtigungen an einen AWS-Service im IAM-Benutzerhandbuch.
  - Dienstbezogene Rolle — Eine dienstbezogene Rolle ist eine Art von Servicerolle, die mit einer verknüpft ist. AWS-Service Der Service kann die Rolle übernehmen, um eine Aktion in Ihrem Namen auszuführen. Servicebezogene Rollen erscheinen in Ihrem Dienst AWS-Konto und gehören dem Dienst. Ein IAM-Administrator kann die Berechtigungen für Service-verknüpfte Rollen anzeigen, aber nicht bearbeiten.
- Auf Amazon ausgeführte Anwendungen EC2 — Sie können eine IAM-Rolle verwenden, um temporäre Anmeldeinformationen für Anwendungen zu verwalten, die auf einer EC2 Instance ausgeführt werden und AWS API-Anfragen stellen AWS CLI . Dies ist dem Speichern von Zugriffsschlüsseln innerhalb der EC2 Instance vorzuziehen. Um einer EC2 Instanz eine AWS Rolle zuzuweisen und sie allen ihren Anwendungen zur Verfügung zu stellen, erstellen Sie ein Instanzprofil, das an die Instanz angehängt ist. Ein Instanzprofil enthält die Rolle und ermöglicht Programmen, die auf der EC2 Instanz ausgeführt werden, temporäre Anmeldeinformationen abzurufen. Weitere Informationen finden Sie im IAM-Benutzerhandbuch unter Verwenden einer IAM-Rolle, um Berechtigungen für Anwendungen zu gewähren, die auf EC2 Amazon-Instances ausgeführt werden.

Verwalten des Zugriffs mit Richtlinien

Sie kontrollieren den Zugriff, AWS indem Sie Richtlinien erstellen und diese an AWS Identitäten oder Ressourcen anhängen. Eine Richtlinie ist ein Objekt, AWS das, wenn es einer Identität oder Ressource zugeordnet ist, deren Berechtigungen definiert. AWS wertet diese Richtlinien aus, wenn ein Prinzipal (Benutzer, Root-Benutzer oder Rollensitzung) eine Anfrage stellt. Die Berechtigungen in den Richtlinien legen fest, ob eine Anforderung zugelassen oder abgelehnt wird. Die meisten Richtlinien werden AWS als JSON-Dokumente gespeichert. Weitere Informationen zu Struktur und Inhalten von JSON-Richtliniendokumenten finden Sie unter Übersicht über JSON-Richtlinien im IAM-Benutzerhandbuch.

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer auf was Zugriff hat. Das heißt, welcher Prinzipal Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen kann.

Standardmäßig haben Benutzer, Gruppen und Rollen keine Berechtigungen. Ein IAM-Administrator muss IAM-Richtlinien erstellen, die Benutzern die Berechtigung erteilen, Aktionen für die Ressourcen auszuführen, die sie benötigen. Der Administrator kann dann die IAM-Richtlinien zu Rollen hinzufügen, und Benutzer können die Rollen annehmen.

IAM-Richtlinien definieren Berechtigungen für eine Aktion unabhängig von der Methode, die Sie zur Ausführung der Aktion verwenden. Angenommen, es gibt eine Richtlinie, die Berechtigungen für die `iam:GetRole`-Aktion erteilt. Ein Benutzer mit dieser Richtlinie kann Rolleninformationen von der AWS Management Console AWS CLI, der oder der AWS API abrufen.

Identitätsbasierte Richtlinien

Identitätsbasierte Richtlinien sind JSON-Berechtigungsrichtliniendokumente, die Sie einer Identität anfügen können, wie z. B. IAM-Benutzern, -Benutzergruppen oder -Rollen. Diese Richtlinien steuern, welche Aktionen die Benutzer und Rollen für welche Ressourcen und unter welchen Bedingungen ausführen können. Informationen zum Erstellen identitätsbasierter Richtlinien finden Sie unter Definieren benutzerdefinierter IAM-Berechtigungen mit vom Kunden verwalteten Richtlinien im IAM-Benutzerhandbuch.

Identitätsbasierte Richtlinien können weiter als Inline-Richtlinien oder verwaltete Richtlinien kategorisiert werden. Inline-Richtlinien sind direkt in einen einzelnen Benutzer, eine einzelne Gruppe oder eine einzelne Rolle eingebettet. Verwaltete Richtlinien sind eigenständige Richtlinien, die Sie mehreren Benutzern, Gruppen und Rollen in Ihrem System zuordnen können AWS-Konto.

Zu den verwalteten Richtlinien gehören AWS verwaltete Richtlinien und vom Kunden verwaltete Richtlinien. Informationen dazu, wie Sie zwischen einer verwalteten Richtlinie und einer Inline-Richtlinie wählen, finden Sie unter Auswählen zwischen verwalteten und eingebundenen Richtlinien im IAM-Benutzerhandbuch.

Ressourcenbasierte Richtlinien

Ressourcenbasierte Richtlinien sind JSON-Richtliniendokumente, die Sie an eine Ressource anfügen. Beispiele für ressourcenbasierte Richtlinien sind IAM-Rollen-Vertrauensrichtlinien und Amazon-S3-Bucket-Richtlinien. In Services, die ressourcenbasierte Richtlinien unterstützen, können Service-Administratoren sie verwenden, um den Zugriff auf eine bestimmte Ressource zu steuern. Für die Ressource, an welche die Richtlinie angehängt ist, legt die Richtlinie fest, welche Aktionen ein bestimmter Prinzipal unter welchen Bedingungen für diese Ressource ausführen kann. Sie müssen in einer ressourcenbasierten Richtlinie einen Prinzipal angeben. Zu den Prinzipalen können Konten, Benutzer, Rollen, Verbundbenutzer oder gehören. AWS-Services

Ressourcenbasierte Richtlinien sind Richtlinien innerhalb dieses Diensts. Sie können AWS verwaltete Richtlinien von IAM nicht in einer ressourcenbasierten Richtlinie verwenden.

Zugriffskontrolllisten () ACLs

Zugriffskontrolllisten (ACLs) steuern, welche Principals (Kontomitglieder, Benutzer oder Rollen) über Zugriffsberechtigungen für eine Ressource verfügen. ACLs ähneln ressourcenbasierten Richtlinien, verwenden jedoch nicht das JSON-Richtliniendokumentformat.

Amazon S3 und Amazon VPC sind Beispiele für Dienste, die Unterstützung ACLs bieten. AWS WAF Weitere Informationen finden Sie unter Übersicht über ACLs die Zugriffskontrollliste (ACL) im Amazon Simple Storage Service Developer Guide.

Weitere Richtlinientypen

AWS unterstützt zusätzliche, weniger verbreitete Richtlinientypen. Diese Richtlinientypen können die maximalen Berechtigungen festlegen, die Ihnen von den häufiger verwendeten Richtlinientypen erteilt werden können.

- Berechtigungsgrenzen – Eine Berechtigungsgrenze ist ein erweitertes Feature, mit der Sie die maximalen Berechtigungen festlegen können, die eine identitätsbasierte Richtlinie einer IAM-Entität (IAM-Benutzer oder -Rolle) erteilen kann. Sie können eine Berechtigungsgrenze für eine Entität festlegen. Die daraus resultierenden Berechtigungen sind der Schnittpunkt der

identitätsbasierten Richtlinien einer Entität und ihrer Berechtigungsgrenzen. Ressourcenbasierte Richtlinien, die den Benutzer oder die Rolle im Feld `Principal` angeben, werden nicht durch Berechtigungsgrenzen eingeschränkt. Eine explizite Zugriffsverweigerung in einer dieser Richtlinien setzt eine Zugriffserlaubnis außer Kraft. Weitere Informationen über Berechtigungsgrenzen finden Sie unter Berechtigungsgrenzen für IAM-Entitäten im IAM-Benutzerhandbuch.

- Dienststeuerungsrichtlinien (SCPs) — SCPs sind JSON-Richtlinien, die die maximalen Berechtigungen für eine Organisation oder Organisationseinheit (OU) in festlegen. AWS Organizations AWS Organizations ist ein Dienst zur Gruppierung und zentralen Verwaltung mehrerer Objekte AWS-Konten , die Ihrem Unternehmen gehören. Wenn Sie alle Funktionen in einer Organisation aktivieren, können Sie Richtlinien zur Servicesteuerung (SCPs) auf einige oder alle Ihre Konten anwenden. Das SCP schränkt die Berechtigungen für Entitäten in Mitgliedskonten ein, einschließlich der einzelnen Root-Benutzer des AWS-Kontos Entitäten. Weitere Informationen zu Organizations und SCPs finden Sie unter Richtlinien zur Servicesteuerung im AWS Organizations Benutzerhandbuch.
- Ressourcenkontrollrichtlinien (RCPs) — RCPs sind JSON-Richtlinien, mit denen Sie die maximal verfügbaren Berechtigungen für Ressourcen in Ihren Konten festlegen können, ohne die IAM-Richtlinien aktualisieren zu müssen, die jeder Ressource zugeordnet sind, deren Eigentümer Sie sind. Das RCP schränkt die Berechtigungen für Ressourcen in Mitgliedskonten ein und kann sich auf die effektiven Berechtigungen für Identitäten auswirken, einschließlich der Root-Benutzer des AWS-Kontos, unabhängig davon, ob sie zu Ihrer Organisation gehören. Weitere Informationen zu Organizations RCPs, einschließlich einer Liste AWS-Services dieser Support-Leistungen RCPs, finden Sie unter Resource Control Policies (RCPs) im AWS Organizations Benutzerhandbuch.
- Sitzungsrichtlinien – Sitzungsrichtlinien sind erweiterte Richtlinien, die Sie als Parameter übergeben, wenn Sie eine temporäre Sitzung für eine Rolle oder einen verbundenen Benutzer programmgesteuert erstellen. Die resultierenden Sitzungsberechtigungen sind eine Schnittmenge der auf der Identität des Benutzers oder der Rolle basierenden Richtlinien und der Sitzungsrichtlinien. Berechtigungen können auch aus einer ressourcenbasierten Richtlinie stammen. Eine explizite Zugriffsverweigerung in einer dieser Richtlinien setzt eine Zugriffserlaubnis außer Kraft. Weitere Informationen finden Sie unter Sitzungsrichtlinien im IAM-Benutzerhandbuch.

Mehrere Richtlinientypen

Wenn mehrere auf eine Anforderung mehrere Richtlinientypen angewendet werden können, sind die entsprechenden Berechtigungen komplizierter. Informationen darüber, wie AWS bestimmt wird, ob eine Anfrage zulässig ist, wenn mehrere Richtlinientypen betroffen sind, finden Sie im IAM-Benutzerhandbuch unter Bewertungslogik für Richtlinien.

Wie AWS IoT TwinMaker funktioniert mit IAM

Bevor Sie IAM zur Verwaltung des Zugriffs auf verwenden, sollten Sie sich darüber informieren AWS IoT TwinMaker, mit welchen IAM-Funktionen Sie arbeiten können. AWS IoT TwinMaker

IAM-Funktionen, die Sie mit verwenden können AWS IoT TwinMaker

IAM-Feature	AWS IoT TwinMaker Unterstützung
Identitätsbasierte Richtlinien	Ja
Ressourcenbasierte Richtlinien	Nein
Richtlinienaktionen	Ja
Richtlinienressourcen	Ja
Bedingungsschlüssel für die Richtlinie	Ja
ACLs	Nein
ABAC (Tags in Richtlinien)	Teilweise
Temporäre Anmeldeinformationen	Ja
Prinzipalberechtigungen	Ja
Servicerollen	Ja
Service-verknüpfte Rollen	Nein

Einen allgemeinen Überblick darüber, wie AWS IoT TwinMaker und andere AWS Dienste mit den meisten IAM-Funktionen funktionieren, finden Sie im AWS IAM Identity Center Benutzerhandbuch unter AWS Dienste, die mit IAM funktionieren.

Identitätsbasierte Richtlinien für AWS IoT TwinMaker

Unterstützt Richtlinien auf Identitätsbasis: Ja

Identitätsbasierte Richtlinien sind JSON-Berechtigungsrichtliniendokumente, die Sie einer Identität anfügen können, wie z. B. IAM-Benutzern, -Benutzergruppen oder -Rollen. Diese Richtlinien steuern, welche Aktionen die Benutzer und Rollen für welche Ressourcen und unter welchen Bedingungen ausführen können. Informationen zum Erstellen identitätsbasierter Richtlinien finden Sie unter Definieren benutzerdefinierter IAM-Berechtigungen mit vom Kunden verwalteten Richtlinien im IAM-Benutzerhandbuch.

Mit identitätsbasierten IAM-Richtlinien können Sie angeben, welche Aktionen und Ressourcen zugelassen oder abgelehnt werden. Darüber hinaus können Sie die Bedingungen festlegen, unter denen Aktionen zugelassen oder abgelehnt werden. Sie können den Prinzipal nicht in einer identitätsbasierten Richtlinie angeben, da er für den Benutzer oder die Rolle gilt, dem er zugeordnet ist. Informationen zu sämtlichen Elementen, die Sie in einer JSON-Richtlinie verwenden, finden Sie in der IAM-Referenz für JSON-Richtlinienelemente im IAM-Benutzerhandbuch.

Beispiele für identitätsbasierte Richtlinien für AWS IoT TwinMaker

Beispiele für AWS IoT TwinMaker identitätsbasierte Richtlinien finden Sie unter. Beispiele für identitätsbasierte Richtlinien für AWS IoT TwinMaker

Ressourcenbasierte Richtlinien finden Sie in AWS IoT TwinMaker

Unterstützt ressourcenbasierte Richtlinien: Nein

Ressourcenbasierte Richtlinien sind JSON-Richtliniendokumente, die Sie an eine Ressource anfügen. Beispiele für ressourcenbasierte Richtlinien sind IAM-Rollen-Vertrauensrichtlinien und Amazon-S3-Bucket-Richtlinien. In Services, die ressourcenbasierte Richtlinien unterstützen, können Service-Administratoren sie verwenden, um den Zugriff auf eine bestimmte Ressource zu steuern. Für die Ressource, an welche die Richtlinie angehängt ist, legt die Richtlinie fest, welche Aktionen ein bestimmter Prinzipal unter welchen Bedingungen für diese Ressource ausführen kann. Sie müssen in einer ressourcenbasierten Richtlinie einen Prinzipal angeben. Zu den Prinzipalen können Konten, Benutzer, Rollen, Verbundbenutzer oder gehören. AWS-Services

Um kontoübergreifenden Zugriff zu ermöglichen, können Sie ein gesamtes Konto oder IAM-Entitäten in einem anderen Konto als Prinzipal in einer ressourcenbasierten Richtlinie angeben. Durch das Hinzufügen eines kontoübergreifenden Auftraggebers zu einer ressourcenbasierten Richtlinie ist nur die halbe Vertrauensbeziehung eingerichtet. Wenn sich der Prinzipal und die Ressource unterscheiden AWS-Konten, muss ein IAM-Administrator des vertrauenswürdigen Kontos auch der Prinzipalentität (Benutzer oder Rolle) die Berechtigung zum Zugriff auf die Ressource erteilen. Sie erteilen Berechtigungen, indem Sie der juristischen Stelle eine identitätsbasierte Richtlinie anfügen.

Wenn jedoch eine ressourcenbasierte Richtlinie Zugriff auf einen Prinzipal in demselben Konto gewährt, ist keine zusätzliche identitätsbasierte Richtlinie erforderlich. Weitere Informationen finden Sie unter Kontoübergreifender Ressourcenzugriff in IAM im IAM-Benutzerhandbuch.

Richtlinienaktionen für AWS IoT TwinMaker

Unterstützt Richtlinienaktionen: Ja

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer Zugriff auf was hat. Das heißt, welcher Prinzipal Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen kann.

Das Element `Action` einer JSON-Richtlinie beschreibt die Aktionen, mit denen Sie den Zugriff in einer Richtlinie zulassen oder verweigern können. Richtlinienaktionen haben normalerweise denselben Namen wie der zugehörige AWS API-Vorgang. Es gibt einige Ausnahmen, z. B. Aktionen, die nur mit Genehmigung durchgeführt werden können und für die es keinen passenden API-Vorgang gibt. Es gibt auch einige Operationen, die mehrere Aktionen in einer Richtlinie erfordern. Diese zusätzlichen Aktionen werden als abhängige Aktionen bezeichnet.

Schließen Sie Aktionen in eine Richtlinie ein, um Berechtigungen zur Durchführung der zugeordneten Operation zu erteilen.

Eine Liste der AWS IoT TwinMaker Aktionen finden Sie unter Aktionen definiert von AWS IoT TwinMaker in der Serviceautorisierungsreferenz.

Bei Richtlinienaktionen wird vor der Aktion das folgende Präfix AWS IoT TwinMaker verwendet:

```
iottwinmaker
```

Um mehrere Aktionen in einer einzigen Anweisung anzugeben, trennen Sie sie mit Kommata:

```
"Action": [
   "iottwinmaker:action1",
   "iottwinmaker:action2"
         ]
```

Beispiele für AWS IoT TwinMaker identitätsbasierte Richtlinien finden Sie unter. Beispiele für identitätsbasierte Richtlinien für AWS IoT TwinMaker

Politische Ressourcen für AWS IoT TwinMaker

Unterstützt Richtlinienressourcen: Ja

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer auf was Zugriff hat. Das heißt, welcher Prinzipal Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen kann.

Das JSON-Richtlinienelement `Resource` gibt die Objekte an, auf welche die Aktion angewendet wird. Anweisungen müssen entweder ein – `Resource`oder ein `NotResource`-Element enthalten. Als bewährte Methode geben Sie eine Ressource mit dem zugehörigen Amazon-Ressourcennamen (ARN) an. Sie können dies für Aktionen tun, die einen bestimmten Ressourcentyp unterstützen, der als Berechtigungen auf Ressourcenebene bezeichnet wird.

Verwenden Sie für Aktionen, die keine Berechtigungen auf Ressourcenebene unterstützen, z. B. Auflistungsoperationen, einen Platzhalter (*), um anzugeben, dass die Anweisung für alle Ressourcen gilt.

```
"Resource": "*"
```

Eine Liste der AWS IoT TwinMaker Ressourcentypen und ihrer ARNs Eigenschaften finden Sie unter Resources defined by AWS IoT TwinMaker in der Service Authorization Reference. Informationen zu den Aktionen, mit denen Sie den ARN einzelner Ressourcen angeben können, finden Sie unter Von AWS IoT TwinMaker definierte Aktionen.

Beispiele für AWS IoT TwinMaker identitätsbasierte Richtlinien finden Sie unter. Beispiele für identitätsbasierte Richtlinien für AWS IoT TwinMaker

Bedingungsschlüssel für Richtlinien für AWS IoT TwinMaker

Unterstützt servicespezifische Richtlinienbedingungsschlüssel: Ja

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer auf was Zugriff hat. Das heißt, welcher Prinzipal kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Das Element `Condition` (oder `Condition` block) ermöglicht Ihnen die Angabe der Bedingungen, unter denen eine Anweisung wirksam ist. Das Element `Condition` ist optional. Sie können bedingte Ausdrücke erstellen, die Bedingungsoperatoren verwenden, z. B. ist gleich oder kleiner als, damit die Bedingung in der Richtlinie mit Werten in der Anforderung übereinstimmt.

Wenn Sie mehrere `Condition`-Elemente in einer Anweisung oder mehrere Schlüssel in einem einzelnen `Condition`-Element angeben, wertet AWS diese mittels einer logischen `AND`-Operation aus. Wenn Sie mehrere Werte für einen einzelnen Bedingungsschlüssel angeben, AWS wertet die Bedingung mithilfe einer logischen `OR` Operation aus. Alle Bedingungen müssen erfüllt werden, bevor die Berechtigungen der Anweisung gewährt werden.

Sie können auch Platzhaltervariablen verwenden, wenn Sie Bedingungen angeben. Beispielsweise können Sie einem IAM-Benutzer die Berechtigung für den Zugriff auf eine Ressource nur dann gewähren, wenn sie mit dessen IAM-Benutzernamen gekennzeichnet ist. Weitere Informationen finden Sie unter IAM-Richtlinienelemente: Variablen und Tags im IAM-Benutzerhandbuch.

AWS unterstützt globale Bedingungsschlüssel und dienstspezifische Bedingungsschlüssel. Eine Übersicht aller AWS globalen Bedingungsschlüssel finden Sie unter Kontextschlüssel für AWS globale Bedingungen im IAM-Benutzerhandbuch.

Eine Liste der AWS IoT TwinMaker Bedingungsschlüssel finden Sie unter Bedingungsschlüssel für AWS IoT TwinMaker in der Service Authorization Reference. Informationen zu den Aktionen und Ressourcen, mit denen Sie einen Bedingungsschlüssel verwenden können, finden Sie unter Aktionen definiert von AWS IoT TwinMaker.

Beispiele für AWS IoT TwinMaker identitätsbasierte Richtlinien finden Sie unter. Beispiele für identitätsbasierte Richtlinien für AWS IoT TwinMaker

Zugriffskontrolllisten () ACLs in AWS IoT TwinMaker

Unterstützt ACLs: Nein

Zugriffskontrolllisten (ACLs) steuern, welche Principals (Kontomitglieder, Benutzer oder Rollen) über Zugriffsberechtigungen für eine Ressource verfügen. ACLs ähneln ressourcenbasierten Richtlinien, verwenden jedoch nicht das JSON-Richtliniendokumentformat.

Attributbasierte Zugriffskontrolle (ABAC) mit AWS IoT TwinMaker

Unterstützt ABAC (Tags in Richtlinien): Teilweise

Die attributbasierte Zugriffskontrolle (ABAC) ist eine Autorisierungsstrategie, bei der Berechtigungen basierend auf Attributen definiert werden. In werden AWS diese Attribute als Tags bezeichnet. Sie können Tags an IAM-Entitäten (Benutzer oder Rollen) und an viele AWS Ressourcen anhängen. Das Markieren von Entitäten und Ressourcen ist der erste Schritt von ABAC. Anschließend entwerfen Sie ABAC-Richtlinien, um Operationen zuzulassen, wenn das Tag des Prinzipals mit dem Tag der Ressource übereinstimmt, auf die sie zugreifen möchten.

ABAC ist in Umgebungen hilfreich, die schnell wachsen, und unterstützt Sie in Situationen, in denen die Richtlinienverwaltung mühsam wird.

Um den Zugriff auf der Grundlage von Tags zu steuern, geben Sie im Bedingungselement einer Richtlinie Tag-Informationen an, indem Sie die Schlüssel `aws:ResourceTag/key-name`, `aws:RequestTag/key-name`, oder Bedingung `aws:TagKeys` verwenden.

Wenn ein Service alle drei Bedingungsschlüssel für jeden Ressourcentyp unterstützt, lautet der Wert für den Service Ja. Wenn ein Service alle drei Bedingungsschlüssel für nur einige Ressourcentypen unterstützt, lautet der Wert Teilweise.

Weitere Informationen zu ABAC finden Sie unter Definieren von Berechtigungen mit ABAC-Autorisierung im IAM-Benutzerhandbuch. Um ein Tutorial mit Schritten zur Einstellung von ABAC anzuzeigen, siehe Attributbasierte Zugriffskontrolle (ABAC) verwenden im IAM-Benutzerhandbuch.

Temporäre Anmeldeinformationen verwenden mit AWS IoT TwinMaker

Unterstützt temporäre Anmeldeinformationen: Ja

Einige funktionieren AWS-Services nicht, wenn Sie sich mit temporären Anmeldeinformationen anmelden. Weitere Informationen, einschließlich Informationen, die mit temporären Anmeldeinformationen AWS-Services funktionieren AWS-Services , finden Sie im IAM-Benutzerhandbuch unter Diese Option funktioniert mit IAM.

Sie verwenden temporäre Anmeldeinformationen, wenn Sie sich mit einer anderen AWS Management Console Methode als einem Benutzernamen und einem Passwort anmelden. Wenn Sie beispielsweise AWS über den Single Sign-On-Link (SSO) Ihres Unternehmens darauf zugreifen, werden bei diesem Vorgang automatisch temporäre Anmeldeinformationen erstellt. Sie erstellen auch automatisch temporäre Anmeldeinformationen, wenn Sie sich als Benutzer bei der Konsole anmelden und dann die Rollen wechseln. Weitere Informationen zum Wechseln von Rollen finden Sie unter Wechseln von einer Benutzerrolle zu einer IAM-Rolle (Konsole) im IAM-Benutzerhandbuch.

Mithilfe der AWS API AWS CLI oder können Sie temporäre Anmeldeinformationen manuell erstellen. Sie können diese temporären Anmeldeinformationen dann für den Zugriff verwenden AWS. AWS empfiehlt, temporäre Anmeldeinformationen dynamisch zu generieren, anstatt langfristige Zugriffsschlüssel zu verwenden. Weitere Informationen finden Sie unter Temporäre Sicherheitsanmeldeinformationen in IAM.

Serviceübergreifende Prinzipalberechtigungen für AWS IoT TwinMaker

Unterstützt Forward Access Sessions (FAS): Ja

Wenn Sie einen IAM-Benutzer oder eine IAM-Rolle verwenden, um Aktionen auszuführen AWS, gelten Sie als Principal. Bei einigen Services könnte es Aktionen geben, die dann eine andere Aktion in einem anderen Service initiieren. FAS verwendet die Berechtigungen des Prinzipals, der einen aufruft AWS-Service, kombiniert mit der Anforderung, Anfragen an nachgelagerte Dienste AWS-Service zu stellen. FAS-Anfragen werden nur gestellt, wenn ein Dienst eine Anfrage erhält, für deren Abschluss Interaktionen mit anderen AWS-Services oder Ressourcen erforderlich sind. In diesem Fall müssen Sie über Berechtigungen zum Ausführen beider Aktionen verfügen. Einzelheiten zu den Richtlinien für FAS-Anfragen finden Sie unter Zugriffssitzungen weiterleiten.

Servicerollen für AWS IoT TwinMaker

Unterstützt Servicerollen: Ja

Eine Servicerolle ist eine IAM-Rolle, die ein Service annimmt, um Aktionen in Ihrem Namen auszuführen. Ein IAM-Administrator kann eine Servicerolle innerhalb von IAM erstellen, ändern und löschen. Weitere Informationen finden Sie unter Erstellen einer Rolle zum Delegieren von Berechtigungen an einen AWS-Service im IAM-Benutzerhandbuch.

> **Warning**
>
> Durch das Ändern der Berechtigungen für eine Servicerolle kann die AWS IoT TwinMaker Funktionalität beeinträchtigt werden. Bearbeiten Sie Servicerollen nur, AWS IoT TwinMaker wenn Sie dazu eine Anleitung erhalten.

Dienstbezogene Rollen für AWS IoT TwinMaker

Unterstützt serviceverknüpfte Rollen: Ja

Eine dienstbezogene Rolle ist eine Art von Servicerolle, die mit einer verknüpft ist. AWS-Service Der Service kann die Rolle übernehmen, um eine Aktion in Ihrem Namen auszuführen. Dienstbezogene Rollen werden in Ihrem Dienst angezeigt AWS-Konto und gehören dem Dienst. Ein IAM-Administrator kann die Berechtigungen für Service-verknüpfte Rollen anzeigen, aber nicht bearbeiten.

Details zum Erstellen oder Verwalten von serviceverknüpften Rollen finden Sie unter AWS -Services, die mit IAM funktionieren. Suchen Sie in der Tabelle nach einem Service mit einem `Yes` in der Spalte Service-linked role (Serviceverknüpfte Rolle). Wählen Sie den Link Yes (Ja) aus, um die Dokumentation für die serviceverknüpfte Rolle für diesen Service anzuzeigen.

Beispiele für identitätsbasierte Richtlinien für AWS IoT TwinMaker

Benutzer und Rollen haben standardmäßig nicht die Berechtigung, AWS IoT TwinMaker -Ressourcen zu erstellen oder zu ändern. Sie können auch keine Aufgaben mithilfe der AWS Management Console, AWS Command Line Interface (AWS CLI) oder AWS API ausführen. Ein IAM-Administrator muss IAM-Richtlinien erstellen, die Benutzern die Berechtigung erteilen, Aktionen für die Ressourcen auszuführen, die sie benötigen. Der Administrator kann dann die IAM-Richtlinien zu Rollen hinzufügen, und Benutzer können die Rollen annehmen.

Informationen dazu, wie Sie unter Verwendung dieser beispielhaften JSON-Richtliniendokumente eine identitätsbasierte IAM-Richtlinie erstellen, finden Sie unter Erstellen von IAM-Richtlinien (Konsole) im IAM-Benutzerhandbuch.

Einzelheiten zu Aktionen und Ressourcentypen, die von definiert wurden AWS IoT TwinMaker, einschließlich des Formats von ARNs für jeden der Ressourcentypen, finden Sie unter Aktionen, Ressourcen und Bedingungsschlüssel für AWS IoT TwinMaker in der Service Authorization Reference.

Themen

- Bewährte Methoden für Richtlinien
- Verwenden der AWS IoT TwinMaker -Konsole
- Gewähren der Berechtigung zur Anzeige der eigenen Berechtigungen für Benutzer

Bewährte Methoden für Richtlinien

Identitätsbasierte Richtlinien legen fest, ob jemand AWS IoT TwinMaker Ressourcen in Ihrem Konto erstellen, darauf zugreifen oder sie löschen kann. Dies kann zusätzliche Kosten für Ihr verursachen AWS-Konto. Befolgen Sie beim Erstellen oder Bearbeiten identitätsbasierter Richtlinien die folgenden Anleitungen und Empfehlungen:

- Beginnen Sie mit AWS verwalteten Richtlinien und wechseln Sie zu Berechtigungen mit den geringsten Rechten — Verwenden Sie die AWS verwalteten Richtlinien, die Berechtigungen für viele gängige Anwendungsfälle gewähren, um Ihren Benutzern und Workloads zunächst Berechtigungen zu gewähren. Sie sind in Ihrem verfügbar. AWS-Konto Wir empfehlen Ihnen, die Berechtigungen weiter zu reduzieren, indem Sie vom AWS Kunden verwaltete Richtlinien definieren, die speziell auf Ihre Anwendungsfälle zugeschnitten sind. Weitere Informationen finden Sie unter AWS -verwaltete Richtlinien oder AWS -verwaltete Richtlinien für Auftrags-Funktionen im IAM-Benutzerhandbuch.

- Anwendung von Berechtigungen mit den geringsten Rechten – Wenn Sie mit IAM-Richtlinien Berechtigungen festlegen, gewähren Sie nur die Berechtigungen, die für die Durchführung einer Aufgabe erforderlich sind. Sie tun dies, indem Sie die Aktionen definieren, die für bestimmte Ressourcen unter bestimmten Bedingungen durchgeführt werden können, auch bekannt als die geringsten Berechtigungen. Weitere Informationen zur Verwendung von IAM zum Anwenden von Berechtigungen finden Sie unter Richtlinien und Berechtigungen in IAM im IAM-Benutzerhandbuch.
- Verwenden von Bedingungen in IAM-Richtlinien zur weiteren Einschränkung des Zugriffs – Sie können Ihren Richtlinien eine Bedingung hinzufügen, um den Zugriff auf Aktionen und Ressourcen zu beschränken. Sie können beispielsweise eine Richtlinienbedingung schreiben, um festzulegen, dass alle Anforderungen mithilfe von SSL gesendet werden müssen. Sie können auch Bedingungen verwenden, um Zugriff auf Serviceaktionen zu gewähren, wenn diese für einen bestimmten Zweck verwendet werden AWS-Service, z. AWS CloudFormation B. Weitere Informationen finden Sie unter IAM-JSON-Richtlinienelemente: Bedingung im IAM-Benutzerhandbuch.
- Verwenden von IAM Access Analyzer zur Validierung Ihrer IAM-Richtlinien, um sichere und funktionale Berechtigungen zu gewährleisten – IAM Access Analyzer validiert neue und vorhandene Richtlinien, damit die Richtlinien der IAM-Richtliniensprache (JSON) und den bewährten IAM-Methoden entsprechen. IAM Access Analyzer stellt mehr als 100 Richtlinienprüfungen und umsetzbare Empfehlungen zur Verfügung, damit Sie sichere und funktionale Richtlinien erstellen können. Weitere Informationen finden Sie unter Richtlinienvalidierung mit IAM Access Analyzer im IAM-Benutzerhandbuch.
- Multi-Faktor-Authentifizierung (MFA) erforderlich — Wenn Sie ein Szenario haben, das IAM-Benutzer oder einen Root-Benutzer in Ihrem System erfordert AWS-Konto, aktivieren Sie MFA für zusätzliche Sicherheit. Um MFA beim Aufrufen von API-Vorgängen anzufordern, fügen Sie Ihren Richtlinien MFA-Bedingungen hinzu. Weitere Informationen finden Sie unter Sicherer API-Zugriff mit MFA im IAM-Benutzerhandbuch.

Weitere Informationen zu bewährten Methoden in IAM finden Sie unter Bewährte Methoden für die Sicherheit in IAM im IAM-Benutzerhandbuch.

Verwenden der AWS IoT TwinMaker -Konsole

Um auf die AWS IoT TwinMaker Konsole zugreifen zu können, benötigen Sie ein Mindestmaß an Berechtigungen. Diese Berechtigungen müssen es Ihnen ermöglichen, Details zu den AWS IoT TwinMaker Ressourcen in Ihrem aufzulisten und anzuzeigen AWS-Konto. Wenn Sie

eine identitätsbasierte Richtlinie erstellen, die strenger ist als die mindestens erforderlichen Berechtigungen, funktioniert die Konsole nicht wie vorgesehen für Entitäten (Benutzer oder Rollen) mit dieser Richtlinie.

Sie müssen Benutzern, die nur die API AWS CLI oder die AWS API aufrufen, keine Mindestberechtigungen für die Konsole gewähren. Stattdessen sollten Sie nur Zugriff auf die Aktionen zulassen, die der API-Operation entsprechen, die die Benutzer ausführen möchten.

Um sicherzustellen, dass Benutzer und Rollen die AWS IoT TwinMaker Konsole weiterhin verwenden können, fügen Sie den Entitäten auch die AWS IoT TwinMaker `ConsoleAccess` oder die `ReadOnly` AWS verwaltete Richtlinie hinzu. Weitere Informationen finden Sie im Benutzerhandbuch unter Hinzufügen von Berechtigungen für einen AWS IAM Identity Center Benutzer.

Gewähren der Berechtigung zur Anzeige der eigenen Berechtigungen für Benutzer

In diesem Beispiel wird gezeigt, wie Sie eine Richtlinie erstellen, die IAM-Benutzern die Berechtigung zum Anzeigen der eingebundenen Richtlinien und verwalteten Richtlinien gewährt, die ihrer Benutzeridentität angefügt sind. Diese Richtlinie umfasst Berechtigungen zum Ausführen dieser Aktion auf der Konsole oder programmgesteuert mithilfe der API AWS CLI oder AWS .

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
```

```
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

Problembehandlung bei AWS IoT TwinMaker Identität und Zugriff

Verwenden Sie die folgenden Informationen, um häufig auftretende Probleme zu diagnostizieren und zu beheben, die bei der Arbeit mit AWS IoT TwinMaker und IAM auftreten können.

Themen

- Ich bin nicht berechtigt, eine Aktion durchzuführen in AWS IoT TwinMaker
- Ich bin nicht berechtigt, iam auszuführen: PassRole
- Ich möchte Personen außerhalb von mir den Zugriff AWS-Konto auf meine AWS IoT TwinMaker Ressourcen ermöglichen

Ich bin nicht berechtigt, eine Aktion durchzuführen in AWS IoT TwinMaker

Wenn Sie eine Fehlermeldung erhalten, dass Sie nicht zur Durchführung einer Aktion berechtigt sind, müssen Ihre Richtlinien aktualisiert werden, damit Sie die Aktion durchführen können.

Der folgende Beispielfehler tritt auf, wenn der IAM-Benutzer `mateojackson` versucht, über die Konsole Details zu einer fiktiven *`my-example-widget`*-Ressource anzuzeigen, jedoch nicht über `iottwinmaker:`*`GetWidget`*-Berechtigungen verfügt.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 iottwinmaker:GetWidget on resource: my-example-widget
```

In diesem Fall muss die Richtlinie für den Benutzer `mateojackson` aktualisiert werden, damit er mit der `iottwinmaker:`*`GetWidget`*-Aktion auf die *`my-example-widget`*-Ressource zugreifen kann.

Wenn Sie Hilfe benötigen, wenden Sie sich an Ihren AWS Administrator. Ihr Administrator hat Ihnen Ihre Anmeldeinformationen zur Verfügung gestellt.

Ich bin nicht berechtigt, iam auszuführen: PassRole

Wenn Sie die Fehlermeldung erhalten, dass Sie nicht zum Durchführen der `iam:PassRole`-Aktion autorisiert sind, müssen Ihre Richtlinien aktualisiert werden, um eine Rolle an AWS IoT TwinMakerübergeben zu können.

Einige AWS-Services ermöglichen es Ihnen, eine bestehende Rolle an diesen Dienst zu übergeben, anstatt eine neue Servicerolle oder eine dienstverknüpfte Rolle zu erstellen. Hierzu benötigen Sie Berechtigungen für die Übergabe der Rolle an den Dienst.

Der folgende Beispielfehler tritt auf, wenn ein IAM-Benutzer mit dem Namen `marymajor` versucht, die Konsole zu verwenden, um eine Aktion in AWS IoT TwinMaker auszuführen. Die Aktion erfordert jedoch, dass der Service über Berechtigungen verfügt, die durch eine Servicerolle gewährt werden. Mary besitzt keine Berechtigungen für die Übergabe der Rolle an den Dienst.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

In diesem Fall müssen die Richtlinien von Mary aktualisiert werden, um die Aktion `iam:PassRole` ausführen zu können.

Wenn Sie Hilfe benötigen, wenden Sie sich an Ihren AWS Administrator. Ihr Administrator hat Ihnen Ihre Anmeldeinformationen zur Verfügung gestellt.

Ich möchte Personen außerhalb von mir den Zugriff AWS-Konto auf meine AWS IoT TwinMaker Ressourcen ermöglichen

Sie können eine Rolle erstellen, die Benutzer in anderen Konten oder Personen außerhalb Ihrer Organisation für den Zugriff auf Ihre Ressourcen verwenden können. Sie können festlegen, wem die Übernahme der Rolle anvertraut wird. Für Dienste, die ressourcenbasierte Richtlinien oder Zugriffskontrolllisten (ACLs) unterstützen, können Sie diese Richtlinien verwenden, um Personen Zugriff auf Ihre Ressourcen zu gewähren.

Weitere Informationen dazu finden Sie hier:

- Informationen darüber, ob diese Funktionen AWS IoT TwinMaker unterstützt werden, finden Sie unter. Wie AWS IoT TwinMaker funktioniert mit IAM

- Informationen dazu, wie Sie Zugriff auf Ihre Ressourcen gewähren können, AWS-Konten die Ihnen gehören, finden Sie im IAM-Benutzerhandbuch unter Gewähren des Zugriffs auf einen IAM-Benutzer in einem anderen AWS-Konto , den Sie besitzen.
- Informationen dazu, wie Sie Dritten Zugriff auf Ihre Ressourcen gewähren können AWS-Konten, finden Sie AWS-Konten im IAM-Benutzerhandbuch unter Gewähren des Zugriffs für Dritte.
- Informationen dazu, wie Sie über einen Identitätsverbund Zugriff gewähren, finden Sie unter Gewähren von Zugriff für extern authentifizierte Benutzer (Identitätsverbund) im IAM-Benutzerhandbuch.
- Informationen zum Unterschied zwischen der Verwendung von Rollen und ressourcenbasierten Richtlinien für den kontoübergreifenden Zugriff finden Sie unter Kontoübergreifender Ressourcenzugriff in IAM im IAM-Benutzerhandbuch.

Verwenden von serviceverknüpften Rollen für AWS IoT TwinMaker

AWS IoT TwinMaker verwendet AWS Identity and Access Management (IAM) serviceverknüpfte Rollen. Eine serviceverknüpfte Rolle ist ein einzigartiger Typ von IAM-Rolle, mit der direkt verknüpft ist. AWS IoT TwinMaker Mit Diensten verknüpfte Rollen sind vordefiniert AWS IoT TwinMaker und enthalten alle Berechtigungen, die der Dienst benötigt, um andere AWS Dienste in Ihrem Namen aufzurufen.

Eine dienstbezogene Rolle AWS IoT TwinMaker erleichtert die Einrichtung, da Sie die erforderlichen Berechtigungen nicht manuell hinzufügen müssen. AWS IoT TwinMaker definiert die Berechtigungen ihrer dienstbezogenen Rollen und AWS IoT TwinMaker kann, sofern nicht anders definiert, nur ihre Rollen übernehmen. Die definierten Berechtigungen umfassen die Vertrauens- und Berechtigungsrichtlinie. Diese Berechtigungsrichtlinie kann keinen anderen IAM-Entitäten zugewiesen werden.

Sie können eine serviceverknüpfte Rolle erst löschen, nachdem ihre verwandten Ressourcen gelöscht wurden. Dadurch werden Ihre AWS IoT TwinMaker Ressourcen geschützt, da Sie nicht versehentlich die Zugriffsberechtigung für die Ressourcen entziehen können.

Informationen zu anderen Diensten, die dienstverknüpfte Rollen unterstützen, finden Sie unter AWS Dienste, die mit IAM funktionieren. Suchen Sie in der Spalte Dienstverknüpfte Rollen nach den Diensten, für die Ja steht. Wählen Sie über einen Link Ja aus, um die Dokumentation zu einer serviceverknüpften Rolle für diesen Service anzuzeigen.

Berechtigungen für dienstverknüpfte Rollen AWS IoT TwinMaker

AWS IoT TwinMaker verwendet die dienstverknüpfte Rolle namens AWSServiceRoleForIoTTwinMaker — Ermöglicht AWS IoT TwinMaker das Aufrufen anderer AWS Dienste und das Synchronisieren ihrer Ressourcen in Ihrem Namen.

Die mit dem Dienst verknüpfte AWSService RoleForIo TTwin Maker-Rolle vertraut darauf, dass die folgenden Dienste die Rolle übernehmen:

- `iottwinmaker.amazonaws.com`

Die genannte Rollenberechtigungsrichtlinie AWSIo TTwin MakerServiceRolePolicy ermöglicht es AWS IoT TwinMaker , die folgenden Aktionen für die angegebenen Ressourcen durchzuführen:

- Aktion: `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` für `all your iotsitewise asset and asset-model resources`

Sie müssen Berechtigungen konfigurieren, damit eine Benutzer, Gruppen oder Rollen eine serviceverknüpfte Rolle erstellen, bearbeiten oder löschen können. Weitere Informationen finden Sie unter serviceverknüpfte Rollenberechtigung im IAM-Benutzerhandbuch.

Erstellen einer dienstbezogenen Rolle für AWS IoT TwinMaker

Sie müssen eine serviceverknüpfte Rolle nicht manuell erstellen. Wenn Sie Ihre AWS IoT SiteWise Assets und Asset-Modelle (Asset-Synchronisierung) in der AWS Management Console, der oder der AWS API synchronisieren AWS CLI, AWS IoT TwinMaker wird die serviceverknüpfte Rolle für Sie erstellt.

Wenn Sie diese serviceverknüpfte Rolle löschen und sie dann erneut erstellen müssen, können Sie dasselbe Verfahren anwenden, um die Rolle in Ihrem Konto neu anzulegen. Wenn Sie Ihre AWS IoT SiteWise Assets und Asset-Modelle synchronisieren (Asset-Synchronisierung), AWS IoT TwinMaker wird die dienstbezogene Rolle erneut für Sie erstellt.

Sie können die IAM-Konsole auch verwenden, um eine serviceverknüpfte Rolle mit dem Anwendungsfall „IoT TwinMaker — Managed Role“ zu erstellen. Erstellen Sie in der AWS CLI oder der AWS API eine serviceverknüpfte Rolle mit dem `iottwinmaker.amazonaws.com` Dienstnamen. Weitere Informationen finden Sie unter Erstellen einer serviceverknüpften Rolle im IAM-Benutzerhandbuch. Wenn Sie diese serviceverknüpfte Rolle löschen, können Sie mit demselben Verfahren die Rolle erneut erstellen.

Bearbeiten einer dienstbezogenen Rolle für AWS IoT TwinMaker

AWS IoT TwinMaker ermöglicht es Ihnen nicht, die mit dem AWSService RoleForIo TTwin Maker-Dienst verknüpfte Rolle zu bearbeiten. Da möglicherweise verschiedene Entitäten auf die Rolle verweisen, kann der Rollenname nach dem Erstellen einer serviceverknüpften Rolle nicht mehr geändert werden. Sie können jedoch die Beschreibung der Rolle mit IAM bearbeiten. Weitere Informationen finden Sie unter Bearbeiten einer serviceverknüpften Rolle im IAM-Benutzerhandbuch.

Löschen einer dienstbezogenen Rolle für AWS IoT TwinMaker

Wenn Sie ein Feature oder einen Dienst, die bzw. der eine serviceverknüpften Rolle erfordert, nicht mehr benötigen, sollten Sie diese Rolle löschen. Auf diese Weise haben Sie keine ungenutzte juristische Stelle, die nicht aktiv überwacht oder verwaltet wird. Sie müssen jedoch alle ServiceLinked-Workspaces bereinigen, die Ihre serviceverknüpfte Rolle noch verwenden, bevor Sie die Rolle manuell löschen können.

> **Note**
>
> Wenn der AWS IoT TwinMaker Dienst die Rolle verwendet, wenn Sie versuchen, die Ressourcen zu löschen, schlägt das Löschen möglicherweise fehl. Wenn dies passiert, warten Sie einige Minuten und versuchen Sie es erneut.

So löschen Sie die serviceverknüpfte Rolle mit IAM

Verwenden Sie die IAM-Konsole, die oder die AWS API AWS CLI, um die mit dem AWSService RoleForIo TTwin Maker-Dienst verknüpfte Rolle zu löschen. Weitere Informationen finden Sie unter Löschen einer serviceverknüpften Rolle im IAM-Benutzerhandbuch.

Unterstützte Regionen für serviceverknüpfte Rollen AWS IoT TwinMaker

AWS IoT TwinMaker unterstützt die Verwendung von dienstbezogenen Rollen in allen Regionen, in denen der Dienst verfügbar ist. Weitere Informationen finden Sie unter AWS Regionen und Endpunkte.

AWS verwaltete Richtlinien für AWS IoT TwinMaker

Um Benutzern, Gruppen und Rollen Berechtigungen hinzuzufügen, ist es einfacher, AWS verwaltete Richtlinien zu verwenden, als Richtlinien selbst zu schreiben. Es erfordert Zeit und Fachwissen, um von Kunden verwaltete IAM-Richtlinien zu erstellen, die Ihrem Team nur die benötigten Berechtigungen bieten. Um schnell loszulegen, können Sie unsere AWS verwalteten Richtlinien verwenden. Diese Richtlinien decken allgemeine Anwendungsfälle ab und sind in Ihrem AWS-Konto verfügbar. Weitere Informationen zu AWS verwalteten Richtlinien finden Sie im IAM-Benutzerhandbuch unter AWS Verwaltete Richtlinien.

AWS Dienste verwalten und aktualisieren AWS verwaltete Richtlinien. Sie können die Berechtigungen in AWS verwalteten Richtlinien nicht ändern. Services fügen einer von AWS verwalteten Richtlinien gelegentlich zusätzliche Berechtigungen hinzu, um neue Features zu unterstützen. Diese Art von Update betrifft alle Identitäten (Benutzer, Gruppen und Rollen), an welche die Richtlinie angehängt ist. Services aktualisieren eine von AWS verwaltete Richtlinie am ehesten, ein neues Feature gestartet wird oder neue Vorgänge verfügbar werden. Dienste entfernen keine Berechtigungen aus einer AWS verwalteten Richtlinie, sodass durch Richtlinienaktualisierungen Ihre bestehenden Berechtigungen nicht beeinträchtigt werden.

AWS Unterstützt außerdem verwaltete Richtlinien für Jobfunktionen, die sich über mehrere Dienste erstrecken. Die ReadOnlyAccess AWS verwaltete Richtlinie bietet beispielsweise schreibgeschützten Zugriff auf alle AWS Dienste und Ressourcen. Wenn ein Dienst eine neue Funktion startet, werden nur Leseberechtigungen für neue Operationen und Ressourcen AWS hinzugefügt. Eine Liste und Beschreibungen der Richtlinien für Auftragsfunktionen finden Sie in Verwaltete AWS -Richtlinien für Auftragsfunktionen im IAM-Leitfaden.

AWS verwaltete Richtlinie: AWSIo TTwin MakerServiceRolePolicy

Sie können keine Verbindungen AWSIo TTwin MakerServiceRolePolicy zu Ihren IAM-Entitäten herstellen. Diese Richtlinie ist an eine servicegebundene Rolle angehängt, die die Durchführung von Aktionen in Ihrem Namen ermöglicht. Weitere Informationen finden Sie unter Berechtigungen für dienstverknüpfte Rollen AWS IoT TwinMaker.

Die genannte Richtlinie für Rollenberechtigungen AWSIo TTwin MakerServiceRolePolicy AWS IoT TwinMaker ermöglicht es, die folgenden Aktionen für die angegebenen Ressourcen durchzuführen:

- Aktion: `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` für `all your iotsitewise asset and asset-model resources`

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [{
            "Sid": "SiteWiseAssetReadAccess",
            "Effect": "Allow",
            "Action": [
                "iotsitewise:DescribeAsset"
            ],
            "Resource": [
                "arn:aws:iotsitewise:*:*:asset/*"
            ]
        },
        {
            "Sid": "SiteWiseAssetModelReadAccess",
            "Effect": "Allow",
            "Action": [
                "iotsitewise:DescribeAssetModel"
```

```
            ],
            "Resource": [
                "arn:aws:iotsitewise:*:*:asset-model/*"
            ]
        },
        {
            "Sid": "SiteWiseAssetModelAndAssetListAccess",
            "Effect": "Allow",
            "Action": [
                "iotsitewise:ListAssets",
                "iotsitewise:ListAssetModels"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Sid": "TwinMakerAccess",
            "Effect": "Allow",
            "Action": [
                "iottwinmaker:GetEntity",
                "iottwinmaker:CreateEntity",
                "iottwinmaker:UpdateEntity",
                "iottwinmaker:DeleteEntity",
                "iottwinmaker:ListEntities",
                "iottwinmaker:GetComponentType",
                "iottwinmaker:CreateComponentType",
                "iottwinmaker:UpdateComponentType",
                "iottwinmaker:DeleteComponentType",
                "iottwinmaker:ListComponentTypes"
            ],
            "Resource": [
                "arn:aws:iottwinmaker:*:*:workspace/*"
            ],
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "iottwinmaker:linkedServices": [
                        "IOTSITEWISE"
                    ]
                }
            }
        }
    ]
```

```
}
```

AWS IoT TwinMaker Aktualisierungen der AWS verwalteten Richtlinien

Hier finden Sie Informationen zu Aktualisierungen AWS verwalteter Richtlinien, die seit Beginn der Nachverfolgung dieser Änderungen durch diesen Dienst vorgenommen wurden. Um automatische Benachrichtigungen über Änderungen an dieser Seite zu erhalten, abonnieren Sie den RSS-Feed auf der Seite -Dokumentverlauf.

Änderung	Beschreibung	Datum
AWSIoTTwinMakerServiceRolePolicy— Eine Richtlinie wurde hinzugefügt	AWS IoT TwinMaker hat die genannte Richtlinie für Rollenberechtigungen hinzugefügt AWSIoTTwinMakerServiceRolePolicy , mit der die folgenden Aktionen für die angegebenen Ressourcen ausgeführt werden können AWS IoT TwinMaker : • Aktion: `iotsitewise:DescribeAsset, iotsitewise:ListAssets, iotsitewise:DescribeAssetModel, and iotsitewise:ListAssetModels, iottwinmaker:GetEntity, iottwinmaker:CreateEntity, iottwinmaker:UpdateEntity,`	17. November 2023

Änderung	Beschreibung	Datum
	`iottwinmaker:DeleteEntity, iottwinmaker:ListEntities, iottwinmaker:GetComponentType, iottwinmaker:CreateComponentType, iottwinmaker:UpdateComponentType, iottwinmaker:DeleteComponentType, iottwinmaker:ListComponentTypes` für `all your iotsitewise asset and asset-model resources` Weitere Informationen finden Sie unter Berechtigungen für dienstverknüpfte Rollen AWS IoT TwinMaker.	
hat begonnen, Änderungen zu verfolgen	hat begonnen, Änderungen für die AWS verwalteten Richtlinien zu verfolgen.	11. Mai 2022

AWS IoT TwinMaker und Schnittstellen-VPC-Endpunkte ()AWS PrivateLink

Sie können eine private Verbindung zwischen Ihrer Virtual Private Cloud (VPC) herstellen und AWS IoT TwinMaker einen VPC-Schnittstellen-Endpunkt erstellen. Schnittstellenendpunkte werden mit Strom versorgt AWS PrivateLink, sodass Sie AWS IoT TwinMaker APIs ohne Internet-Gateway, NAT-Gerät (Network Address Translation), VPN-Verbindung oder AWS Direct Connect-Verbindung privat

darauf zugreifen können. AWS IoT TwinMaker unterstützt IPv4 sowohl als auch IPv6 (Dual-Stack) über seine Schnittstellenendpunkte. Instances in Ihrer VPC benötigen keine öffentlichen IP-Adressen für die Kommunikation. AWS IoT TwinMaker APIs Datenverkehr zwischen Ihrer VPC und dem, der das Amazon-Netzwerk AWS IoT TwinMaker nicht verlässt.

Jeder Schnittstellenendpunkt wird durch eine oder mehrere Elastic-Network-Schnittstellen in Ihren Subnetzen dargestellt.

Weitere Informationen finden Sie unter Schnittstellen-VPC-Endpunkte (AWS PrivateLink) im Amazon-VPC-Benutzerhandbuch.

Überlegungen zu AWS IoT TwinMaker VPC-Endpunkten

Bevor Sie einen Schnittstellen-VPC-Endpunkt für einrichten AWS IoT TwinMaker, lesen Sie die Eigenschaften und Einschränkungen des Schnittstellen-Endpunkts im Amazon VPC-Benutzerhandbuch.

AWS IoT TwinMaker unterstützt Aufrufe aller API-Aktionen von Ihrer VPC aus.

- Verwenden Sie für API-Operationen auf Datenebene den folgenden Endpunkt:

  ```
  data.iottwinmaker.region.amazonaws.com
  ```

  Die API-Operationen auf der Datenebene umfassen Folgendes:

  - GetPropertyValue
  - GetPropertyValueHistory
  - BatchPutPropertyValues
- Verwenden Sie für die API-Operationen auf der Steuerungsebene den folgenden Endpunkt:

  ```
  api.iottwinmaker.region.amazonaws.com
  ```

  Zu den unterstützten API-Vorgängen auf der Kontrollebene gehören:

  - CreateComponentType
  - CreateEntity
  - CreateScene
  - CreateWorkspace
  - DeleteComponentType

- DeleteEntity
- DeleteScene
- DeleteWorkspace
- GetComponentType
- GetEntity
- GetScene
- GetWorkspace
- ListComponentTypes
- ListComponentTypes
- ListEntities
- ListScenes
- ListTagsForResource
- ListWorkspaces
- TagResource
- UntagResource
- UpdateComponentType
- UpdateEntity
- UpdateScene
- UpdateWorkspace

Erstellen eines Schnittstellen-VPC-Endpunkts für AWS IoT TwinMaker

Sie können einen VPC-Endpunkt für den AWS IoT TwinMaker Service erstellen, indem Sie entweder die Amazon VPC-Konsole oder die AWS Command Line Interface ()AWS CLI verwenden. Weitere Informationen finden Sie unter Erstellung eines Schnittstellenendpunkts im Benutzerhandbuch für Amazon VPC.

Erstellen Sie einen VPC-Endpunkt AWS IoT TwinMaker , der den folgenden Dienstnamen verwendet.

- Verwenden Sie für API-Operationen auf Datenebene den folgenden Dienstnamen:

```
com.amazonaws.region.iottwinmaker.data
```

- Verwenden Sie für API-Operationen auf Steuerungsebene den folgenden Dienstnamen:

```
com.amazonaws.region.iottwinmaker.api
```

Wenn Sie privates DNS für den Endpunkt aktivieren, können Sie API-Anfragen an richten, AWS IoT TwinMaker indem Sie den Standard-DNS-Namen für die Region verwenden, `iottwinmaker.us-east-1.amazonaws.com` z. B.

Weitere Informationen finden Sie unter Zugriff auf einen Service über einen Schnittstellenendpunkt im Benutzerhandbuch für Amazon VPC.

AWS IoT TwinMaker PrivateLink wird in den folgenden Regionen unterstützt:

- us-east-1

  Der ControlPlane Dienst wird in den folgenden Verfügbarkeitszonen unterstützt: `use1-az1use1-az2`, und`use1-az6`.

  Der DataPlane Dienst wird in den folgenden Verfügbarkeitszonen unterstützt: `use1-az1use1-az2`, und`use1-az4`.
- us-west-2

  Die DataPlane Dienste ControlPlane und werden in den folgenden Verfügbarkeitszonen unterstützt: `usw2-az1usw2-az2`, und`usw2-az3`.
- eu-west-1
- eu-central-1
- ap-southeast-1
- ap-southeast-2

Weitere Informationen zu Availability Zones finden Sie unter Availability Zone IDs for your AWS resources — AWS Resource Access Manager.

Zugriff AWS IoT TwinMaker über eine Schnittstelle (VPC-Endpunkt)

Wenn Sie einen Schnittstellenendpunkt erstellen, AWS IoT TwinMaker generiert dieser endpunktspezifische DNS-Hostnamen, mit denen Sie kommunizieren können. AWS IoT TwinMaker Die private DNS-Option ist standardmäßig aktiviert. Weitere Informationen finden Sie unter Verwenden von privat gehosteten Zonen im Amazon VPC-Benutzerhandbuch.

Wenn Sie privates DNS für den Endpunkt aktivieren, können Sie API-Anfragen an einen AWS IoT TwinMaker der folgenden VPC-Endpunkte stellen.

- Verwenden Sie für die API-Operationen auf der Datenebene den folgenden Endpunkt. *region*Ersetzen Sie es durch Ihre AWS Region.

```
data.iottwinmaker.region.amazonaws.com
```

- Verwenden Sie für die API-Operationen auf der Kontrollebene den folgenden Endpunkt. *region*Ersetzen Sie es durch Ihre AWS Region.

```
api.iottwinmaker.region.amazonaws.com
```

Wenn Sie privates DNS für den Endpunkt deaktivieren, müssen Sie für den Zugriff AWS IoT TwinMaker über den Endpunkt wie folgt vorgehen:

- Geben Sie die VPC-Endpunkt-URL in API-Anfragen an.
  - Verwenden Sie für die API-Operationen auf der Datenebene die folgende Endpunkt-URL. Ersetzen Sie *vpc-endpoint-id* und *region* durch Ihre VPC-Endpunkt-ID und Region.

```
vpc-endpoint-id.data.iottwinmaker.region.vpce.amazonaws.com
```

  - Verwenden Sie für die API-Operationen auf der Kontrollebene die folgende Endpunkt-URL. Ersetzen Sie *vpc-endpoint-id* und *region* durch Ihre VPC-Endpunkt-ID und Region.

```
vpc-endpoint-id.api.iottwinmaker.region.vpce.amazonaws.com
```

- Deaktivieren Sie die Host-Präfix-Injektion. Der AWS SDKs Dienstendpunkt AWS CLI und stellt dem Dienstendpunkt verschiedene Hostpräfixe voran, wenn Sie die einzelnen API-Operationen aufrufen. Dies führt dazu AWS SDKs , dass das AWS CLI und ungültig URLs für erzeugt AWS IoT TwinMaker , wenn Sie einen VPC-Endpunkt angeben.

> **Important**
>
> Sie können die Hostpräfixinjektion in AWS CLI oder AWS -Tools für PowerShell nicht deaktivieren. Das heißt, wenn Sie privates DNS deaktiviert haben, können Sie den VPC-Endpunkt nicht verwenden AWS CLI oder AWS IoT TwinMaker über ihn AWS -Tools für

PowerShell darauf zugreifen. Wenn Sie diese Tools für den Zugriff AWS IoT TwinMaker über den Endpunkt verwenden möchten, aktivieren Sie privates DNS.

Weitere Informationen zur Deaktivierung der Hostpräfixinjektion im finden Sie in den folgenden Dokumentationsabschnitten für jedes SDK: AWS SDKs

- AWS SDK für C++
- AWS SDK für Go
- AWS SDK für Go v2
- AWS SDK für Java
- AWS SDK for Java 2.x
- AWS SDK für JavaScript
- AWS SDK für .NET
- AWS SDK für PHP
- AWS SDK für Python (Boto3)
- AWS SDK für Ruby

Weitere Informationen finden Sie unter Zugriff auf einen Service über einen Schnittstellenendpunkt im Benutzerhandbuch für Amazon VPC.

Erstellen einer VPC-Endpunktrichtlinie für AWS IoT TwinMaker

Sie können eine Endpunktrichtlinie an Ihren VPC-Endpunkt anhängen, der den Zugriff auf AWS IoT TwinMaker steuert. Die Richtlinie gibt die folgenden Informationen an:

- Prinzipal, der die Aktionen ausführen kann.
- Aktionen, die ausgeführt werden können
- Die Ressourcen, für die Aktionen ausgeführt werden können.

Weitere Informationen finden Sie unter Steuerung des Zugriffs auf Services mit VPC-Endpunkten im Amazon-VPC-Benutzerhandbuch.

Beispiel: VPC-Endpunktrichtlinie für Aktionen AWS IoT TwinMaker

Im Folgenden finden Sie ein Beispiel für eine Endpunktrichtlinie für AWS IoT TwinMaker. Wenn diese Richtlinie an einen Endpunkt angehängt ist, gewährt sie dem IAM-Benutzer `iottwinmakeradmin` im AWS Konto Zugriff auf die aufgelisteten AWS IoT TwinMaker Aktionen `123456789012` auf allen Ressourcen.

```
{
   "Statement":[
      {
        "Principal": {
            "AWS": "arn:aws:iam::123456789012:user/role"
                },
         "Resource": "*",
         "Effect":"Allow",
         "Action":[
            "iottwinmaker:CreateEntity",
            "iottwinmaker:GetScene",
            "iottwinmaker:ListEntities"
         ]
        }
    ]
}
```

Konformitätsprüfung für AWS IoT TwinMaker

Informationen darüber, ob AWS-Service ein AWS-Services in den Geltungsbereich bestimmter Compliance-Programme fällt, finden Sie unter Umfang nach Compliance-Programm AWS-Services unter . Wählen Sie dort das Compliance-Programm aus, an dem Sie interessiert sind. Allgemeine Informationen finden Sie unter AWS Compliance-Programme AWS .

Sie können Prüfberichte von Drittanbietern unter herunterladen AWS Artifact. Weitere Informationen finden Sie unter Berichte herunterladen unter .

Ihre Verantwortung für die Einhaltung der Vorschriften bei der Nutzung AWS-Services hängt von der Vertraulichkeit Ihrer Daten, den Compliance-Zielen Ihres Unternehmens und den geltenden Gesetzen und Vorschriften ab. AWS stellt die folgenden Ressourcen zur Verfügung, die Sie bei der Einhaltung der Vorschriften unterstützen:

- Compliance und Governance im Bereich Sicherheit – In diesen Anleitungen für die Lösungsimplementierung werden Überlegungen zur Architektur behandelt. Außerdem werden Schritte für die Bereitstellung von Sicherheits- und Compliance-Features beschrieben.

- Referenz für berechtigte HIPAA-Services – Listet berechtigte HIPAA-Services auf. Nicht alle AWS-Services sind HIPAA-fähig.
- AWS Compliance-Ressourcen — Diese Sammlung von Arbeitsmappen und Leitfäden gilt möglicherweise für Ihre Branche und Ihren Standort.
- AWS Leitfäden zur Einhaltung von Vorschriften für Kunden — Verstehen Sie das Modell der gemeinsamen Verantwortung aus dem Blickwinkel der Einhaltung von Vorschriften. In den Leitfäden werden die bewährten Verfahren zur Sicherung zusammengefasst AWS-Services und die Leitlinien den Sicherheitskontrollen in verschiedenen Frameworks (einschließlich des National Institute of Standards and Technology (NIST), des Payment Card Industry Security Standards Council (PCI) und der International Organization for Standardization (ISO)) zugeordnet.
- Evaluierung von Ressourcen anhand von Regeln im AWS Config Entwicklerhandbuch — Der AWS Config Service bewertet, wie gut Ihre Ressourcenkonfigurationen den internen Praktiken, Branchenrichtlinien und Vorschriften entsprechen.
- AWS Security Hub— Auf diese AWS-Service Weise erhalten Sie einen umfassenden Überblick über Ihren internen Sicherheitsstatus. AWS Security Hub verwendet Sicherheitskontrollen, um Ihre AWS -Ressourcen zu bewerten und Ihre Einhaltung von Sicherheitsstandards und bewährten Methoden zu überprüfen. Die Liste der unterstützten Services und Kontrollen finden Sie in der Security-Hub-Steuerelementreferenz.
- Amazon GuardDuty — Dies AWS-Service erkennt potenzielle Bedrohungen für Ihre Workloads AWS-Konten, Container und Daten, indem es Ihre Umgebung auf verdächtige und böswillige Aktivitäten überwacht. GuardDuty kann Ihnen helfen, verschiedene Compliance-Anforderungen wie PCI DSS zu erfüllen, indem es die in bestimmten Compliance-Frameworks vorgeschriebenen Anforderungen zur Erkennung von Eindringlingen erfüllt.
- AWS Audit Manager— Auf diese AWS-Service Weise können Sie Ihre AWS Nutzung kontinuierlich überprüfen, um das Risikomanagement und die Einhaltung von Vorschriften und Industriestandards zu vereinfachen.

Resilienz in AWS IoT TwinMaker

Die AWS globale Infrastruktur basiert auf Availability AWS-Regionen Zones. AWS-Regionen bieten mehrere physisch getrennte und isolierte Availability Zones, die über Netzwerke mit niedriger Latenz, hohem Durchsatz und hoher Redundanz miteinander verbunden sind. Mithilfe von Availability Zones können Sie Anwendungen und Datenbanken erstellen und ausführen, die automatisch Failover zwischen Zonen ausführen, ohne dass es zu Unterbrechungen kommt. Availability Zones sind

besser verfügbar, fehlertoleranter und skalierbarer als herkömmliche Infrastrukturen mit einem oder mehreren Rechenzentren.

Weitere Informationen zu Availability Zones AWS-Regionen und Availability Zones finden Sie unter AWS Globale Infrastruktur.

Zusätzlich zur AWS globalen Infrastruktur AWS IoT TwinMaker bietet es mehrere Funktionen zur Unterstützung Ihrer Datenausfallsicherheit und Backup-Anforderungen.

Sicherheit der Infrastruktur in AWS IoT TwinMaker

Als verwalteter Service AWS IoT TwinMaker ist er durch die AWS globalen Netzwerksicherheitsverfahren geschützt, die im Whitepaper Amazon Web Services: Sicherheitsprozesse im Überblick beschrieben sind.

Sie verwenden AWS veröffentlichte API-Aufrufe, um AWS IoT TwinMaker über das Netzwerk darauf zuzugreifen. Clients müssen Transport Layer Security (TLS) 1.2 oder höher unterstützen. Wir empfehlen TLS 1.3 oder höher. Clients müssen außerdem Verschlüsselungssammlungen mit PFS (Perfect Forward Secrecy) wie DHE (Ephemeral Diffie-Hellman) oder ECDHE (Elliptic Curve Ephemeral Diffie-Hellman) unterstützen. Die meisten modernen Systemen wie Java 7 und höher unterstützen diese Modi.

Außerdem müssen Anforderungen mit einer Zugriffsschlüssel-ID und einem geheimen Zugriffsschlüssel signiert sein, der einem IAM-Prinzipal zugeordnet ist. Alternativ können Sie mit AWS Security Token Service (AWS STS) temporäre Sicherheitsanmeldeinformationen erstellen, um die Anforderungen zu signieren.

Endpunkte und Kontingente

AWS IoT TwinMaker Endpunkte und Kontingente

Informationen zu AWS IoT TwinMaker Endpunkten und Kontingenten finden Sie in der AWS Allgemeinen Referenz.

- Informationen zu Dienstendpunkten finden Sie unter AWS IoT TwinMaker Dienstendpunkte.
- Informationen zu Kontingenten finden Sie unter AWS IoT TwinMaker Dienstkontingente.
- Informationen zu API-Drosselungsgrenzwerten finden Sie unter AWS IoT TwinMaker API-Drosselungsgrenzen.

Zusätzliche Informationen zu Endpunkten AWS IoT TwinMaker

Verwenden Sie einen Endpunkt AWS IoT TwinMaker, um programmgesteuert eine Verbindung herzustellen. Wenn Sie einen HTTP-Client verwenden, müssen Sie der Steuerungsebene und der Datenebene APIs wie folgt ein Präfix hinzufügen. Es ist jedoch nicht erforderlich, dem AWS SDK und den AWS Command Line Interface Befehlen ein Präfix hinzuzufügen, da sie automatisch das erforderliche Präfix hinzufügen.

- Verwenden Sie das `api` Präfix für die Steuerungsebene APIs. Beispiel, `api.iottwinmaker.us-west-1.amazonaws.com`.
- Verwenden Sie das `data` Präfix für die Datenebene APIs. Beispiel, `data.iottwinmaker.us-west-1.amazonaws.com`.

Dokumentenverlauf für das AWS IoT TwinMaker Benutzerhandbuch

In der folgenden Tabelle werden die Dokumentationsversionen für beschrieben AWS IoT TwinMaker.

Änderung	Beschreibung	Datum
Neue serviceverknüpfte Rolle und neue IAM-Richtlinie	AWS IoT TwinMaker hat eine neue dienstbezogene Rolle namens AWSServiceRoleForIoTTwinMaker hinzugefügt. AWS IoT TwinMaker hat diese neue dienstbezogene Rolle hinzugefügt, um es AWS IoT TwinMaker zu ermöglichen, andere AWS Dienste anzurufen und deren Ressourcen in Ihrem Namen zu synchronisieren. Die neue AWSIo TTwin MakerServ iceRolePolicy IAM-Richtlinie ist mit dieser Rolle verknüpft , und die Richtlinie gewährt die Erlaubnis, andere AWS Dienste anzurufen und deren Ressourcen in Ihrem Namen zu synchronisieren. AWS IoT TwinMaker	17. November 2023
Erstversion	Erste Version des AWS IoT TwinMaker Benutzerh andbuchs	30. November 2021

Die vorliegende Übersetzung wurde maschinell erstellt. Im Falle eines Konflikts oder eines Widerspruchs zwischen dieser übersetzten Fassung und der englischen Fassung (einschließlich infolge von Verzögerungen bei der Übersetzung) ist die englische Fassung maßgeblich.