



User Guide

# AWS IoT TwinMaker



# AWS IoT TwinMaker: User Guide

Copyright © 2024 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 Konnektor zu einer externen Datenquelle .....	15
Ändern Sie Ihre Workspace-IAM-Rolle, um den Athena-Datenkonnektor zu verwenden .....	17
Einen Workspace erstellen .....	18
Erstellen Sie Ihre erste Entität .....	21
Ein AWS Konto einrichten .....	24
Melden Sie sich an für ein AWS-Konto .....	24
Erstellen Sie einen Benutzer mit Administratorzugriff .....	25
Komponententypen verwenden und erstellen .....	27
Integrierte Komponententypen .....	27
Kernmerkmale der AWS IoT TwinMaker Komponententypen .....	28
Eigenschaftsdefinitionen erstellen .....	30
Erstellung von Funktionen .....	31
Beispiele für Komponententypen .....	31
Alarm (Zusammenfassung) .....	32
Timestream-Telemetrie .....	33
Alarm (erbt vom abstrakten Alarm) .....	34
Beispiele für Geräte .....	35
Massenoperationen .....	38
Schlüsselkonzepte und Terminologie .....	38
AWS IoT TwinMaker metadataTransferJob Funktionalität .....	39
Durchführen von Massenimport- und Exportvorgängen .....	40
metadataTransferJob Voraussetzungen .....	41

IAM-Berechtigungen .....	41
Ausführen einer Massenoperation .....	45
Fehlerbehandlung .....	49
Importieren von Metadatenvorlagen .....	49
AWS IoT TwinMaker metadataTransferJob Beispiele .....	53
AWS IoT TwinMaker Schema für Metadatenübertragungsaufträge .....	54
Datenanschlüsse .....	72
Datenanschlüsse .....	72
Konnektor für den Schema-Initialisierer .....	73
DataReaderByEntity .....	74
DataReaderByComponentType .....	75
DataReader .....	77
AttributePropertyValueReaderByEntity .....	78
DataWriter .....	79
Beispiele .....	80
Athena tabellarischer Datenkonnektor .....	89
AWS IoT TwinMaker Voraussetzungen für den Athena-Datenkonnektor .....	89
Verwenden des Athena-Datenkonnektors .....	90
Verwenden der JSON-Referenz für den Tabellendatenkonnektor von Athena .....	94
Verwenden des Athena-Datenkonnektors .....	95
Visualisieren Sie Athena-Tabellendaten in Grafana .....	96
AWS IoT TwinMaker Konnektor für Zeitreihen-Daten .....	98
AWS IoT TwinMaker Voraussetzungen für Zeitreihen-Datenkonnektoren .....	99
Hintergrund des Datenkonnektors für Zeitreihen .....	99
Entwicklung eines Datenkonnektors für Zeitreihen .....	101
Verbesserung Ihres Datenkonnektors .....	110
Testen Sie Ihren Connector .....	111
Sicherheit .....	111
AWS IoT TwinMaker Ressourcen erstellen .....	111
Was kommt als Nächstes .....	113
AWS IoT TwinMaker Cookie Factory-Datenanschluss .....	113
AWS IoT TwinMaker Szenen erstellen .....	119
Vor dem Erstellen von Szenen .....	119
Optimieren Sie Ihre Ressourcen, bevor Sie sie importieren AWS IoT TwinMaker .....	119
Bewährte Methoden für die Leistung in AWS IoT TwinMaker .....	120
Weitere Informationen .....	120

Ressourcen hochladen in AWS IoT TwinMaker .....	121
Laden Sie Dateien mithilfe der Konsole in die Ressourcenbibliothek hoch .....	121
Erstelle deine Szenen .....	122
Verwenden Sie die 3D-Navigation AWS IoT TwinMaker in Ihren In-Szenen .....	123
Fügen Sie feste Kameras hinzu .....	125
Verbesserte Bearbeitung .....	126
Gezielte Platzierung von Szenenobjekten .....	126
Auswahl eines Untermodells .....	126
Bearbeiten Sie Entitäten in der Szenenhierarchie .....	127
Fügen Sie Anmerkungen zu Entitäten hinzu .....	128
Fügen Sie Überlagerungen zu Tags hinzu .....	133
Bearbeiten Sie Ihre Szenen .....	141
Fügen Sie Modelle hinzu .....	141
Fügen Sie Widgets hinzu .....	142
Hinzufügen von Tags .....	146
Optimieren Sie Ihr 3D-Modell .....	146
Verwenden von 3D-Kacheln in Ihrer Szene .....	146
Dynamische Szenen .....	149
Statische und dynamische Szenen .....	149
Typen und Entitäten von Szenenkomponenten .....	150
Dynamische Szenenkonzepte .....	151
AWS IoT TwinMaker App-Kit-Integration .....	152
AWS IoT TwinMaker Preismodus wechseln .....	153
Wissensgraph .....	155
AWS IoT TwinMaker Kernkonzepte des Wissensgraphen .....	155
Verwenden von Knowledge Graph .....	156
Generieren Sie ein Szenengraph .....	159
AWS IoT TwinMaker Voraussetzungen für Szenendiagramme .....	160
Binden Sie 3D-Knoten in Ihrer Szene .....	160
Erstellen Sie eine Webanwendung .....	163
Wissensgraph Grafana-Panel .....	165
AWS IoT TwinMaker Voraussetzungen für den Abfrage-Editor .....	165
Wissensgraph Grafana-Berechtigungen .....	165
Knowledge Graph zusätzliche Ressourcen .....	170
Asset-Synchronisierung mit AWS IoT SiteWise .....	183
Verwenden Sie die Asset-Synchronisierung mit AWS IoT SiteWise .....	183

Verwenden eines benutzerdefinierten Arbeitsbereichs .....	184
Das Internet der Dinge nutzen SiteWiseDefaultWorkspace .....	187
Unterschiede zwischen benutzerdefinierten und Standardarbeitsbereichen .....	188
Ressourcen, die synchronisiert wurden von AWS IoT SiteWise .....	189
Benutzerdefinierte und Standardarbeitsbereiche .....	189
Nur Standard-Workspace .....	190
Ressourcen wurden nicht synchronisiert .....	191
Verwenden Sie synchronisierte Entitäten und Komponententypen in AWS IoT TwinMaker ..	191
Analysieren Sie den Synchronisierungsstatus und die Fehler .....	192
Synchronisieren Sie den Jobstatus .....	193
Löschen Sie einen Synchronisierungsauftrag .....	195
Beschränkungen der Asset-Synchronisierung .....	196
Einrichten von Grafana-Dashboards .....	198
CORS-Konfiguration .....	199
Einrichten Ihrer Grafana-Umgebung .....	200
Amazon Managed Grafana .....	200
Selbstverwaltetes Grafana .....	201
Erstellen einer Dashboard-Rolle .....	202
Eine IAM-Richtlinie erstellen .....	202
Video vom Edge hochladen .....	206
Weitere Berechtigungen hinzufügen .....	206
Erstellen der Grafana-Dashboard-IAM-Rolle .....	208
Erstellen einer -AWS IoT TwinMakerVideo-Player-Richtlinie .....	209
Einschränken des Zugriffs auf Ihre -Ressourcen .....	210
Einschränken von GET-Berechtigungen .....	210
AWS IoT SiteWise BatchPutAssetPropertyValue Eingrenzungsberechtigung .....	212
Alarmer mit Grafana-Dashboards verbinden .....	215
AWS IoT SiteWise Voraussetzungen für die Alarmkonfiguration .....	215
Definieren Sie die AWS IoT SiteWise IAM-Rolle der Alarmkomponente .....	216
Abfragen und Aktualisierungen über die AWS IoT TwinMaker API .....	217
Konfigurieren Sie Ihr Grafana-Dashboard für Alarmer .....	219
Verwenden Sie das Grafana-Dashboard zur Alarmvisualisierung .....	221
Matterport-Integration .....	223
Überblick über die Integration .....	224
Voraussetzungen für die Matterport-Integration .....	225
Matterport SDK-Anmeldeinformationen .....	227

Speichern Sie die Matterport-Anmeldeinformationen in AWS Secrets Manager .....	228
Matterport-Scans in Szenen AWS IoT TwinMaker .....	231
Matterport in deinem AWS IoT TwinMaker Grafana-Dashboard .....	237
Matterport-Integration mit dem App-Kit AWS IoT .....	237
Streamen von Videos an AWS IoT TwinMaker .....	239
Verwenden des Edge-Konnektors für den Kinesis-Videostrom zum Streamen von Videos in AWS IoT TwinMaker .....	239
Voraussetzungen .....	239
Erstellen von Videokomponenten für AWS IoT TwinMaker Szenen .....	240
Video und Metadaten aus dem Kinesis-Videostrom zu einem Grafana-Dashboard hinzufügen .	241
Verwenden der AWS IoT TwinMaker Flink-Bibliothek .....	242
Protokollierung und Überwachung .....	243
Überwachung mit CloudWatch Amazon-Metriken .....	244
Metriken .....	244
Protokollierung von API-Aufrufen mit AWS CloudTrail .....	247
AWS IoT TwinMakerInformationen in CloudTrail .....	248
Sicherheit .....	250
Datenschutz .....	251
Verschlüsselung im Ruhezustand .....	252
Verschlüsselung während der Übertragung .....	252
Identitäts- und Zugriffsverwaltung .....	252
Zielgruppe .....	253
Authentifizierung mit Identitäten .....	254
Verwalten des Zugriffs mit Richtlinien .....	258
Wie AWS IoT TwinMaker funktioniert mit IAM .....	260
Beispiele für identitätsbasierte Richtlinien .....	268
Fehlerbehebung .....	271
Verwenden von serviceverknüpften Rollen .....	274
AWS verwaltete Richtlinien .....	276
VPC-Endpunkte (AWS PrivateLink) .....	281
Überlegungen zu AWS IoT TwinMaker VPC-Endpunkten .....	281
Erstellen eines Schnittstellen-VPC-Endpunkts für AWS IoT TwinMaker .....	283
Zugriff AWS IoT TwinMaker über eine Schnittstelle (VPC-Endpunkt) .....	284
Erstellen einer VPC-Endpunktrichtlinie für AWS IoT TwinMaker .....	286
Compliance-Validierung .....	287
Ausfallsicherheit .....	288

---

Sicherheit der Infrastruktur .....	288
Endpunkte und Kontingente .....	290
AWS IoT TwinMaker Endpunkte und Kontingente .....	290
Zusätzliche Endpunktinformationen .....	290
Dokumentverlauf .....	291
.....	ccxcii



# 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, 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 geht auf diese Herausforderungen ein, indem es die folgenden Funktionen bereitstellt.

- Wissensgraph für Entitätskomponentensysteme: AWS IoT TwinMaker bietet Tools zur Modellierung von 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

Daten, die in AWS IoT SiteWise und Kinesis Video Streams gespeichert sind. 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.

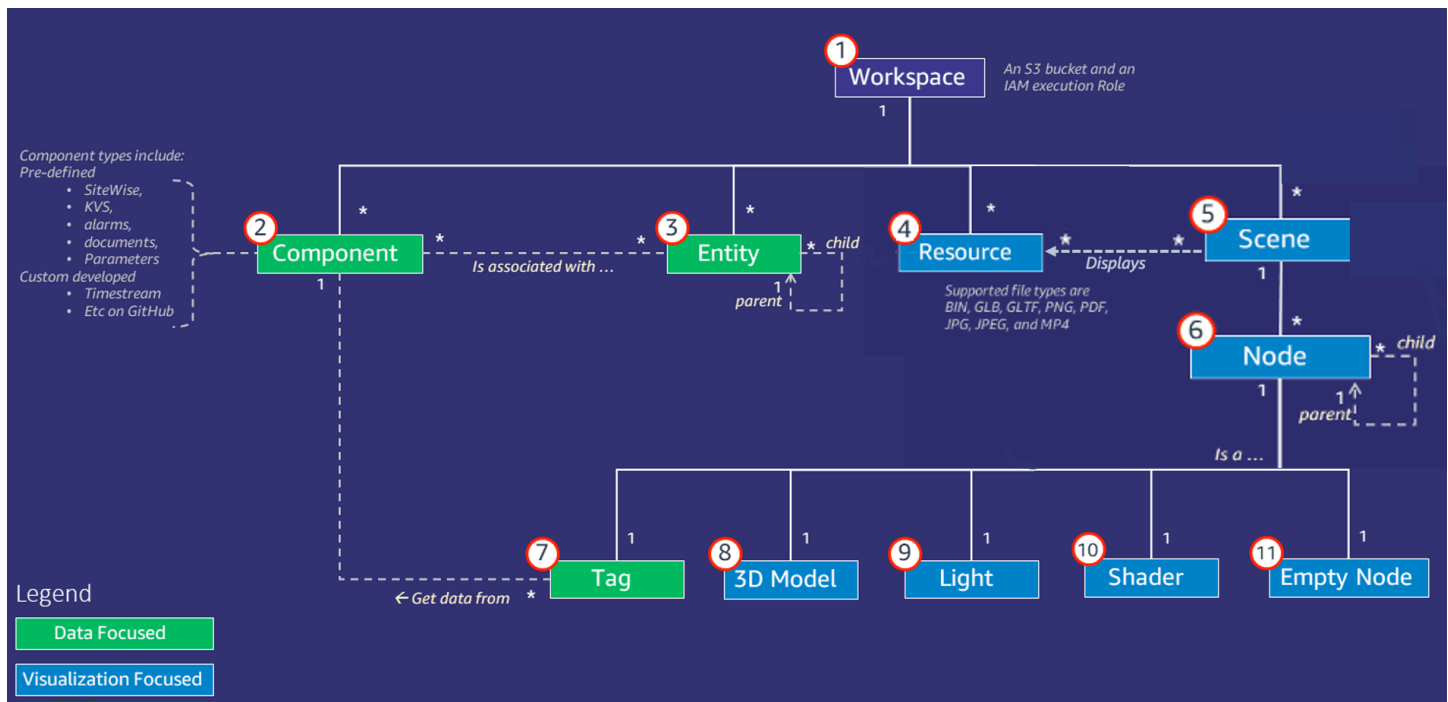
- **Scene Composer:** Die AWS IoT TwinMaker Konsole bietet ein Tool zur Szenenkomposition zum Erstellen von Szenen in 3D. Sie laden Ihre zuvor erstellten 3D-/CAD-Modelle hoch, die für die Webanzeige optimiert und in das .gltf- oder .glb-Format konvertiert wurden. Anschließend verwenden Sie den Scene 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 von der Konsole aus 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, z. B. Benutzerhandbücher oder Bilder, die sich unter bestimmten URLs befinden.
- Zeitreihen, z. B. Sensordaten von AWS IoT SiteWise.
- Alarmer, z. B. Zeitreihenalarmer 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 Ort, an dem sich die Quelldaten Ihres digitalen Zwillings befinden. 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 .gltb-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 Folgendes tun 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.

```
{
  "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.

```
{
  "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.

```
{
  "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.

```

{
  "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 auslesen müssen, 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 Konnektor 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.

```
{
  "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 resources arn"
  ]
}, // Athena permission
{
  "Effect": "Allow",
  "Action": [
    "glue:GetTable",
    "glue:GetTables",
    "glue:GetDatabase",
    "glue:GetDatabases"
  ]
}
```

```

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

```

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 erscheint oben auf der Workspaces-Seite.



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.

```

{
  "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": "*"
    }
  ]
}

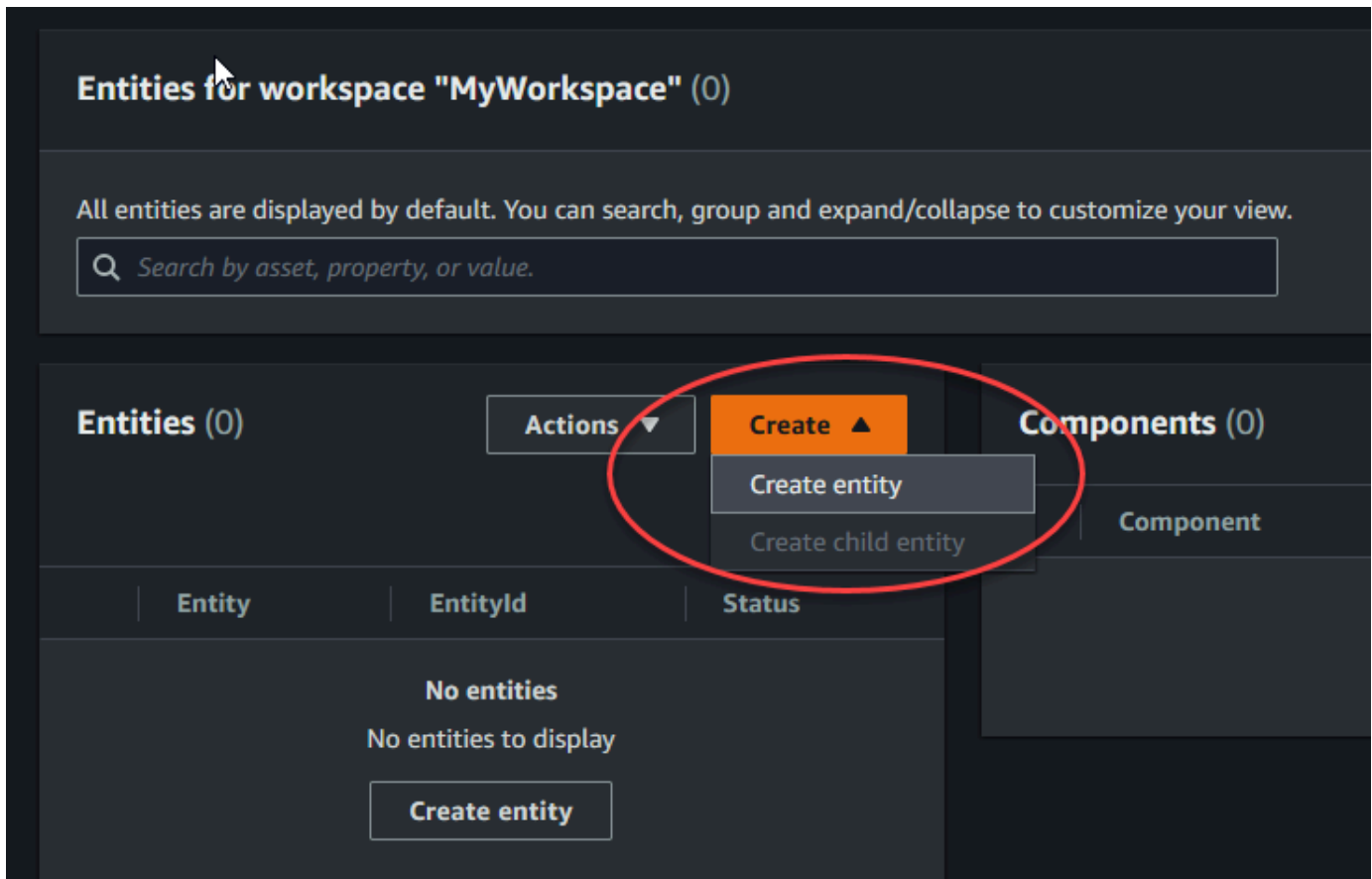
```

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 bietet die folgenden integrierten Komponententypen.

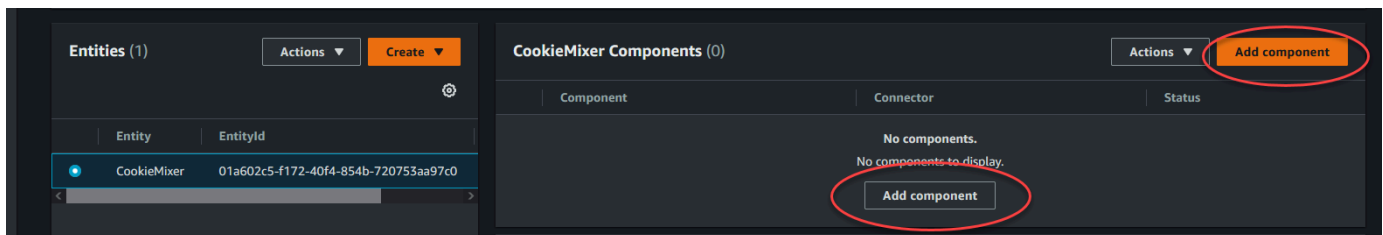
- Parameter: Fügt eine Reihe von Schlüssel-Wert-Eigenschaften hinzu.

- **Dokument:** Fügt einen Namen und eine URL für ein Dokument hinzu, das Informationen über die Entität enthält.
- **Alarmer:** 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 unter [Komponententypen 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.

**Add component** ✕

Name

MixerDescription

Type

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

com.amazon.iottwinmaker.documents ▼

Edit form  Edit JSON

Document editor

No docs associated to the entity

**Add a doc**

▼ **Properties**

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

Value

\_\_\_\_\_

**Add another property**

Cancel **Add component**

3. 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 URLs speichern, 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/signup>.
2. Folgen Sie den Online-Anweisungen.

Bei der Anmeldung müssen Sie auch einen Telefonanruf entgegennehmen und einen Verifizierungscode über die Telefontasten eingeben.

Wenn Sie sich für eine anmelden AWS-Konto, Root-Benutzer des AWS-Kontos wird eine erstellt. Der Root-Benutzer hat Zugriff auf alle AWS-Services und Ressourcen des Kontos. Aus Sicherheitsgründen sollten Sie einem Benutzer Administratorzugriff zuweisen und nur den Root-Benutzer verwenden, um [Aufgaben auszuführen, für die Root-Benutzerzugriff erforderlich](#) ist.

## Melden Sie sich 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/signup>.
2. Folgen Sie den Online-Anweisungen.

Bei der Anmeldung müssen Sie auch einen Telefonanruf entgegennehmen und einen Verifizierungscode über die Telefontasten eingeben.

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



Sicherheitsgründen sollten Sie einem Benutzer Administratorzugriff zuweisen und nur den Root-Benutzer verwenden, um [Aufgaben auszuführen, für die Root-Benutzerzugriff erforderlich](#) ist.

AWS sendet Ihnen nach Abschluss des Anmeldevorgangs eine Bestätigungs-E-Mail. Sie können jederzeit Ihre aktuelle Kontoaktivität anzeigen und Ihr Konto verwalten. Rufen Sie dazu <https://aws.amazon.com/> auf und klicken Sie auf Mein Konto.

## Erstellen Sie einen Benutzer 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 Console](#) als 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 Sie einen Benutzer 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 Benutzer in IAM Identity Center Administratorzugriff.

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

## Melden Sie sich als Benutzer mit Administratorzugriff an

- 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 Zugriffsportal](#).

## Weisen Sie weiteren Benutzern Zugriff zu

1. Erstellen Sie in IAM Identity Center einen Berechtigungssatz, der der bewährten Methode zur Anwendung von Berechtigungen mit den geringsten Rechten folgt.

Anweisungen finden Sie im Benutzerhandbuch unter [Einen Berechtigungssatz erstellen](#).AWS IAM Identity Center

2. Weisen Sie Benutzer einer Gruppe zu und weisen Sie der Gruppe dann Single Sign-On-Zugriff zu.

Anweisungen finden [Sie im AWS IAM Identity Center Benutzerhandbuch unter Gruppen hinzufügen](#).

# 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 [CreateComponentType](#)API oder mithilfe des Komponententyp-Editors in der AWS IoT TwinMaker Konsole übergeben können.

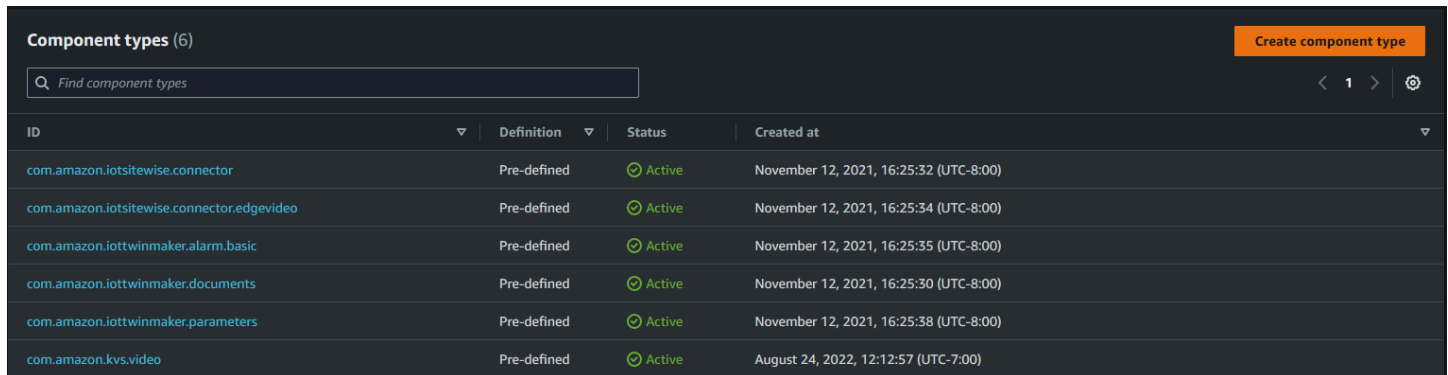
Komponenten bieten Kontext für Eigenschaften und Daten für die zugehörigen Entitäten.

## 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 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 für Dokumente, 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



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

## Kernmerkmale der AWS IoT TwinMaker Komponententypen

In der folgenden Liste werden die Kernfunktionen der Komponententypen beschrieben.

- **Eigenschaftsdefinitionen:** Das [PropertyDefinitionRequest](#) Objekt 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 [FunctionRequest](#) Objekt 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 [GetPropertyValueHistory](#)API-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 [GetPropertyValueHistory](#)API-Vorgang nur entitätsübergreifende Abfragen für Eigenschaften in diesem Komponententyp. (Sie können nur die Historie der Eigenschaftswerte für `anforderncomponentTypeId` anfordern.)

- `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-Telemetrikomponenten, 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 `PropertyDefinitionRequest` beschrieben.

Parameter	Beschreibung
<code>isExternalId</code>	<p>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.</p> <p>Standardmäßig wird der Wert dieser Eigenschaft auf <code>false</code> gesetzt.</p>
<code>isStoredExternally</code>	<p>Ein boolescher Wert, der angibt, ob der Eigenschaftswert extern gespeichert wird.</p> <p>Standardmäßig wird der Wert dieser Eigenschaft auf <code>false</code> gesetzt.</p>
<code>isTimeSeries</code>	<p>Ein boolescher Wert, der angibt, ob die Eigenschaft Zeitreihendaten speichert.</p> <p>Der Standardwert dieser Eigenschaft ist <code>false</code>.</p>
<code>isRequiredInEntity</code>	<p>Ein boolescher Wert, der angibt, ob die Eigenschaft einen Wert in einer Entität haben muss, die den Komponententyp verwendet.</p>
<code>dataType</code>	<p>Ein <a href="#">DataType</a>-Objekt, das den Datentyp (z. B. Zeichenfolge, Map, Liste und Maßeinheit) der Eigenschaft angibt.</p>
<code>defaultValue</code>	<p>Ein <a href="#">DataValue</a>-Objekt, das den Standardwert der Eigenschaft angibt.</p>

Parameter	Beschreibung
<code>configuration</code>	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 beschreibenFunctionRequest`.

Parameter	Beschreibung
<code>implementedBy</code>	Ein <a href="#">DataConnector</a> Objekt, das die Lambda-Funktion spezifiziert, die eine Verbindung zur externen Datenquelle herstellt.
<code>requiredProperties</code>	Eine Liste von Eigenschaften, die die Funktion benötigt, um aus einer externen Datenquelle zu lesen und in diese zu schreiben.
<code>scope</code>	Der Geltungsbereich der Funktion. Wird <code>Workspace</code> für Funktionen verwendet, deren Geltungsbereich sich über einen gesamten Workspace erstreckt. Wird <code>Entity</code> 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 `bestehdataReader`, 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 abrufen. 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 Typ der Timestream-Telemetrie. Es spezifiziert eine eigene Lambda-Funktion, die Alarmdaten abrufen.

```

{
  "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 eine `metadataTransferJob`, um Ihre AWS IoT TwinMaker Ressourcen in großem Umfang zu übertragen und zu verwalten. Mit einem `metadataTransferJob` können Sie Massenoperationen durchführen und Ressourcen zwischen AWS IoT TwinMaker und AWS IoT SiteWise Amazon S3 übertragen.

Sie können Massenoperationen in den folgenden Szenarien verwenden:

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

Themen

- [Schlüsselkonzepte und Terminologie](#)
- [Durchführen von Massenimport- und Exportvorgängen](#)
- [AWS IoT TwinMaker Schema für Metadatenübertragungsaufträge](#)

## Schlüsselkonzepte und Terminologie

AWS IoT TwinMaker -Massenoperationen verwenden die folgenden Konzepte und Terminologie:

- Importieren von : Die Aktion zum Verschieben von Ressourcen in einen AWS IoT TwinMaker -Workspace. Zum Beispiel aus einer lokalen Datei, einer Datei in einem Amazon S3-Bucket oder aus AWS IoT SiteWise einem - AWS IoT TwinMaker Workspace.
- Exportieren von : Die Aktion zum Verschieben von Ressourcen aus einem AWS IoT TwinMaker -Workspace zu einem lokalen Computer oder einem Amazon S3-Bucket.
- Quelle : Der Ausgangspunkt, von dem aus Sie Ressourcen verschieben möchten.

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

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

Ein Amazon S3-Bucket ist beispielsweise ein Exportziel und ein - AWS IoT TwinMaker Workspace ist ein Importziel.

- AWS IoT SiteWise Schema : Ein Schema, das zum Importieren und Exportieren von Ressourcen in und aus verwendet wird AWS IoT SiteWise.
- AWS IoT TwinMaker Schema : Ein Schema, das zum Importieren und Exportieren von Ressourcen in und aus verwendet wird AWS IoT TwinMaker.
- AWS IoT TwinMaker Ressourcen der obersten Ebene: Ressourcen, die in vorhandenen APIs verwendet werden. Insbesondere eine -Entität oder eine ComponentType.
- AWS IoT TwinMaker Ressourcen auf Unterebene: Verschachtelte Ressourcentypen, die in Metadatendefinitionen verwendet werden. Insbesondere eine Komponente .
- Metadaten: Schlüsselinformationen, die für den erfolgreichen Import oder Export von - AWS IoT SiteWise und - AWS IoT TwinMaker Ressourcen erforderlich sind.
- metadataTransferJob: Das Objekt, das erstellt wurde, wenn Sie ausführenCreateMetadataTransferJob.

## AWS IoT TwinMaker metadataTransferJob Funktionalität

In diesem Thema wird das AWS IoT TwinMaker folgende Verhalten erläutert, wenn Sie einen Massenvorgang ausführen – wie ein verarbeitet metadataTransferJob wird. Außerdem wird erläutert, wie Sie ein Schema mit den Metadaten definieren, die für die Übertragung Ihrer resources. AWS IoT TwinMaker bulk-Operationen erforderlich sind, und wie Sie die folgenden Funktionen unterstützen:

- Erstellen oder Ersetzen von Ressourcen der obersten Ebene: 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 ersetzt, die in der Vorlage unter dem Entity Schlüssel definiert ist.

- Erstellen oder Ersetzen einer Subressource:

Auf der EntityComponent Ebene können Sie nur eine Komponente erstellen oder ersetzen. Die Entität muss bereits vorhanden sein, andernfalls erzeugt die Aktion eine ValidationException.

Auf Eigenschafts- oder Beziehungsebene können Sie nur eine Eigenschaft oder Beziehung erstellen oder ersetzen, und das enthaltende EntityComponent muss bereits vorhanden sein.

- Löschen einer Subressource:

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 übergeordnete Ressource und lassen die Definition der Unterressource weg.

- Keine Löschung von Ressourcen der obersten Ebene: AWS IoT TwinMaker löscht niemals Ressourcen der obersten Ebene. Eine Ressource der obersten Ebene bezieht sich auf eine Entität oder ComponentType.
- Keine Subressourcendefinitionen für dieselbe Ressource der obersten Ebene in einer Vorlage:

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

Wenn eine entityId in Entity verwendet wird, können Sie nicht dieselbe ID in Entity, EntityComponentEigenschaft oder Beziehung verwenden.

Wenn eine Kombination aus entityId oder componentName in verwendet wird EntityComponent, können Sie nicht dieselbe Kombination in EntityComponent, Eigenschaft oder Beziehung verwenden.

Wenn eine Kombination aus entityId ,componentName und propertyName in Eigenschaft oder Beziehung verwendet wird, können Sie nicht dieselbe Kombination in der Eigenschaft oder Beziehung verwenden.

- ExternalId ist optional für AWS IoT TwinMaker: Die ExternalId kann verwendet werden, um Ihnen bei der Identifizierung Ihrer -Ressourcen zu helfen.

## Durchführen von Massenimport- und Exportvorgängen

In diesem Thema wird beschrieben, wie Sie Massenimport- und Exportvorgänge durchführen und Fehler in Ihren Übertragungsaufträgen behandeln. Es enthält Beispiele für Übertragungsaufträge mit CLI-Befehlen.



Die AWS IoT TwinMaker -API-Referenz enthält Informationen über die [CreateMetadataTransferJob](#) und andere API-Aktionen.

## Themen

- [metadataTransferJob Voraussetzungen](#)
- [IAM-Berechtigungen](#)
- [Ausführen einer Massenoperation](#)
- [Fehlerbehandlung](#)
- [Importieren von Metadatenvorlagen](#)
- [AWS IoT TwinMaker metadataTransferJob Beispiele](#)

## metadataTransferJob Voraussetzungen

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

- Erstellen Sie einen - AWS IoT TwinMaker Workspace. Der Workspace kann das Importziel oder die Exportquelle für ein sein metadataTransferJob. Informationen zum Erstellen eines Workspace finden Sie unter [Einen Workspace erstellen](#).
- Erstellen Sie einen Amazon 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 ausführen, müssen Sie eine IAM-Richtlinie mit Berechtigungen erstellen AWS IoT TwinMaker, um den Austausch von AWS Ressourcen zwischen Amazon S3 AWS IoT SiteWise und Ihrem lokalen Computer zu ermöglichen. Weitere Informationen zum Erstellen von IAM-Richtlinien finden Sie unter [Erstellen von IAM-Richtlinien](#).

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

- AWS IoT TwinMaker Richtlinie :

```
{
  "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 Richtlinie :

```

{
  "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-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-Richtlinie so einschränken, dass sie nur auf einen einzelnen Amazon S3-Bucket zugreift, siehe die folgende Richtlinie.

### Richtlinie für einen einzelnen Amazon S3-Bucket

```

{
  "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/*"
  ]
}

```

## Festlegen der Zugriffskontrolle für ein metadataTransferJob

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

### Note

Diese Richtlinie gewährt nur 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"
      ]
    }
  }
}
```

## Ausführen einer Massenoperation

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

### Importieren von Daten aus Amazon S3 in AWS IoT TwinMaker

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

Beispielschemata finden Sie unter [Importieren von Metadatenvorlagen](#).

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

Im Folgenden finden Sie ein Beispiel für einen Anforderungstext:

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

```

    "type": "s3",
    "s3Configuration": {
      "location": "arn:aws:s3:::your-S3-bucket-name/your_import_data.json"
    }
  ]],
  "destination": {
    "type": "iottwinmaker",
    "iotTwinMakerConfiguration": {
      "workspace": "arn:aws:iottwinmaker:us-
east-1:111122223333:workspace/your-worksapce-name"
    }
  }
}

```

Notieren Sie sich den Dateinamen, den Sie Ihrem Anforderungstext gegeben haben. Sie benötigen ihn im nächsten Schritt. In diesem Beispiel heißt der Anforderungstext `createMetadataTransferJobImport.json`.

3. Führen Sie den folgenden CLI-Befehl aus, um aufzurufen `CreateMetadataTransferJob` (ersetzen Sie den Eingabe-JSON-Dateinamen durch den Namen, den Sie Ihrem Anforderungstext gegeben haben):

```

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

```

Dadurch wird ein erstellt `metadataTransferJob` und der Prozess der Übertragung Ihrer ausgewählten Ressourcen beginnt.

## Exportieren von Daten aus AWS IoT TwinMaker nach Amazon S3

1. Erstellen Sie einen JSON-Anforderungstext mit den entsprechenden Filtern, um die Ressourcen auszuwählen, die Sie exportieren möchten. In diesem 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:::your-S3-bucket-location"
    }
}
}
}

```

Mit dem `filtersArray` können Sie angeben, welche Ressourcen exportiert werden sollen. 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-Bucket als Ziel des Metadatenübertragungsauftrags angeben.

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

2. Führen Sie den folgenden CLI-Befehl aus, um aufzurufen `CreateMetadataTransferJob` (ersetzen Sie den `input-json`-Dateinamen durch den Namen, den Sie Ihrem Anforderungstext gegeben haben):

```

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

```

Dadurch wird ein `createMetadataTransferJob` erstellt und der Prozess der Übertragung Ihrer ausgewählten Ressourcen beginnt.

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

- Um einen Auftrag abzubrechen, verwenden Sie die [CancelMetadataTransferJob](#) -API-Aktion. Wenn Sie aufrufen `CancelMetadataTransferJob`, bricht die API nur eine laufende ab, und alle Ressourcen `metadataTransferJob`, die bereits exportiert oder importiert wurden, sind von diesem API-Aufruf nicht betroffen.
- Um Informationen zu einem bestimmten Auftrag abzurufen, verwenden Sie die [GetMetadataTransferJob](#) -API-Aktion.

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

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

Wenn Sie `GetMetadataTransferJob` für einen nicht vorhandenen AWS IoT TwinMaker Import- oder Exportauftrag aufrufen, erhalten Sie als Antwort eine `ResourceNotFoundException` Fehlermeldung.

- Um aktuelle Aufträge aufzulisten, verwenden Sie die [ListMetadataTransferJobs](#) -API-Aktion.

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 mit der Quelle und dem Ziel Ihres Import- oder Exportauftrags übereinstimmen.

Weitere Beispiele für CLI-Befehle, die diese API-Aktionen aufrufen, finden Sie unter [AWS IoT TwinMaker metadataTransferJob Beispiele](#).

Wenn während des Übertragungsauftrags Fehler auftreten, finden Sie weitere Informationen unter [Fehlerbehandlung](#).



## Fehlerbehandlung

Nachdem Sie einen Übertragungsauftrag erstellt und ausgeführt haben, können Sie aufrufen `GetMetadataTransferJob`, um aufgetretene Fehler zu diagnostizieren:

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

Sobald Sie sehen, dass der Status des Auftrags zu `COMPLETED` wechselt, können Sie die Ergebnisse des Auftrags überprüfen. `GetMetadataTransferJob` gibt ein Objekt mit dem Namen `MetadataTransferJobProgress` zurück, das die folgenden Felder enthält:

- `failedCount` : Gibt die Anzahl der Ressourcen an, die während des Übertragungsprozesses fehlgeschlagen sind.
- `skippedCount` : Gibt die Anzahl der Ressourcen an, die während des Übertragungsprozesses übersprungen wurden.
- `succeededCount` : Gibt die Anzahl der Ressourcen an, die während des Übertragungsprozesses erfolgreich waren.
- `totalCount` Gibt die Gesamtzahl der Ressourcen an, die am Übertragungsprozess beteiligt sind.

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

## Importieren von Metadatenvorlagen

Sie können viele Komponenten, `componentTypes` oder Entitäten mit einem einzigen Massenimportvorgang importieren. Die Beispiele in diesem Abschnitt zeigen, wie Sie dies tun.

template: Importing entities

Verwenden Sie das folgende Vorlagenformat für einen Auftrag, 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 Auftrag, 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 Auftrag, 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 Metadatenübertragungen zu verwalten:

- [CreateMetadataTransferJob](#) API-Aktion.

Beispiel für einen -CLI-Befehl:

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

- Um einen Auftrag abubrechen, verwenden Sie die [CancelMetadataTransferJob](#) -API-Aktion.

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 Metadatenübertragungsauftrag abgebrochen, und alle bereits exportierten oder importierten Ressourcen sind davon nicht betroffen.

- Um Informationen zu einem bestimmten Auftrag abzurufen, verwenden Sie die [GetMetadataTransferJob](#) -API-Aktion.

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 [ListMetadataTransferJobs](#)-API-Aktion, um aktuelle Aufträge 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 ihn und notieren Sie sich den Dateinamen. Sie benötigen ihn bei der Eingabe des CLI-Befehls .

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 Schema für Metadatenübertragungsaufträge

`metadataTransferJob` Importschema: Verwenden Sie dieses AWS IoT TwinMaker Metadatenschema, um Ihre Daten zu validieren, wenn Sie sie in einen Amazon 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": "$ROOT"
  },
  "DisplayName": {
    "description": "A friendly name for the property.",
    "type": "string",
    "pattern": ".*[^\u0000-\u001F\u007F]*.*",
    "minLength": 0,
    "maxLength": 256
  },
  "Tags": {
    "description": "Metadata that you can use to manage the entity / componentType",
    "patternProperties": {
      "^[a-zA-Z0-9_./-]*$": {
        "type": "string",
        "minLength": 1,
        "maxLength": 256
      }
    }
  },
  "existingJavaType": "java.util.Map<String,String>",
  "minProperties": 0,
  "maxProperties": 50
},
"Relationship": {
  "description": "The type of the relationship.",
  "type": "object",
  "properties": {
    "relationshipType": {
      "description": "The type of the relationship.",
      "type": "string",
      "pattern": ".*",
      "minLength": 1,
      "maxLength": 256
    }
  }
},

```



```

    "targetComponentTypeId": {
      "description": "The ID of the target component type associated with this
relationship.",
      "$ref": "#/definitions/ComponentTypeId"
    }
  },
  "additionalProperties": false
},
"DataValue": {
  "description": "An object that specifies a value for a property.",
  "type": "object",
  "properties": {
    "booleanValue": {
      "description": "A Boolean value.",
      "type": "boolean"
    },
    "doubleValue": {
      "description": "A double value.",
      "type": "number"
    },
    "expression": {
      "description": "An expression that produces the value.",
      "type": "string",
      "pattern": "(^\\$\\$\\{Parameters\\. [a-zA-z]+([a-zA-z_0-9]*)\\}$)",
      "minLength": 1,
      "maxLength": 316
    },
    "integerValue": {
      "description": "An integer value.",
      "type": "integer"
    },
    "listValue": {
      "description": "A list of multiple values.",
      "type": "array",
      "minItems": 0,
      "maxItems": 50,
      "uniqueItems": false,
      "insertionOrder": false,
      "items": {
        "$ref": "#/definitions/DataValue"
      },
      "default": null
    },
    "longValue": {

```

```

    "description": "A long value.",
    "type": "integer",
    "existingJavaType": "java.lang.Long"
  },
  "stringValue": {
    "description": "A string value.",
    "type": "string",
    "pattern": ".*",
    "minLength": 1,
    "maxLength": 256
  },
  "mapValue": {
    "description": "An object that maps strings to multiple DataValue objects.",
    "type": "object",
    "patternProperties": {
      "[a-zA-Z_\\-0-9]+": {
        "$ref": "#/definitions/DataValue"
      }
    },
    "additionalProperties": {
      "$ref": "#/definitions/DataValue"
    }
  },
  "relationshipValue": {
    "description": "A value that relates a component to another component.",
    "type": "object",
    "properties": {
      "TargetComponentName": {
        "type": "string",
        "pattern": "[a-zA-Z_\\-0-9]+",
        "minLength": 1,
        "maxLength": 256
      },
      "TargetEntityId": {
        "type": "string",
        "pattern": "[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}|
^ [a-zA-Z0-9][a-zA-Z_\\-0-9.:]*[a-zA-Z0-9]+",
        "minLength": 1,
        "maxLength": 128
      }
    },
    "additionalProperties": false
  }
},

```

```
    "additionalProperties": false
  },
  "DataType": {
    "description": "An object that specifies the data type of a property.",
    "type": "object",
    "properties": {
      "allowedValues": {
        "description": "The allowed values for this data type.",
        "type": "array",
        "minItems": 0,
        "maxItems": 50,
        "uniqueItems": false,
        "insertionOrder": false,
        "items": {
          "$ref": "#/definitions/DataValue"
        },
        "default": null
      },
      "nestedType": {
        "description": "The nested type in the data type.",
        "$ref": "#/definitions/DataType"
      },
      "relationship": {
        "description": "A relationship that associates a component with another component.",
        "$ref": "#/definitions/Relationship"
      },
      "type": {
        "description": "The underlying type of the data type.",
        "type": "string",
        "enum": [
          "RELATIONSHIP",
          "STRING",
          "LONG",
          "BOOLEAN",
          "INTEGER",
          "DOUBLE",
          "LIST",
          "MAP"
        ]
      },
      "unitOfMeasure": {
        "description": "The unit of measure used in this data type.",
        "type": "string",
```

```

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

```

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

```

    "additionalProperties": false
  },
  "Properties": {
    "type": "object",
    "patternProperties": {
      "[a-zA-Z_\\-0-9]+": {
        "$ref": "#/definitions/Property"
      }
    },
    "additionalProperties": {
      "$ref": "#/definitions/Property"
    }
  },
  "PropertyName": {
    "type": "string",
    "pattern": "[a-zA-Z_\\-0-9]+"
  },
  "PropertyGroup": {
    "description": "An object that specifies information about a property group.",
    "type": "object",
    "properties": {
      "groupType": {
        "description": "The type of property group.",
        "type": "string",
        "enum": [
          "TABULAR"
        ]
      },
      "propertyName": {
        "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 und eine Entität namens `component.type.initial` erstellt:

```

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

```

Hier ist ein Beispiel, das vorhandene 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 [GetPropertyValueHistory](#)API-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 immer noch 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 [GetPropertyValueHistory](#) API-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 aus 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 [GetPropertyValueHistory](#) API-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 [GetPropertyValueHistory](#) API-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 [BatchPutPropertyValues](#) API-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](#).
- 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](#).

## Verwenden des Athena-Datenkonnektors

Um den Athena-Datenkonnektor zu verwenden, 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 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.

### Component type information

ID

Description

Must be less than 2048 characters

Base Type

Choose a pre-defined Component Type or create your own

5. Wählen Sie den Basistyp. Wählen Sie aus der Drop-down-Liste den Athena Tabular Data Connector aus, der als `com.amazon.athena.connector` bezeichnet ist.
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

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

### External ID Column

assetid

### Athena workgroup


Select an Athena workgroup

Testworkgroup

9. (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 Komponentennamen 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.

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

- 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 Eigenschaftensreihenfolgen. 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.

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

Query 1 Transform 0

Query type: Get Property value

Entity: TabularEntity1

Component Name: TabularComponent

Property Group: tabularPropertyGroup (TABULAR)

Selected Properties: won (INTEGER) × status (STRING) × total (INTEGER) × crit (INTEGER) × description (STRING) × equipment\_type (STRING) ×

Filter: crit (INTEGER) = 5

OrderBy: total (INTEGER) DESC

## 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

TelemetryAssetId	TelemetryAssetType	measure_name	time	measure_value:varchar	measure_value:double
Mixer_22_680b5b8e-1afe-4a77-87ab-834f8e5ba01e	Mixer	Temperature	2022-04-19 00:28:00.241000000	-	99.1292877197266
Mixer_20_0568f25f-116c-429c-a974-5ceec065a6ac	Mixer	RPM	2022-04-19 00:28:00.241000000	-	59.4233207702637
Mixer_22_680b5b8e-1afe-4a77-87ab-834f8e5ba01e	Mixer	RPM	2022-04-19 00:28:00.241000000	-	59.9421195983887
Mixer_24_7ff0b75b-f0fa-43f0-bc89-b96337586d00	Mixer	Temperature	2022-04-19 00:28:00.241000000	-	99.1292877197266
Mixer_25_cf42effc-ba19-48ba-bbc3-d21d2508ce31	Mixer	RPM	2022-04-19 00:28:00.241000000	-	59.8453979492188
Mixer_20_0568f25f-116c-429c-a974-5ceec065a6ac	Mixer	Temperature	2022-04-19 00:28:00.241000000	-	99.1292877197266
Mixer_24_7ff0b75b-f0fa-43f0-bc89-b96337586d00	Mixer	RPM	2022-04-19 00:28:00.241000000	-	60.4532585144043
Mixer_15_0bb566cd-d6f3-4804-9fe1-7d2abcd82d0	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.

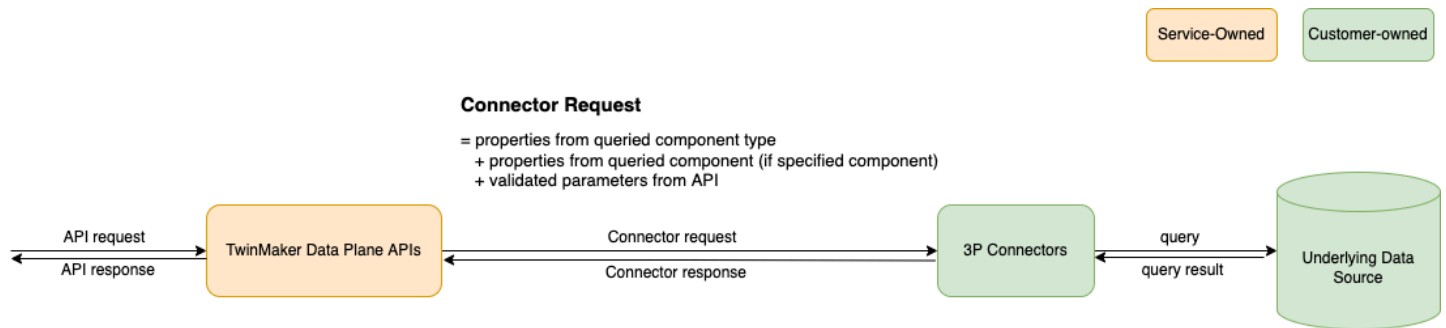
## Datenfluss von Zeitreihen-Datenkonnektoren

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 Komponententypeneigenschaften 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 Sie denselben Workspace aus, in dem Sie die Komponententypen erstellt haben, und öffnen Sie ihn.
  - 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 `startDateTime`, `startTime`, `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.

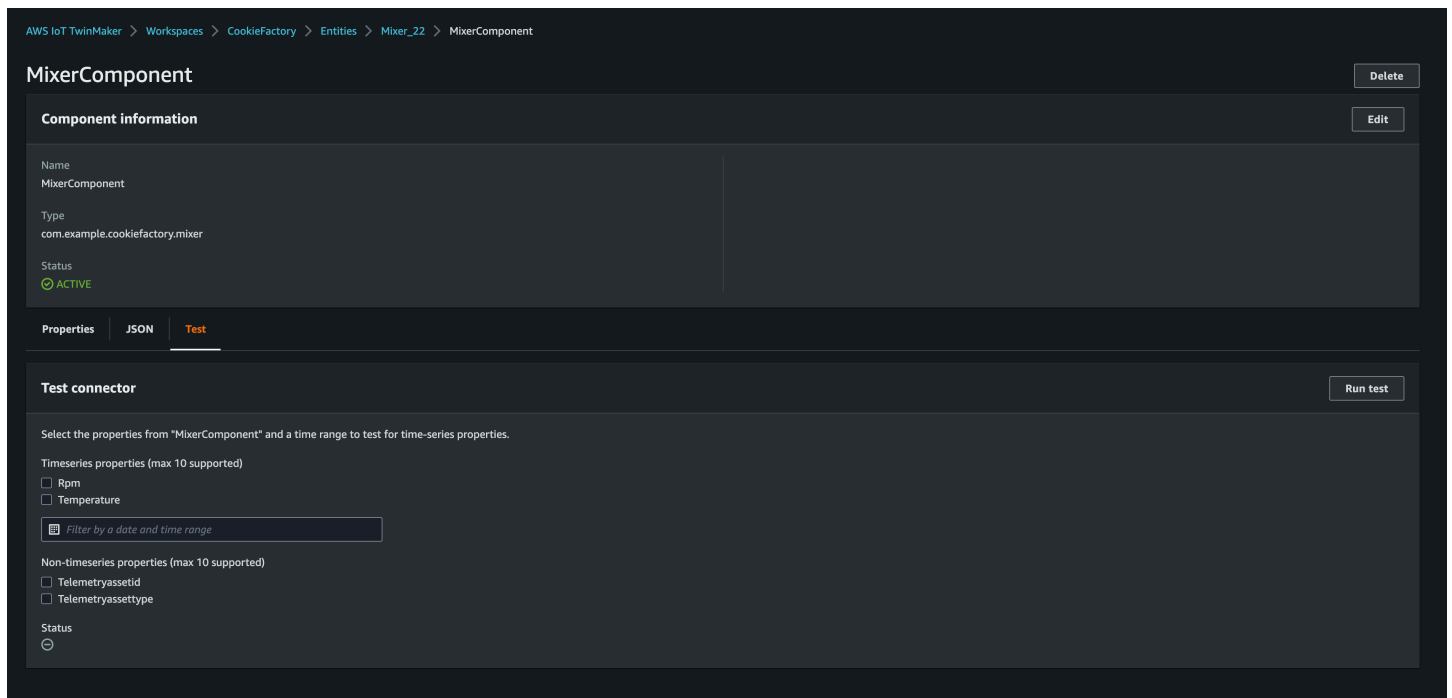
### Note

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

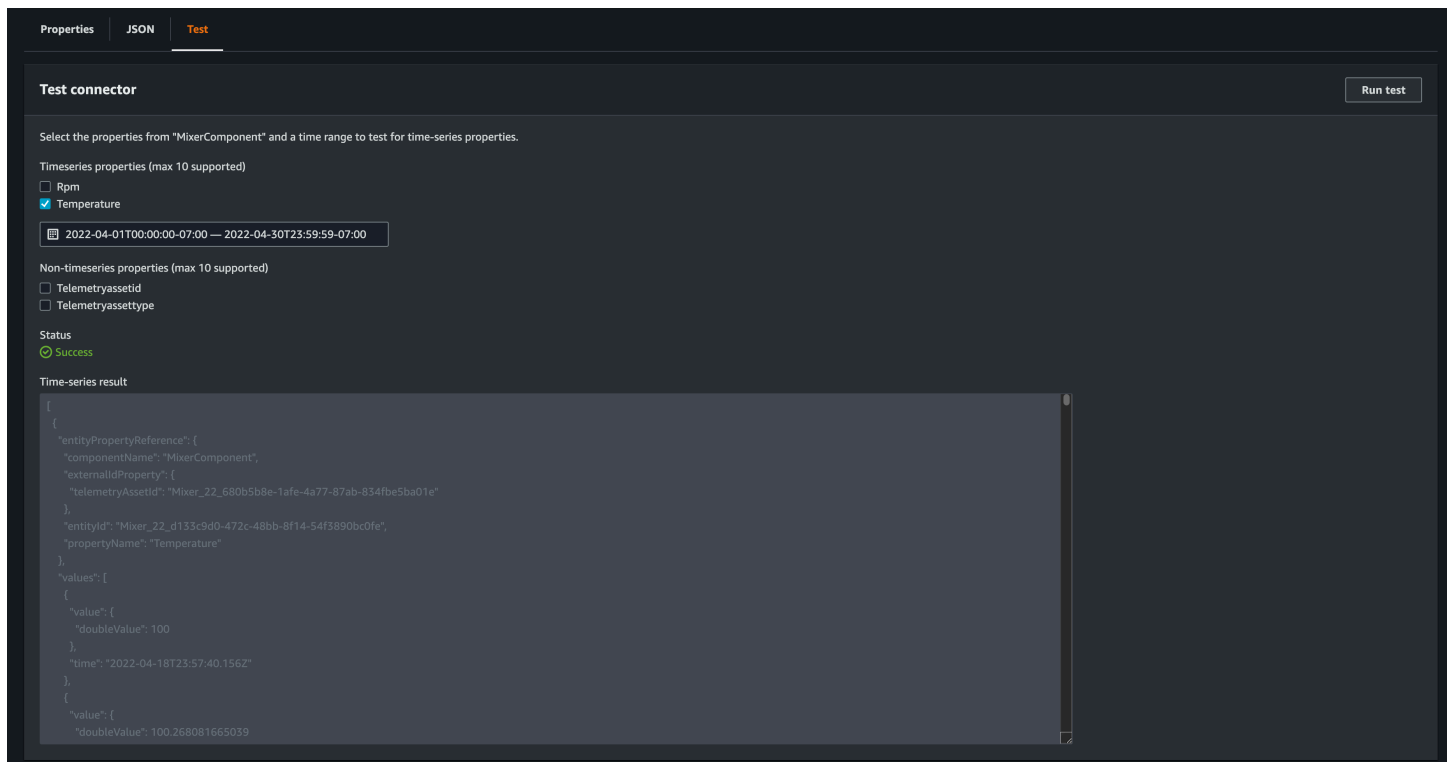
Wir empfehlen, einen integrierten Test durchzuführen AWS IoT TwinMaker , um zu überprüfen, ob die Datenebenenabfrage funktioniert end-to-end. Sie können dies über die [GetPropertyValueHistoryAPI](#) oder einfach in der [AWS IoT TwinMaker Konsole](#) durchführen.



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

Gehen Sie in der AWS IoT TwinMaker Konsole zu den Komponentendetails, und dann werden Sie unter Test sehen, dass dort alle Eigenschaften der Komponente aufgelistet sind. 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 definierte `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 an das Lambda gesendete Beispielanforderung wird im folgenden JSON-Beispiel gezeigt.

```

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

```

```
'endTime': 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 Komponententypeebene 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](#)

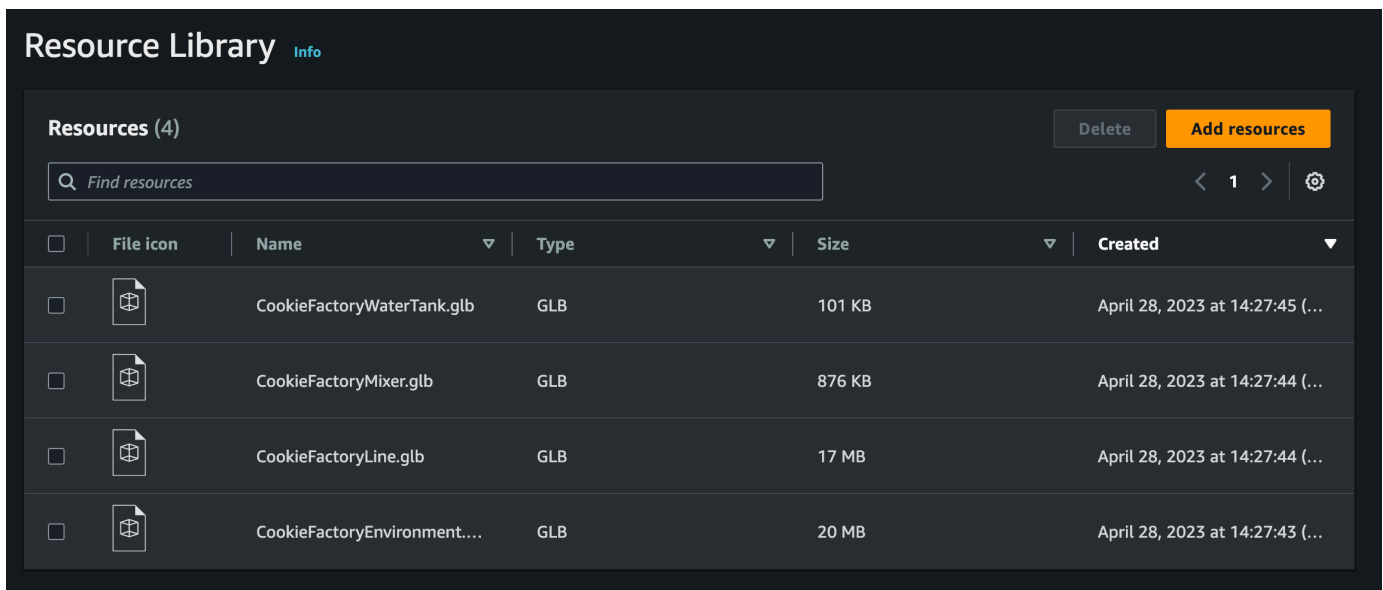
## Laden Sie Ressourcen in die AWS IoT TwinMaker Ressourcenbibliothek hoch

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 Konsole 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

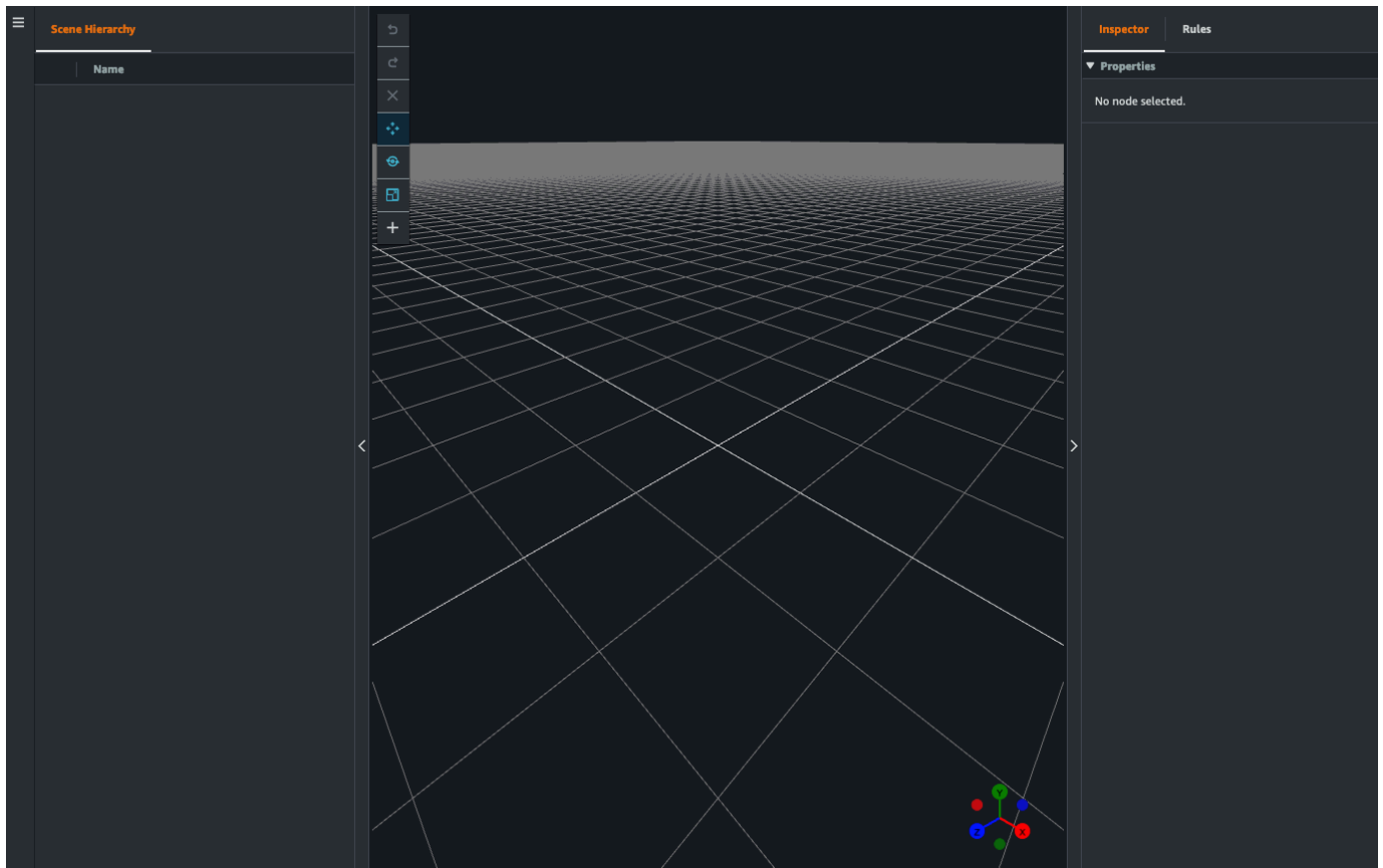
In diesem Abschnitt richten Sie eine Szene ein, damit Sie Ihren digitalen Zwilling bearbeiten können. Sie können ein 3D-Modell importieren, das in die [Ressourcenbibliothek](#) hochgeladen wurde, dann Widgets hinzufügen und Eigenschaftsdaten an Objekte binden, um Ihren digitalen Zwilling zu vervollständigen. 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.

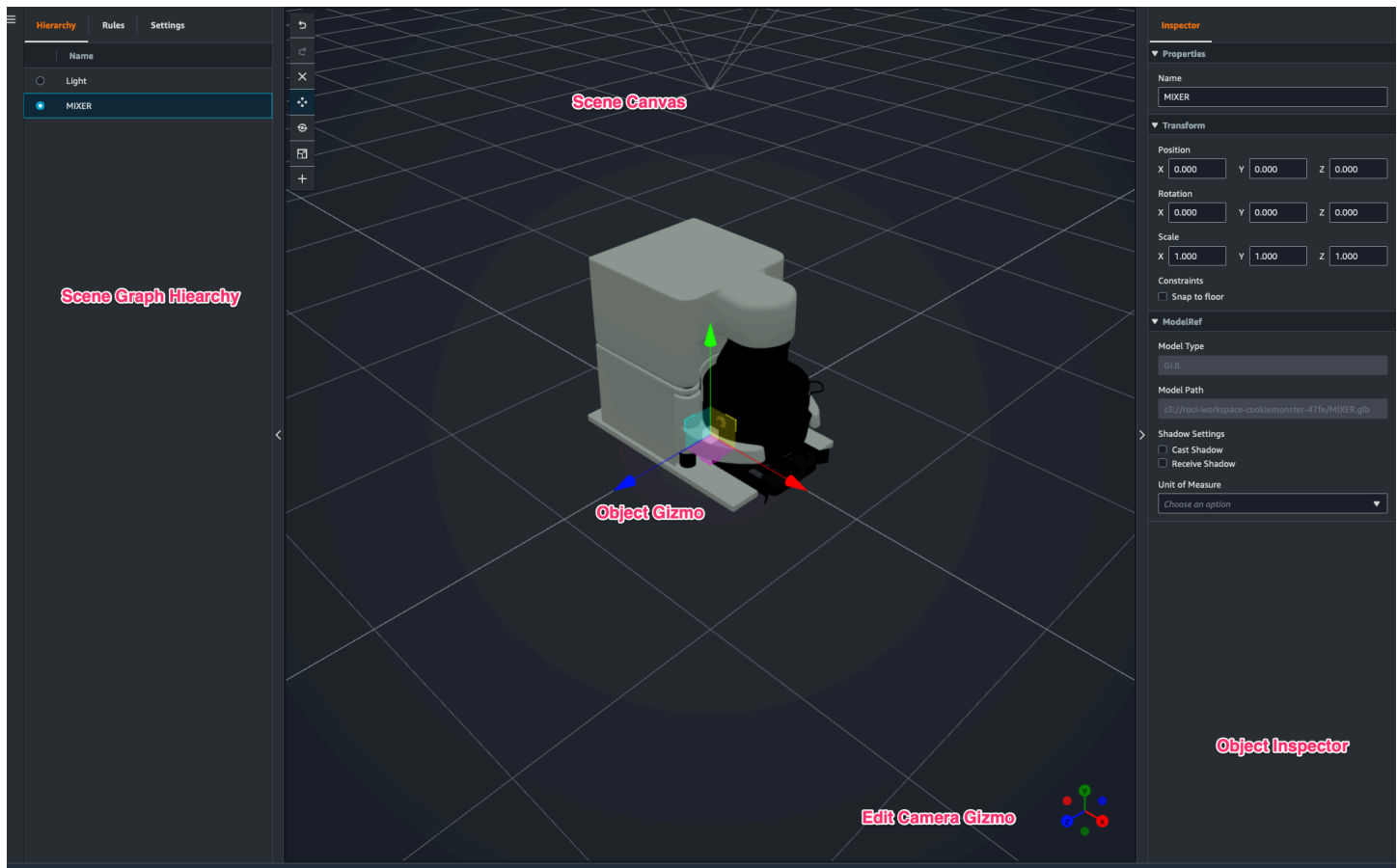


## Verwenden Sie die 3D-Navigation AWS IoT TwinMaker in Ihren In-Szenen

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 Modellshader hinzufügen.
- Navigationsmethode ändern: Erhalten Sie Zugriff auf die Navigationsoptionen der Szenenkamera, Orbit und Pan.
- 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 können.

- [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. Dabei kann es sich um ein Modell, eine Leuchte, ein Etikett oder etwas anderes aus dem Menü „+“ handeln.

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.](#)

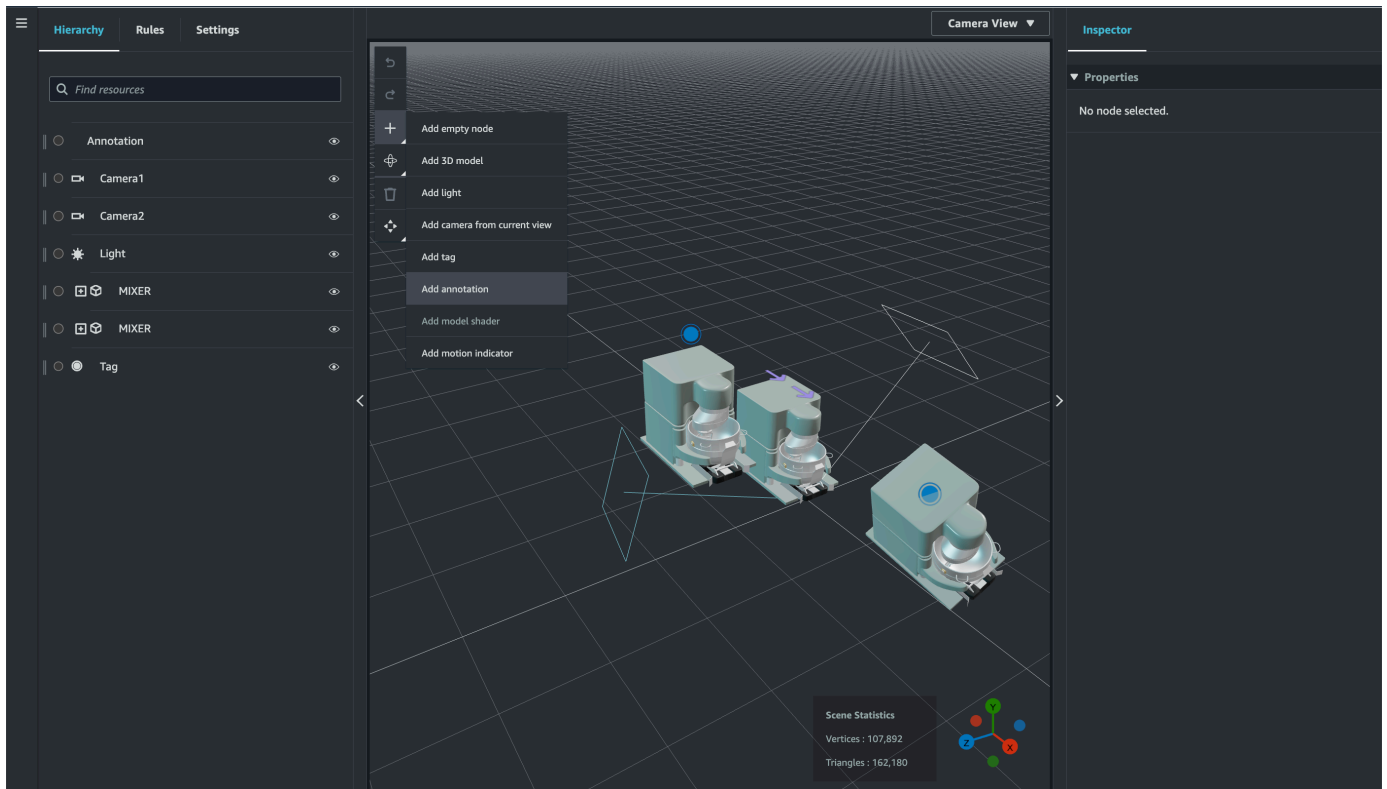
### Note

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

Fügen Sie einer Entität eine Anmerkung hinzu

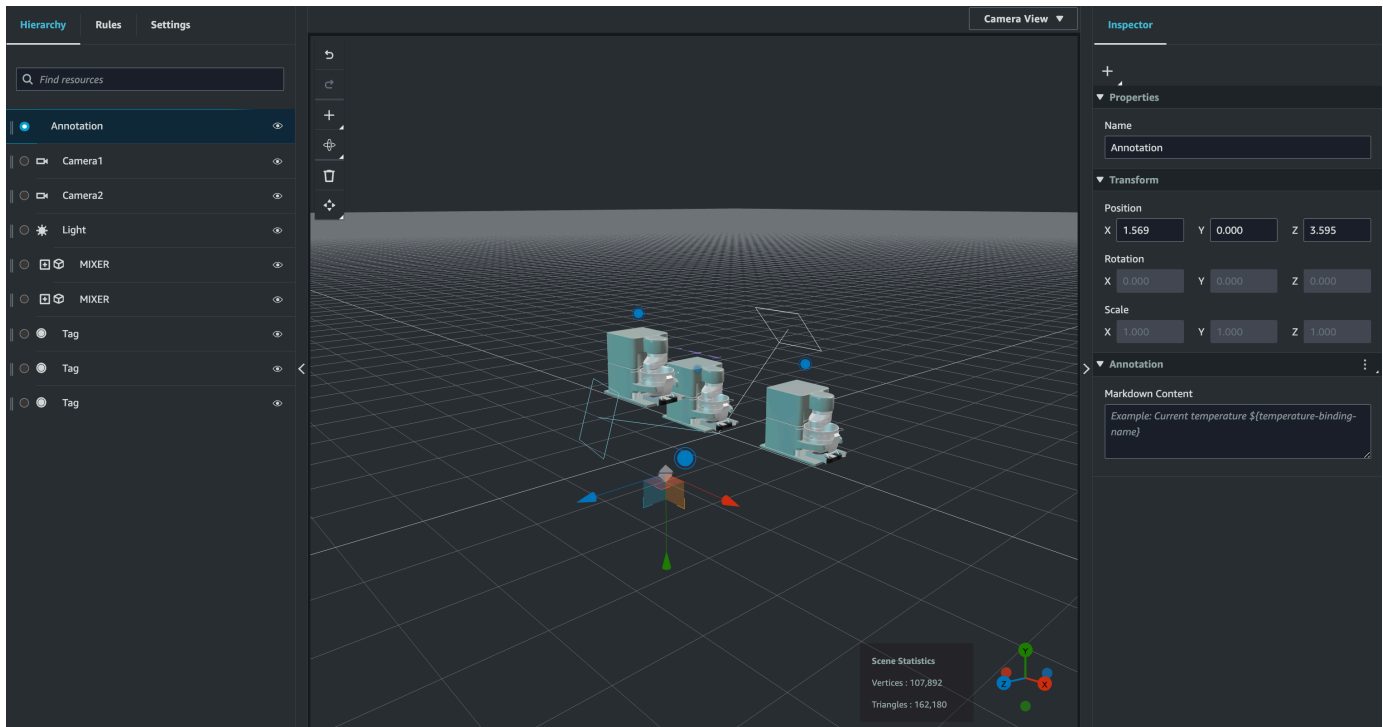
1. Navigieren Sie in der [AWS IoT TwinMaker Konsole](#) zu Ihrer Szene.
2. Wählen Sie ein Element aus der Szenenhierarchie aus, das Sie mit Anmerkungen versehen möchten. Wenn kein Element in der Hierarchie ausgewählt ist, können Sie dem Stamm eine Anmerkung hinzufügen.
3. Drücken Sie das Pluszeichen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

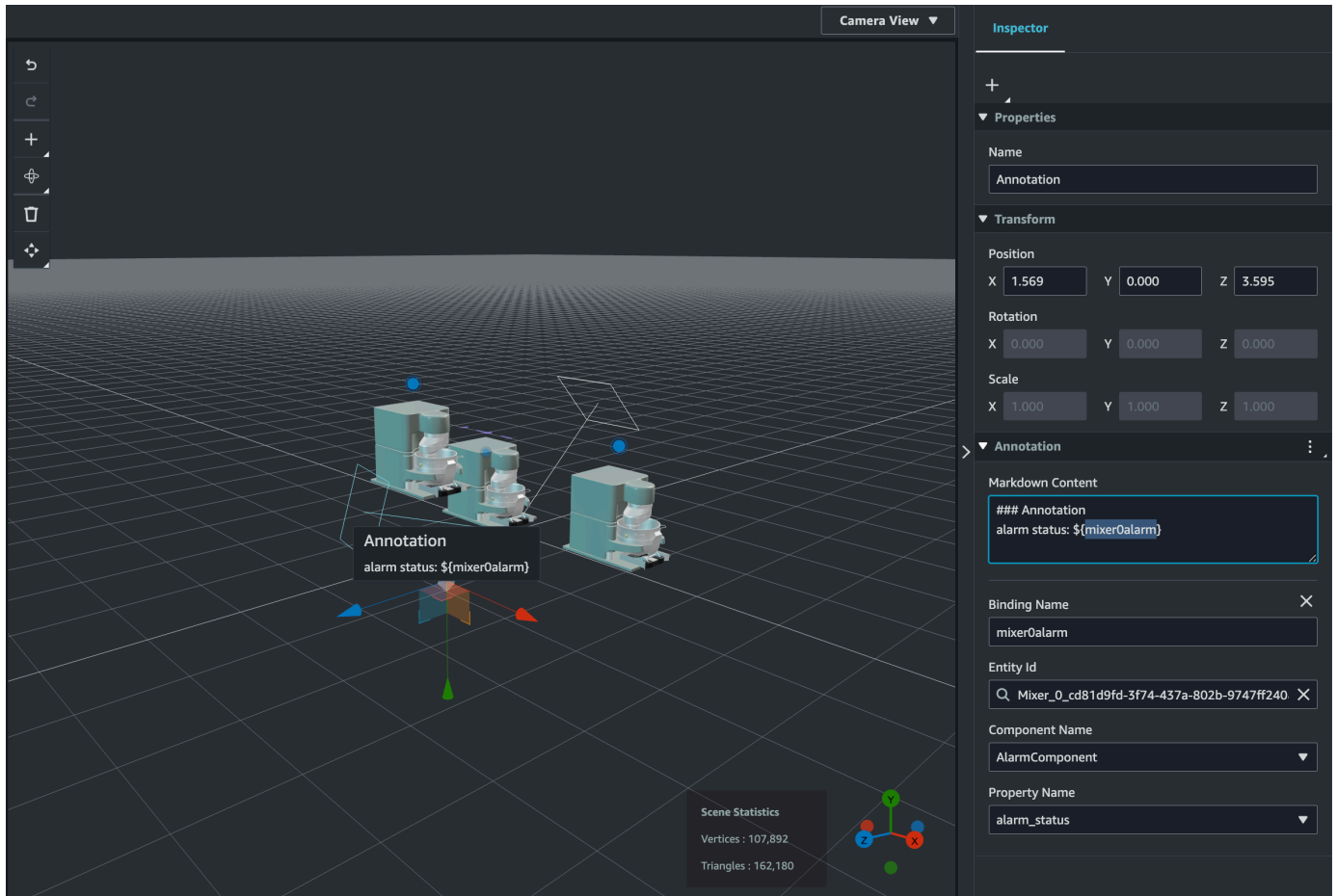
#### Markdown Content

*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.

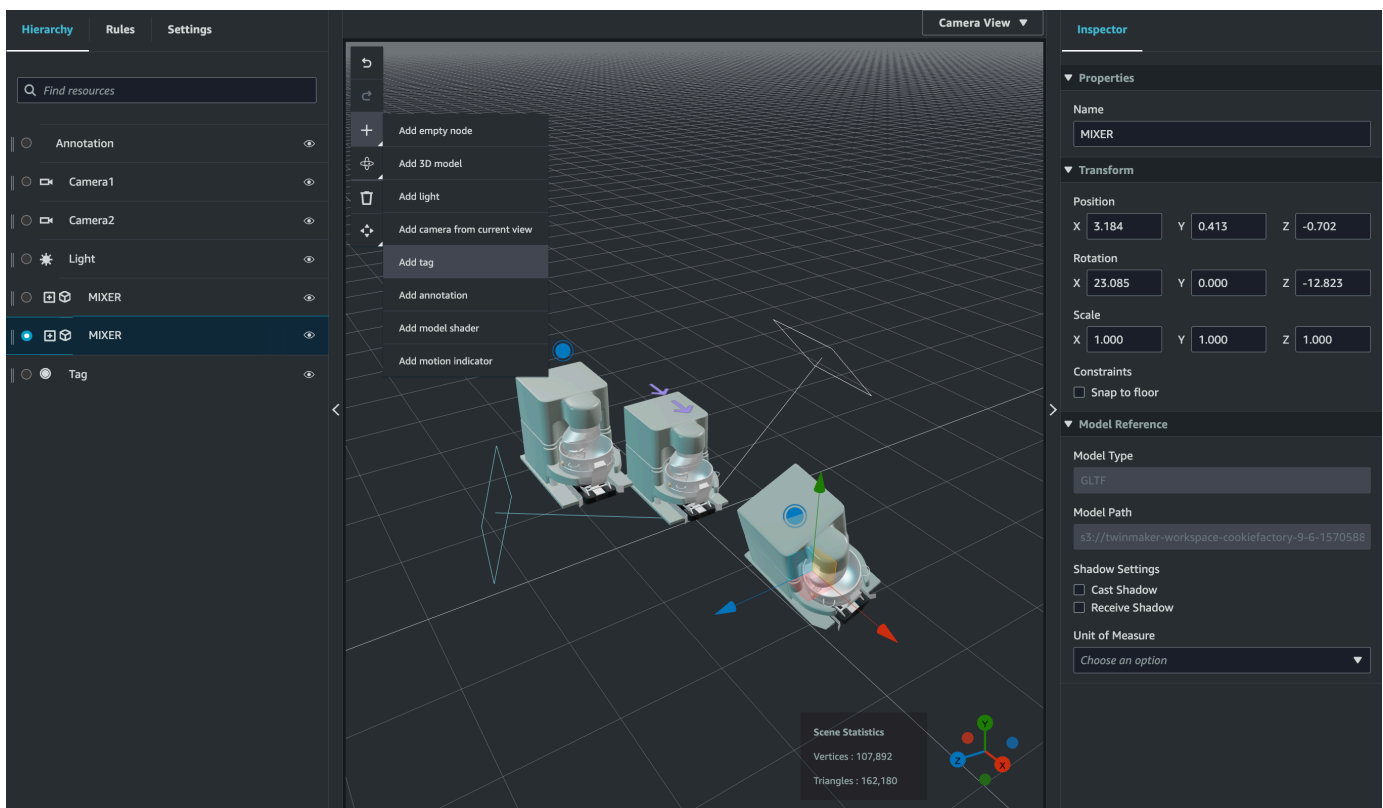
[Weitere Informationen zum Schreiben in Markdown finden Sie in der offiziellen Dokumentation zur Markdown-Syntax, Basic Syntax.](#)

**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 Inspektorenfenster auf der rechten Seite die Schaltfläche + (Plussymbol) und dann Overlay hinzufügen.

## Inspector



Add overlay

Add entity binding

Tag

### Default Icon

Choose an icon



### Entity Id



### Component Name

Select an option



### Property Name

Select an option



### Rule Id

Choose a rule

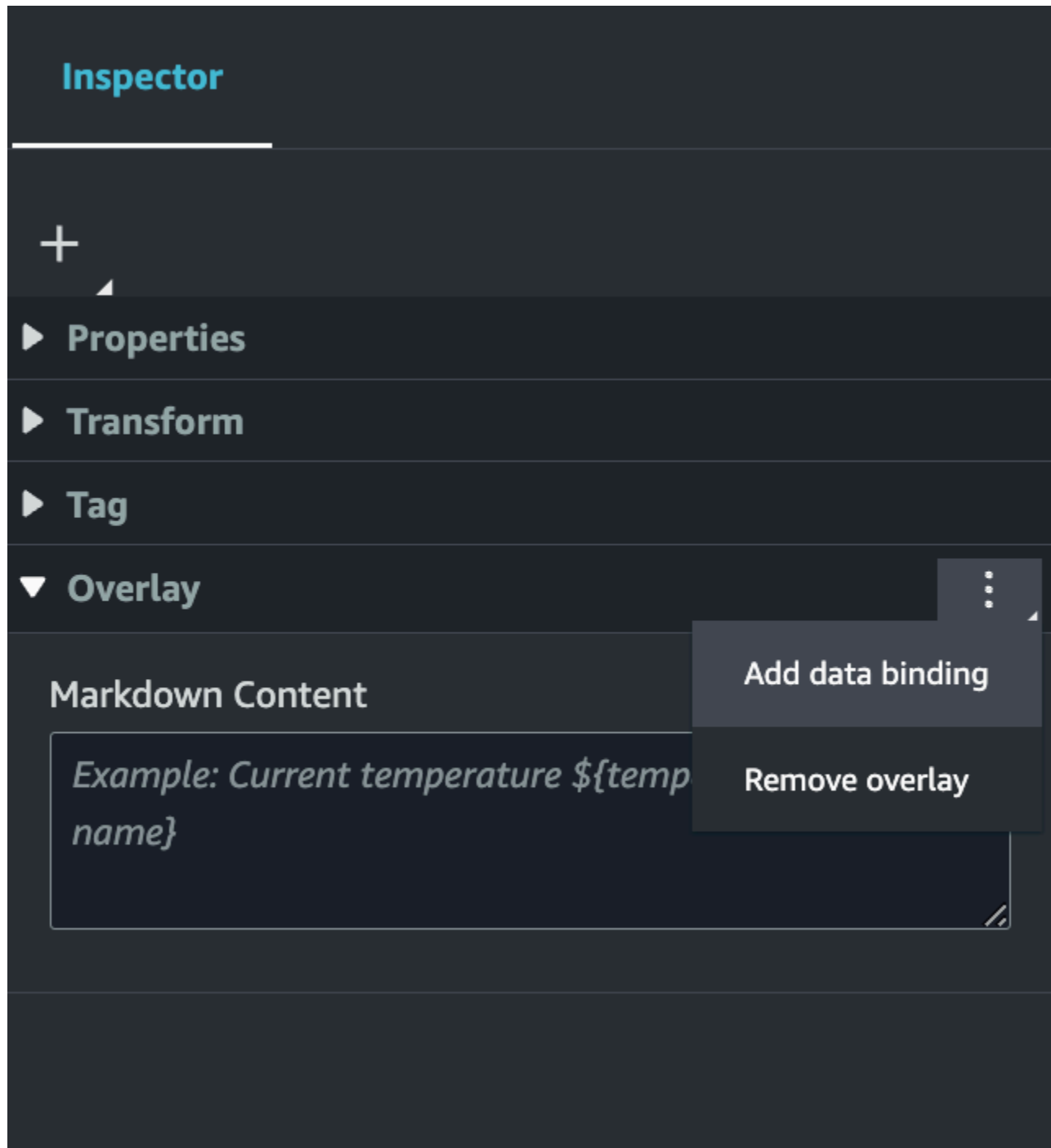


### Link Target

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

- 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

Fügen Sie Überlagerungen zu Tags hinzu

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.

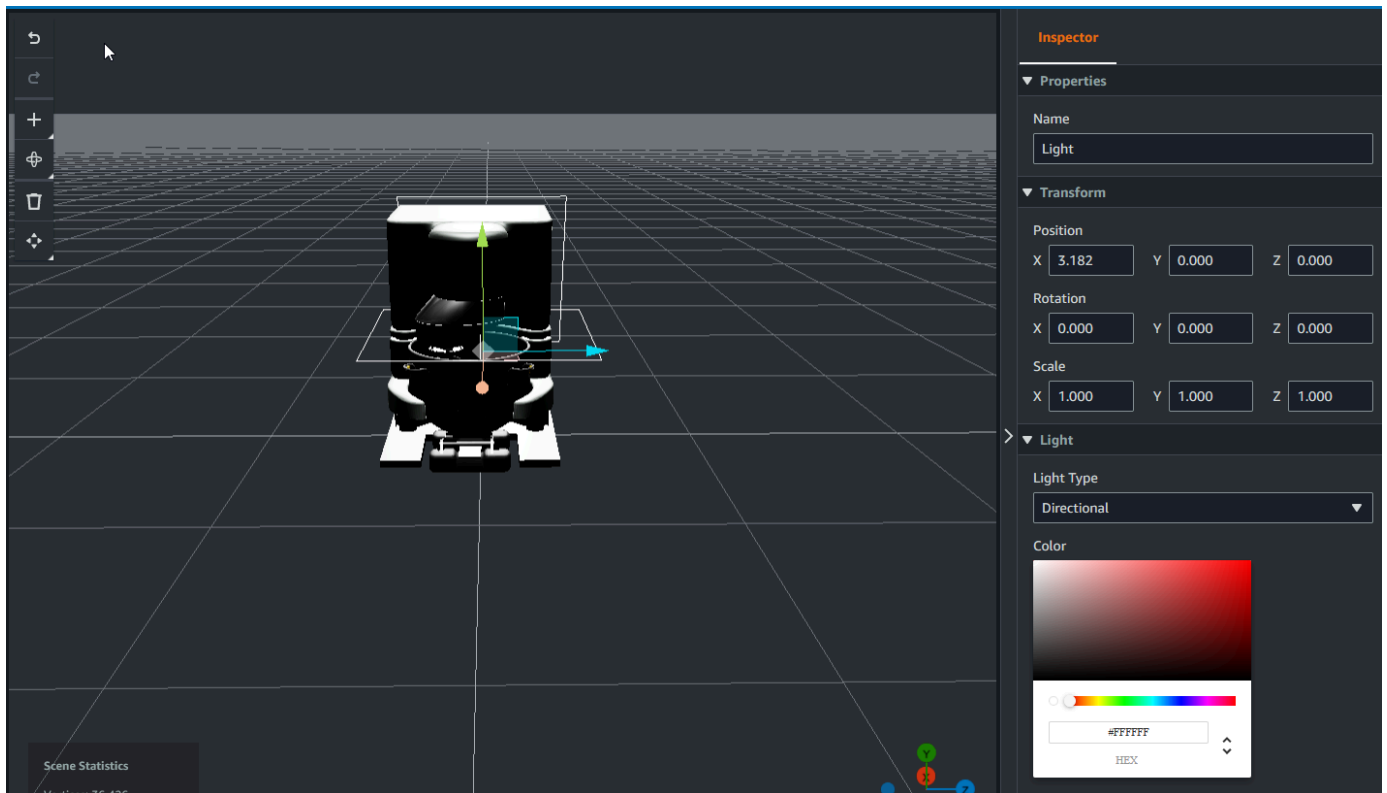
## Fügen Sie Ihren Szenen Modelle hinzu

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.

The screenshot displays the configuration interface for a rule in AWS IoT TwinMaker. It features a dark theme and a vertical list of three statements. Each statement consists of an 'Expression' field, a 'Target' section with an 'Icon' dropdown and a severity dropdown, and a 'Remove statement' button. The first statement has the expression 'temperature >= 40' and a target of 'Error' with a red 'X' icon. The second statement has the expression 'temperature >= 20' and a target of 'Warning' with a yellow exclamation mark icon. The third statement has the expression 'temperature < 20' and a target of 'Info' with a blue circle icon. At the bottom of the rule configuration, there is an 'Add new statement' button, a 'Remove Rule' button, and a section for the rule name 'sampleTimeSeriesColorRule' with a 'Rule Id' field.

Expression

temperature >= 40

Target


Icon ▼ Error ▼ 

Remove statement

Expression

temperature >= 20

Target


Icon ▼ Warning ▼ 

Remove statement

Expression

temperature < 20

Target

Icon ▼ Info ▼ 

Remove statement

Add new statement

Remove Rule

▶ sampleTimeSeriesColorRule

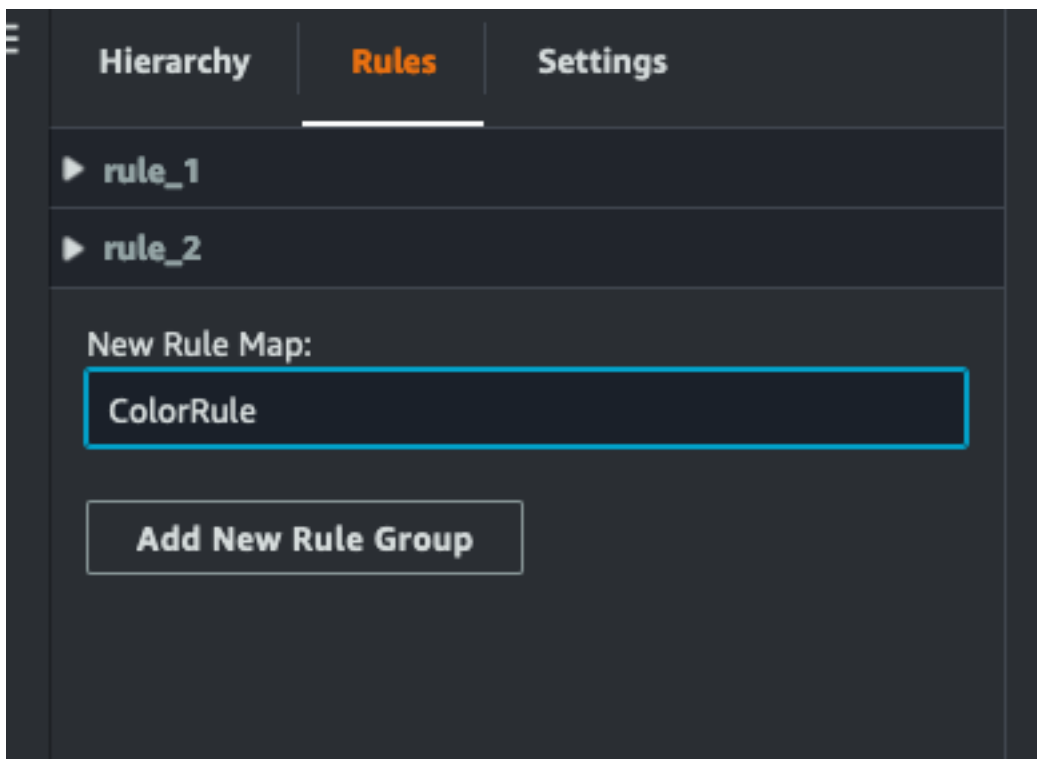
Rule Id



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. #fcb03 (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. Geben Sie dem Tag einen Namen. 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 Cäsium-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. (Weitere Informationen finden Sie unter [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 TypTiles3D.

**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 `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  `angezeigtTiles3D`.

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 die Leistungsfähigkeit des [Knowledge Graph](#), 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

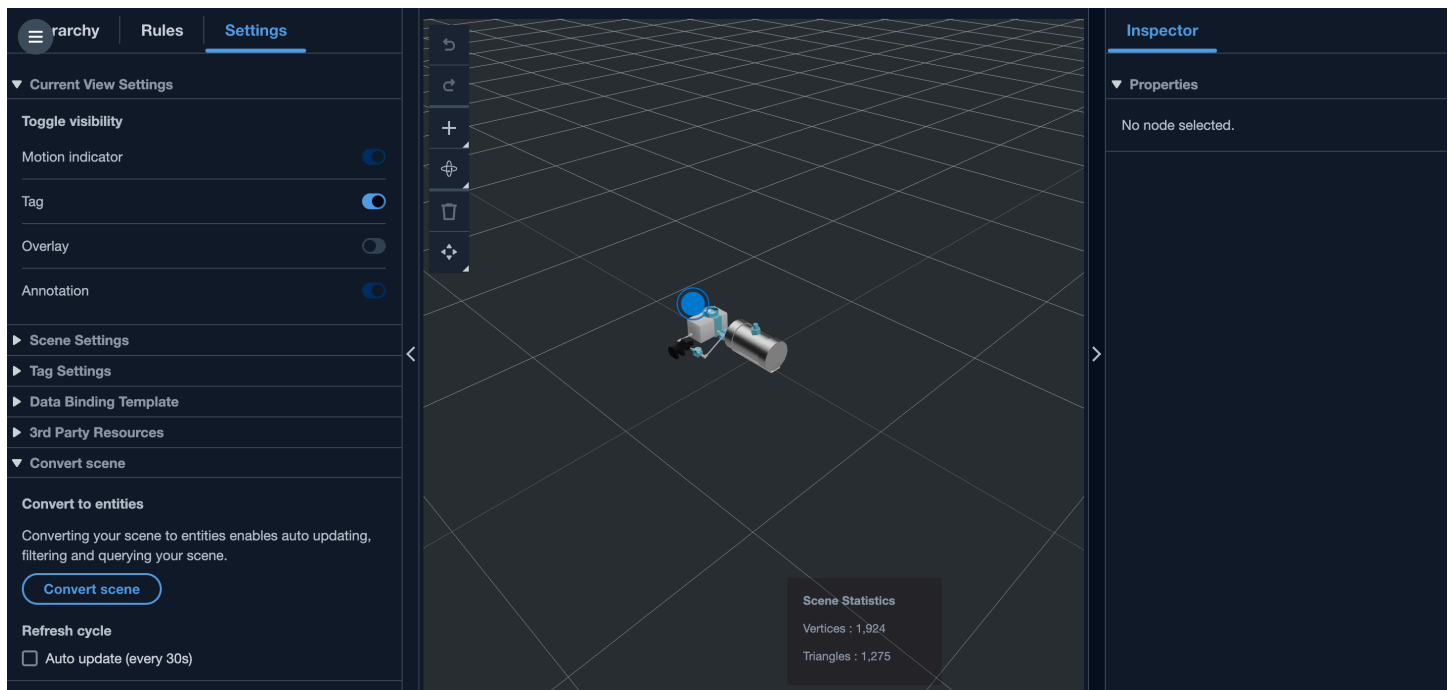
im Knowledge Graph gespeichert werden. Dynamische Szenen werden nur in Standard- und gestaffelten Paket-Preisplänen unterstützt. Informationen [AWS IoT TwinMaker Preismodus wechseln](#) zum Upgrade Ihres Preisplans finden Sie unter).

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 Stamm-Entität 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 einer benutzerdefinierten 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 benutzerdefinierte Webanwendungen mit aktivierter AWS IoT TwinMaker Funktion für ihre digitalen Komprimierungen erstellen.

AWS IoT TwinMaker UI-Komponenten sind Teil des AWS IoT Application Kit, einer clientseitigen Open-Source-Bibliothek, die es IoT-Anwendungsentwicklern ermöglicht, die Entwicklung komplexer IoT-Anwendungen zu vereinfachen

AWS IoT TwinMaker Zu den UI-Komponenten gehören:

- AWS IoT TwinMaker Quelle :

Eine Daten-Connector-Komponente, mit der Sie Daten abrufen und mit Ihren AWS IoT TwinMaker Daten und digitalen Komprimierungen interagieren können.

Weitere Informationen finden Sie in der [AWS IoT TwinMaker Quelldokumentation](#).

- Szenenbetrachter:

Eine auf aufgebaute 3D-Rendering-Komponente@react-three/fiber, die Ihre digitale Komprimierung rendert und Ihnen ermöglicht, damit zu interagieren.

Weitere Informationen finden Sie in der Dokumentation zu [Scene Viewer](#).

- Videoplayer:

Eine Videoplayer-Komponente, mit der Sie ein Video aus den Kinesis Video Streams über 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 [AWS IoT Github-Seite von Application Kit](#).

Anweisungen zum Starten einer neuen Webanwendung mit AWS IoT Application Kit finden Sie auf der offiziellen Dokumentationsseite von [IoT App Kit](#).



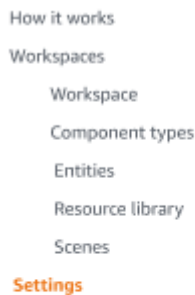
# AWS IoT TwinMaker Preismodus wechseln

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

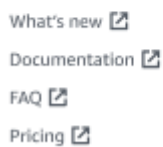
Sie können jederzeit vom nutzungsbasierten zum gestaffelten Preismodell wechseln, die Änderung wird jedoch zu Beginn Ihres nächsten Abrechnungszeitraums wirksam. Sobald Sie vom nutzungsbasierten zum gestaffelten Preismodell gewechselt sind, können Sie für die nächsten drei Nutzungszyklen nicht mehr zum 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.



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



What's new   
Documentation   
FAQ   
Pricing 

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

### Select price mode

**Basic**

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

**Standard (current price mode)**

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

**Tiered bundle**

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

---

### Standard pricing

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

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

Cancel Save

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

#### i 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 Knowledge Graph:

Der Wissensgraph stellt Beziehungen zwischen Entitäten und ihren Komponenten zu den vorhandenen [UpdateEntity](#) APIs [CreateEntity](#) her. Eine Beziehung ist nur eine Eigenschaft eines speziellen Datentyps [RELATIONSHIP](#), die für eine Komponente einer Entität definiert ist. AWS IoT TwinMaker Knowledge Graph ruft die [ExecuteQuery](#) API 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 für eine Komponente einer Entität definierte Beziehungseigenschaft 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 [UpdateEntity](#) API AWS IoT TwinMaker [CreateEntity](#) oder 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

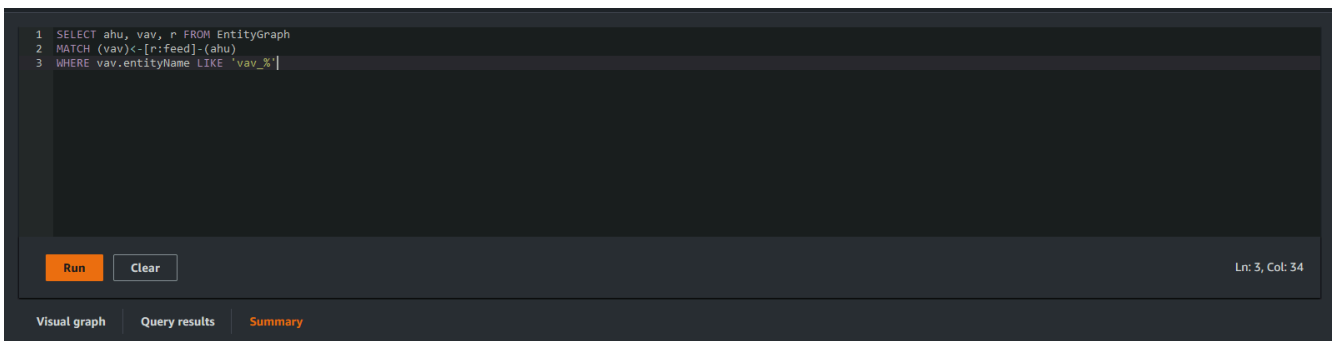
Um zum Knowledge Graph-Abfrage-Editor zu navigieren

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.



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

Run Clear Ln: 3, Col: 34

Visual graph Query results Summary

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.

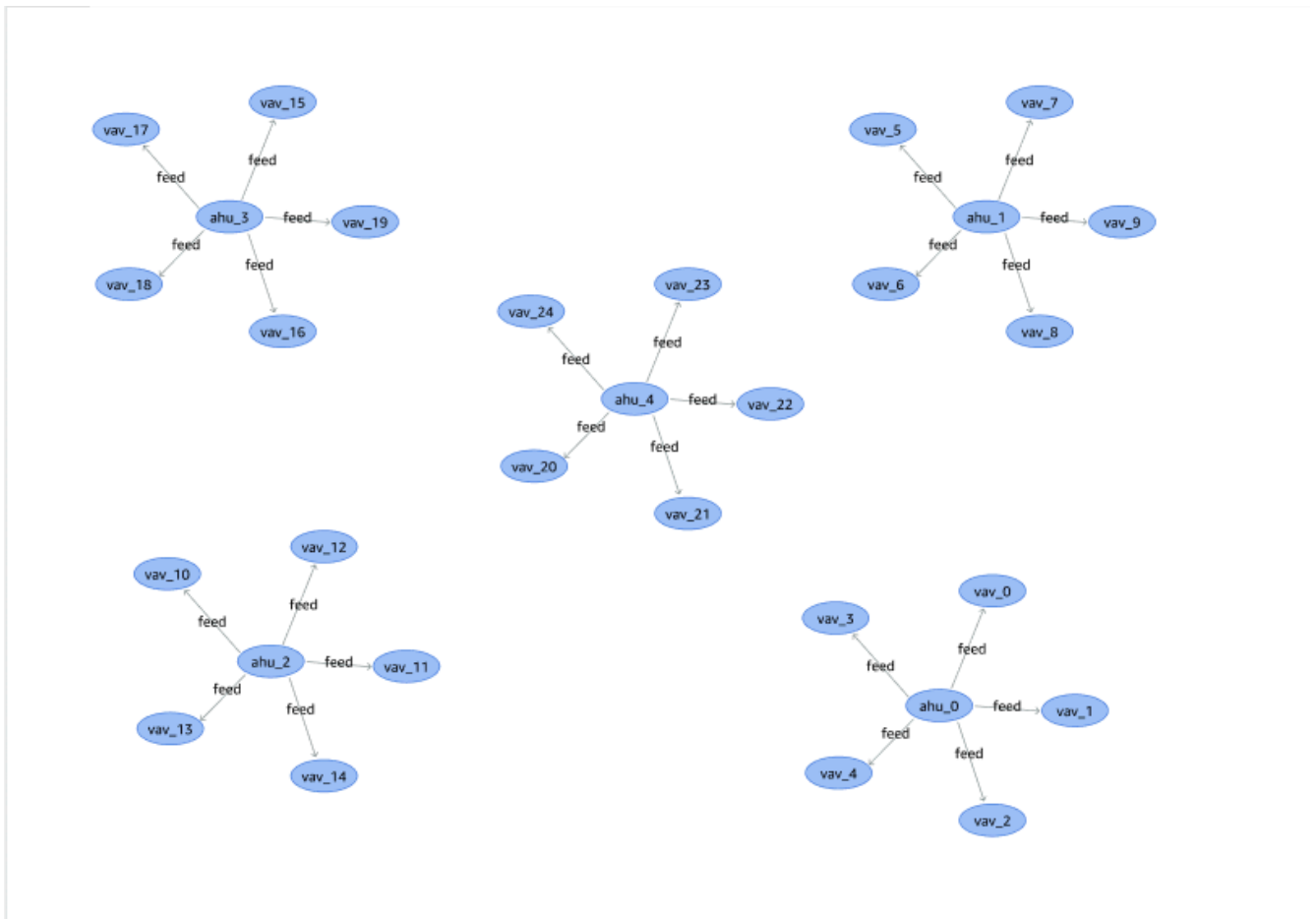
```
SELECT ahu, vav, r FROM EntityGraph
MATCH (vav)<-[r:feed]-(ahu)
WHERE vav.entityName LIKE 'vav_%'
```

**Note**

Die Knowledge Graph-Syntax verwendet [PartiQL](#). Informationen zu dieser Syntax finden Sie unter [AWS 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.



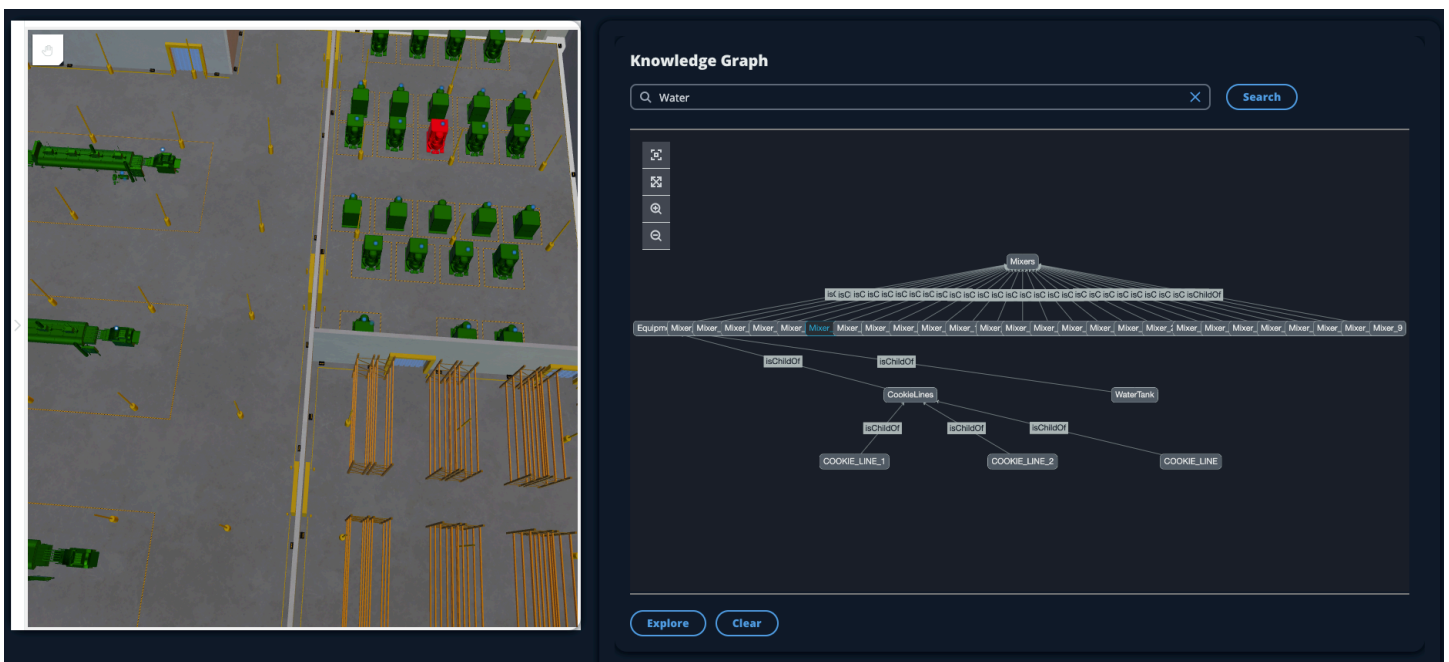
Das oben gezeigte Beispieldiagramm basiert auf dem Abfragebeispiel in Schritt 2.

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 unter [AWS 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 Szenendiagramme

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 unter [Erstellen einer benutzerdefinierten Webanwendung mithilfe von AWS IoT TwinMaker UI-Komponenten](#).
- Machen Sie sich mit den Konzepten und Schlüsselbegriffen von Wissensgraphen 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 unter [AWS IoT TwinMaker Preismodus wechseln](#).

## Binden Sie 3D-Knoten in Ihrer Szene

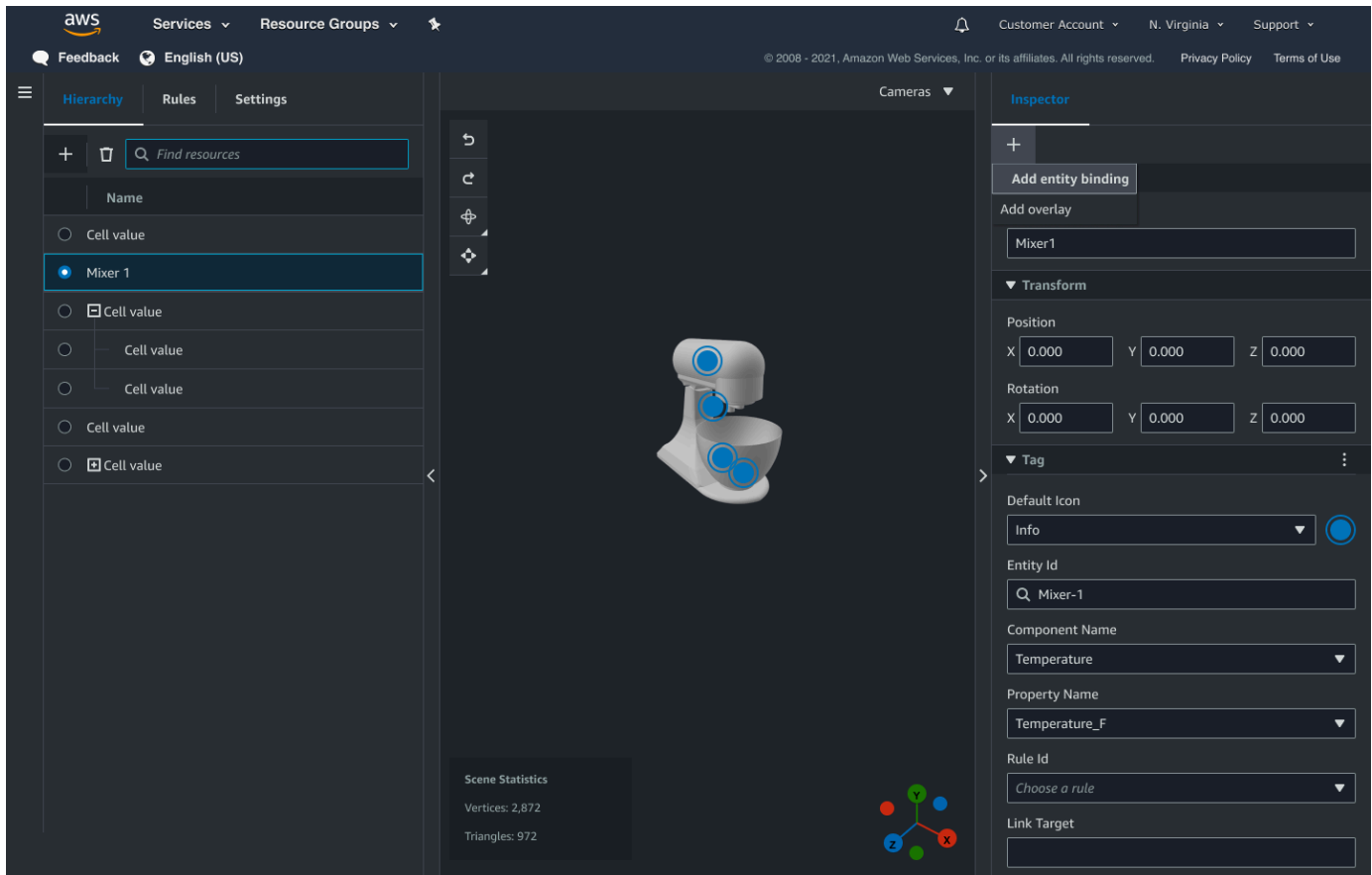
Bevor Sie eine Web-App erstellen, die Knowledge Graph in Ihre Szene integriert, binden Sie die in Ihrer Szene vorhandenen 3D-Modelle, die als 3D-Knoten bezeichnet werden, an die zugehörige Workspace-Entität. Wenn Sie beispielsweise ein Modell einer Mixer-Ausrüstung in einer Szene haben und eine entsprechende 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

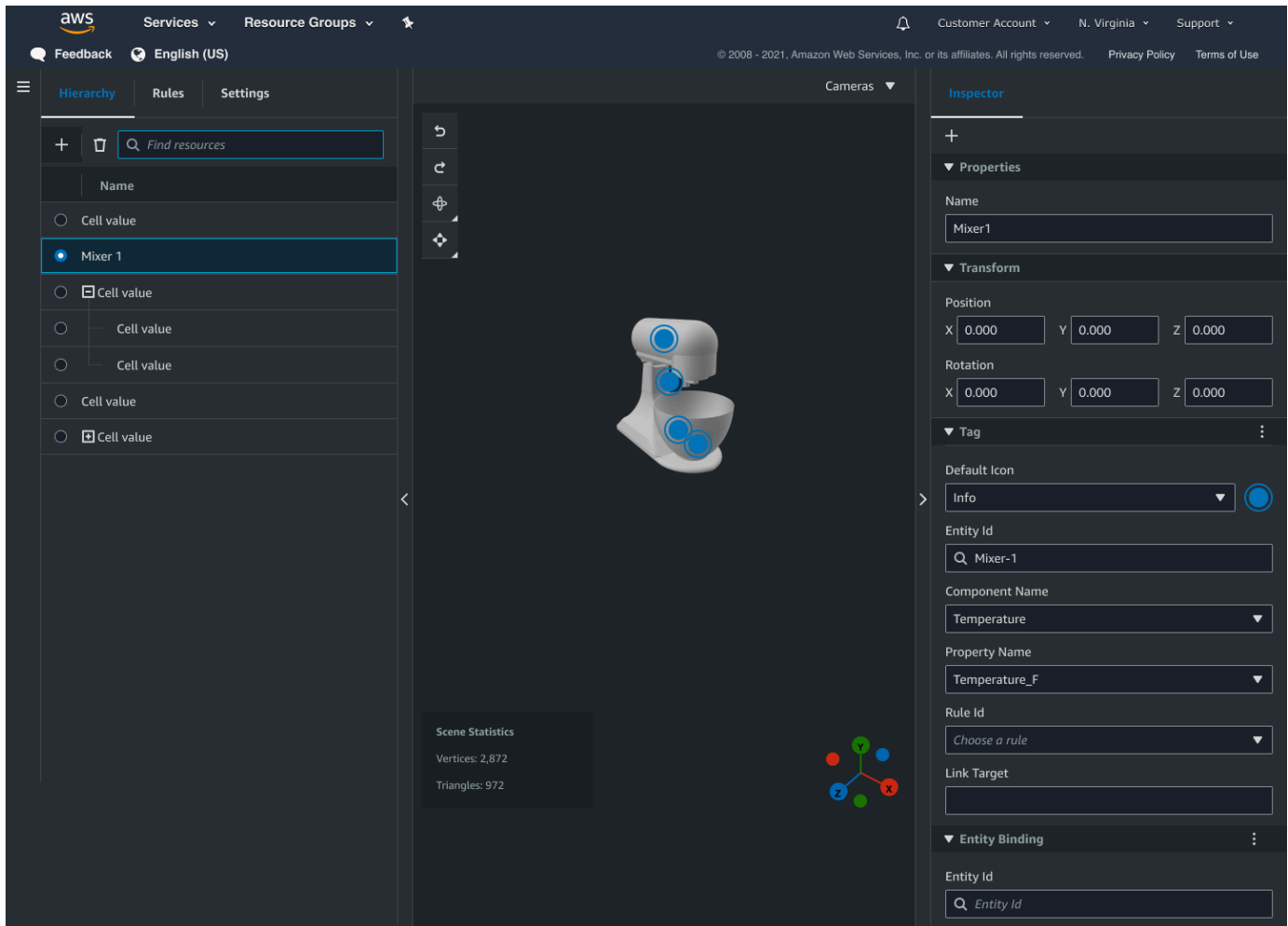
1. Melden Sie sich bei der [AWS IoT TwinMaker -Konsole](#) an.



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 Komponentename und Eigenschaftsname die Komponenten und Eigenschaften aus, die Sie binden möchten.



Sobald Sie eine Auswahl für die Felder Entitäts-ID, Komponentename 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 des AWS IoT App-Kits](#).

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

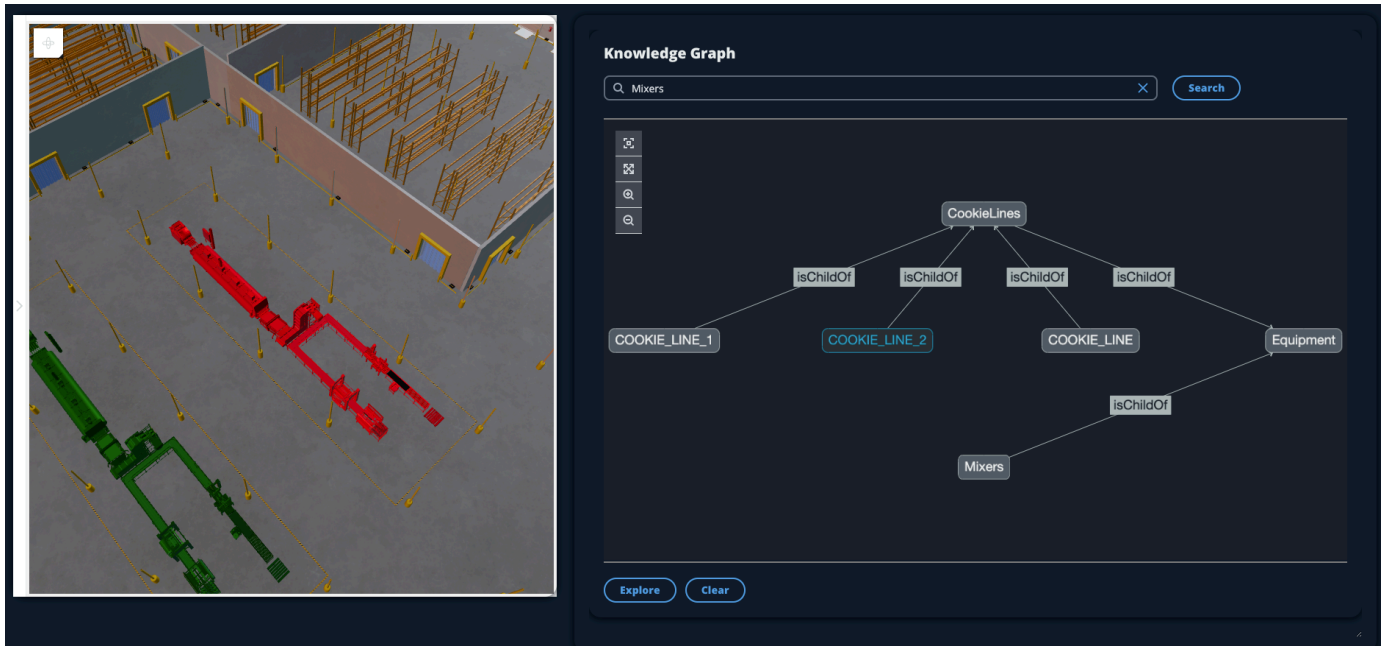
### Note

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.



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



- 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 Abfrageeditor

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. 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:

```
{
  "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.

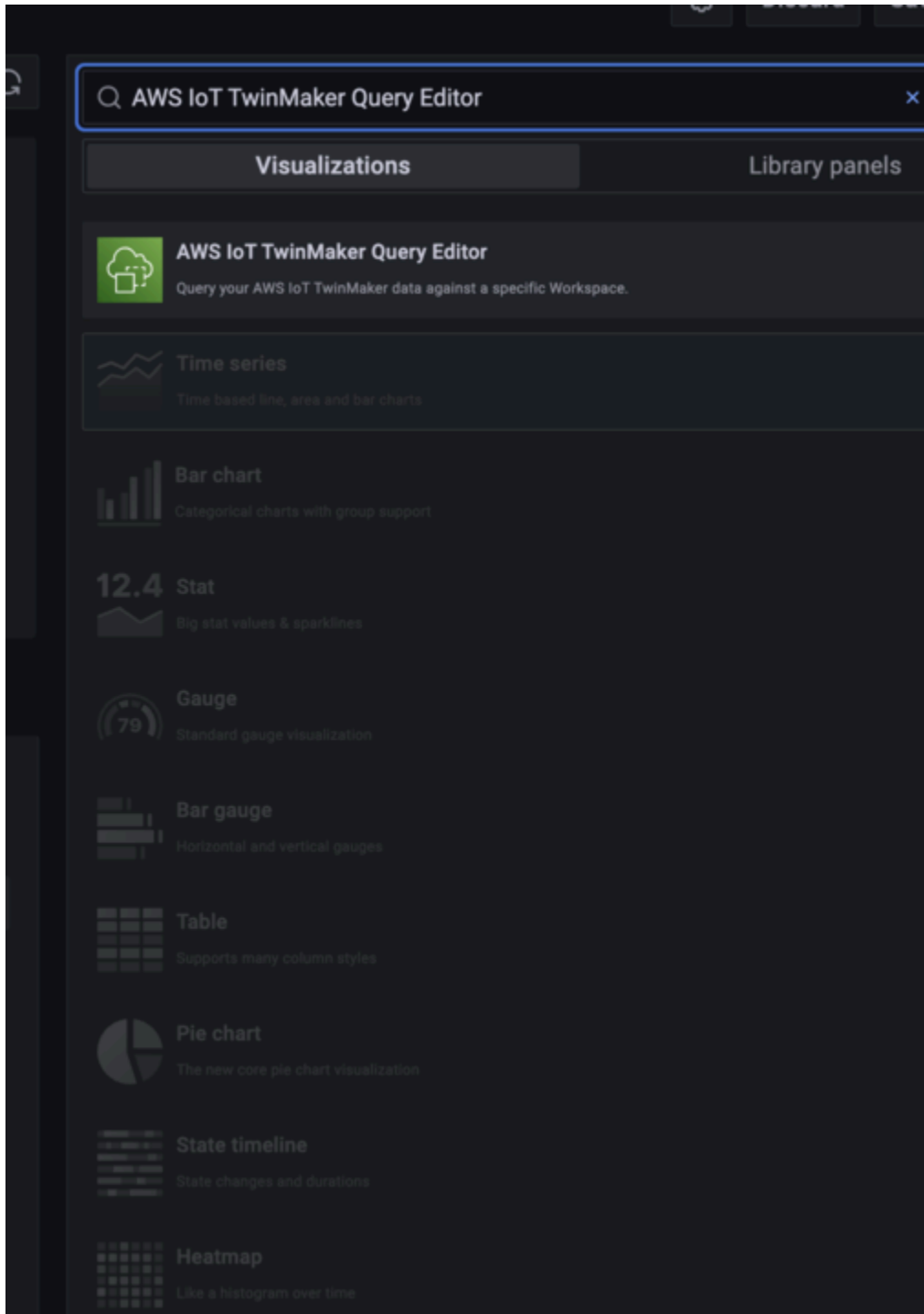
Weitere Informationen finden Sie unter [Erstellen einer Dashboard-IAM-Rolle](#).

## 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.



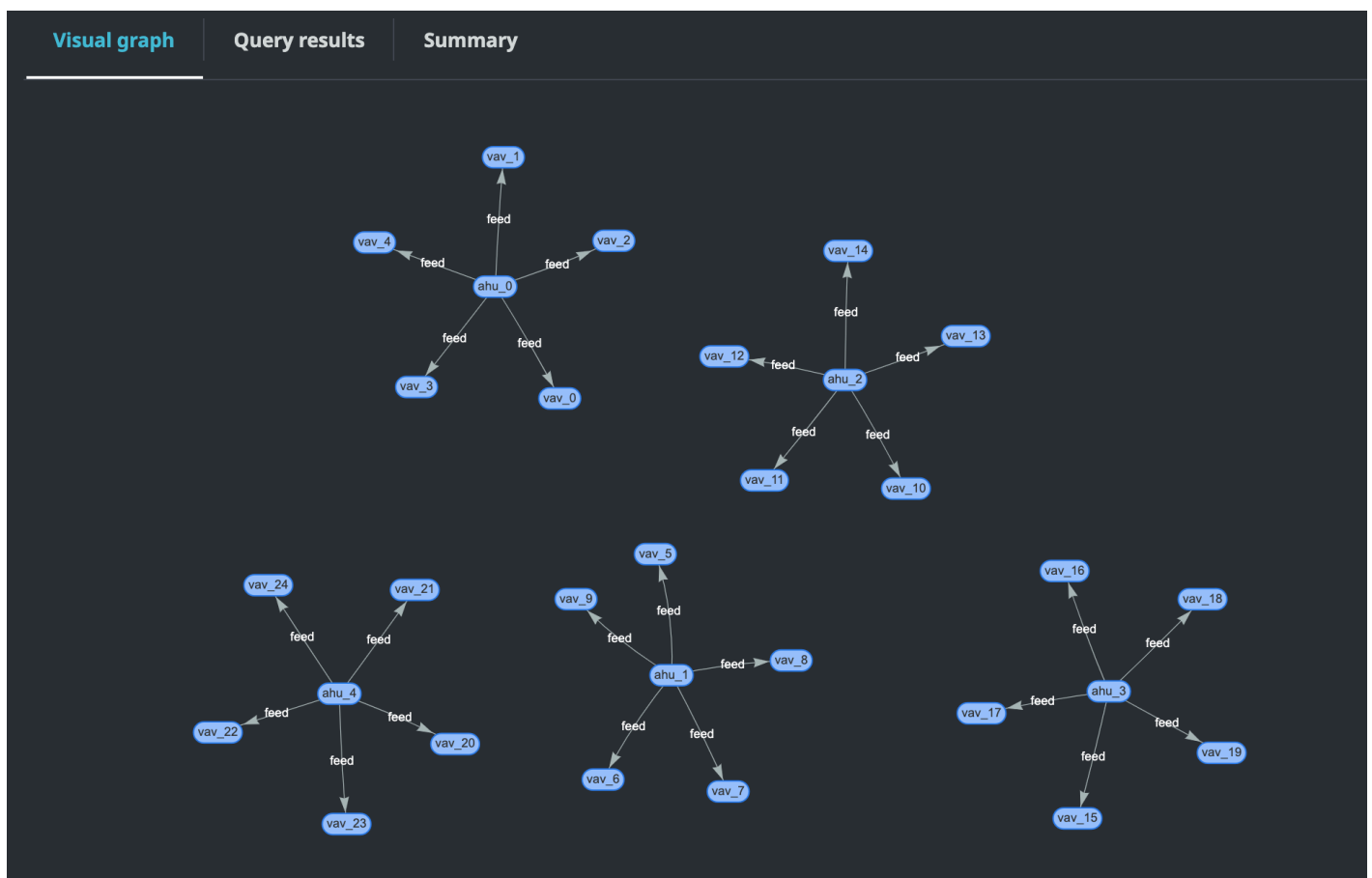
4. Wählen Sie die Datenquelle aus, für die Abfragen ausgeführt werden sollen.
5. (Optional) Fügen Sie im dafür vorgesehenen Feld einen Namen für das neue Panel hinzu.
6. Wählen Sie Anwenden, um Ihr neues Panel zu speichern und zu bestätigen.

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 unter [AWS 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:



Visual graph
Query results
Summary

**Results returned (25)** Export as ▼

< 1 >
⚙️

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

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

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

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 `id` und `name` sowie Felder auf Eigenschaftsebene `properties` wie `componentTypeId` und `componentName`, 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 zurück `isLocationOf`.

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 wird `relationship`.

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

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. Dies `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:

- 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]-
(lightningZone)-[: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 ANDNOT, undOR.
- 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 Ihre Werte '%' enthalten, fügen Sie ein Escape-Zeichen ein LIKE und geben Sie das Escape-Zeichen mit ESCAPE an.

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

## EINDEUTIG:

```
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 jedes Pfads/Musters 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 [ExecuteQuery API-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 (einschließlich 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 Traversierungen (+) 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](#)
- [Das Internet der Dinge nutzen SiteWiseDefaultWorkspace](#)

## Verwenden eines benutzerdefinierten Arbeitsbereichs

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

### Voraussetzungen

Stellen Sie vor der Verwendung sicher AWS IoT SiteWise, dass Sie Folgendes getan haben:

- Sie haben einen AWS IoT TwinMaker Arbeitsbereich.
- Sie haben in einige Assets und Asset-Modelle erstellt AWS IoT SiteWise. Weitere Informationen finden Sie unter [Asset-Modelle erstellen](#).
- Sie haben eine IAM-Rolle erstellt, die über Leseberechtigungen für die folgenden AWS IoT SiteWise Aktionen verfügt: `ListAssets`, `ListAssetModelsDescribeAsset`, und `DescribeAssetModel`.

Die IAM-Rolle muss außerdem über die folgenden Schreibberechtigungen verfügen für `AWS IoT TwinMaker:CreateEntity,UpdateEntity,, DeleteEntity` `CreateComponentTypeUpdateComponentType`, und `DeleteComponentType`

Sie können die folgende IAM-Rolle als Vorlage für die erforderliche Rolle verwenden:

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

// permissions
{
```



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

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.

**Settings**  
Settings for your TwinMaker account

Pricing options | **Model sources**

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

Connector status	Workspace	Last updated
⊖ Disabled	-	-

3. 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.

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

## Add entity model source

Add an entity model source to your workspace.

### Add entity model source

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

AWS IoT SiteWise

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

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

Select IAM role

6. 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.

**cookieFactory** Info
View ▼
Delete

---

**Workspace information**
Edit

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

**Entity model sources (1)**
Add source

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

## Das Internet der Dinge nutzen SiteWiseDefaultWorkspace

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 sicherIoTSiteWiseDefaultWorkspace, 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 die IAM-Konsole unter <https://console.aws.amazon.com/iam/>.

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:

```
{
  "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 `SITWISE_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 `namenssitewiseBase`, 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 `aspropertyId`. 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 `asPropertyId`. 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 - Preisgestaltung](#).

## 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 in synchronisiert AWS IoT SiteWise sind. 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 [ListSyncResources](#)API 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.

The screenshot displays the 'AWS IoT SiteWise source' configuration page. It includes a 'Disconnect' button in the top right corner. The 'Overview' section shows the following details:

- Data Source:** AWS IoT SiteWise
- Role:** syncRole
- Status:** ACTIVE (indicated by a green checkmark icon)
- Status reason:** -
- Date created:** January 20, 1970 at 02:23:23 (UTC-5:00)
- Last modified:** January 20, 1970 at 02:23:23 (UTC-5:00)

Below the overview, a summary shows: Total resources: 8, In Sync: 6 (green checkmark), Error: 2 (red X).

The 'Errors (2)' section contains a search bar and a table with the following data:

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

3. 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 ROOT 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.

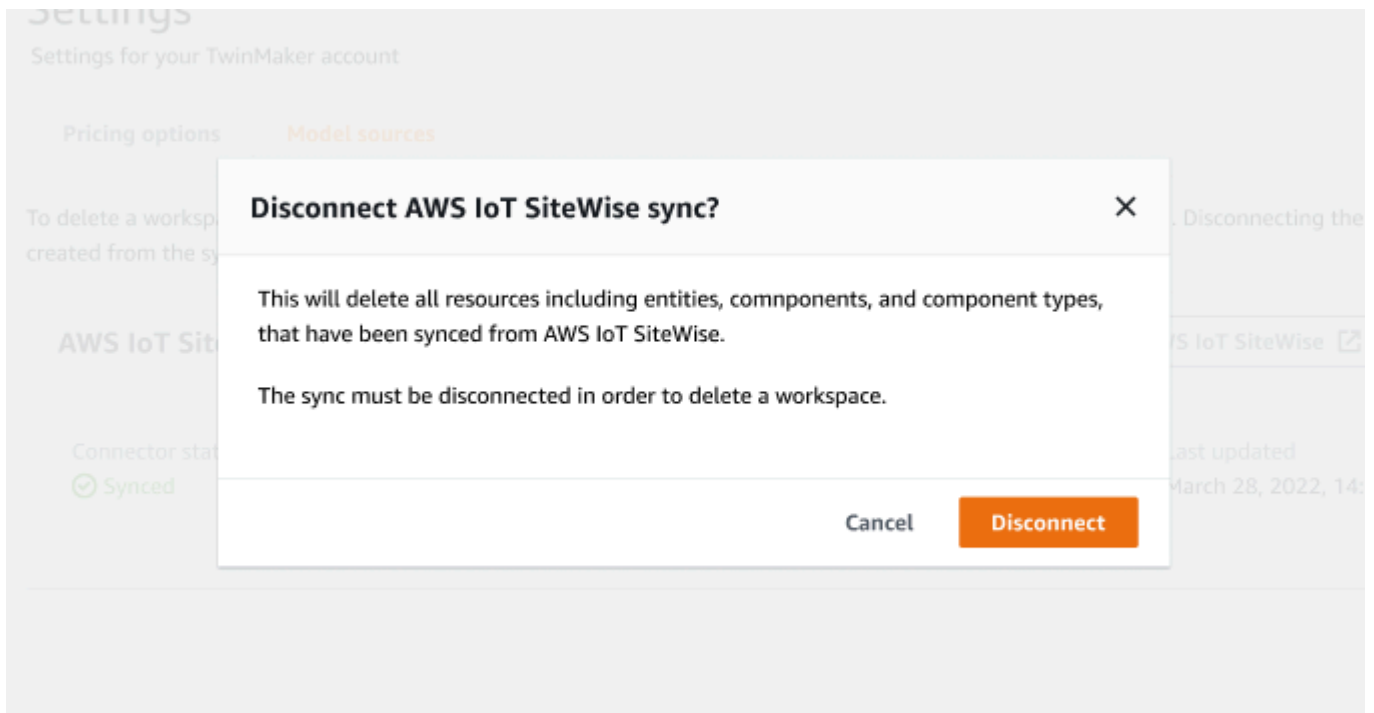
# Löschen Sie einen Synchronisierungsauftrag

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, falls 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 [GetEntity](#)API gibt nur die ersten 50 untergeordneten Entitäten für eine Hierarchieeigenschaft zurück, aber Sie können die [GetPropertyValue](#)API 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. Sie können Grafana Version 8.2.0 und höher verwenden, um mit Ihrer digitalen Zwangsanwendung zu interagieren. Das AWS IoT TwinMaker Plug-In bietet benutzerdefinierte Bereiche, Dashboard-Vorlagen und eine Datenquelle, um eine Verbindung zu Ihren digitalen Wolkendaten herzustellen.

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

## Themen

- [CORS-Konfiguration für Grafana-Szenenbetrachter](#)
- [Einrichten Ihrer Grafana-Umgebung](#)
- [Erstellen einer Dashboard-IAM-Rolle](#)
- [Erstellen einer AWS IoT TwinMaker -Videoplayer-Richtlinie](#)

### Note

Sie müssen die CORS-Konfiguration (Cross-Origin Resource Sharing) des Amazon S3-Buckets ändern, damit die Grafana-Benutzeroberfläche Ressourcen aus dem Bucket laden kann. Die Anweisungen finden Sie unter [CORS-Konfiguration für Grafana-Szenenbetrachter](#).

Weitere Informationen zum AWS IoT TwinMaker Grafana-Plugin finden Sie in der [AWS IoT TwinMakerApp](#)-Dokumentation.

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

- [AWS IoT TwinMaker Datenquelle](#)
- [Dashboard-Vorlagen](#)
- [Szenenanzeige-Bereich](#)
- [Video-Player-Bereich](#)

# CORS-Konfiguration für Grafana-Szenenbetrachter

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-Bucket zu laden. Ohne die CORS-Konfiguration erhalten Sie die Fehlermeldung „Load 3D Scene failed with Network Failure“ im Scene-Viewer, da die Grafana-Domain nicht auf die Ressourcen im Amazon S3-Bucket zugreifen kann.

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

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

## Note

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

```
{
  [
    "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-Buckets finden Sie unter [Cross-Origin Resource Sharing \(CORS\) verwenden](#).

## Einrichten 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 für Ihre Anforderungen schnell bereitstellen, betreiben und skalieren. Alternativ können Sie Ihre eigene Infrastruktur zur Verwaltung von Grafana-Servern einrichten.

Weitere Informationen zu den 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, mit dem Sie schnell AWS IoT TwinMaker in Grafana integrieren können. Da Amazon Managed Grafana Grafana-Server für Sie verwaltet, können Sie Ihre Daten visualisieren, ohne Hardware oder eine 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-Plugins.

## Voraussetzungen für Amazon Managed Grafana

Um AWS IoT TwinMaker in einem Amazon Managed Grafana-Dashboard zu verwenden, müssen Sie zunächst die folgenden Voraussetzungen erfüllen:

- Erstellen Sie einen -AWS IoT TwinMakerWorkspace. 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 -AWSManagementkonsole erstellen, AWS IoT TwinMaker wird nicht aufgeführt. Das Plugin ist jedoch bereits auf allen Workspaces installiert. Sie finden das AWS IoT TwinMaker Plugin in der Liste der Open-Source-Grafana-Plugins. 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. Es 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 nichtAWS IoT TwinMaker, daher müssen Sie diese Berechtigungen manuell einrichten. Weitere Informationen zum Einrichten manueller Berechtigungen finden Sie unter [Erstellen einer Dashboard-IAM-Rolle](#).

## Selbstverwaltetes Grafana

Sie können Ihre eigene Infrastruktur hosten, um Grafana auszuführen. Informationen zum lokalen Ausführen von Grafana auf Ihrem Computer finden Sie unter [Installieren von Grafana](#). Das AWS IoT TwinMaker Plugin ist im öffentlichen Grafana-Katalog verfügbar. Informationen zum Installieren dieses Plugins in Ihrer Grafana-Umgebung finden Sie unter [AWS IoT TwinMaker -App](#).

Wenn Sie Grafana lokal ausführen, können Sie Dashboards nicht einfach freigeben oder mehreren Benutzern Zugriff gewähren. Eine Kurzstartanleitung zum Freigeben von Dashboards mit lokalem Grafana finden Sie unter [AWS IoT TwinMaker Beispiel-Repository](#). Diese Ressource führt Sie durch das Hosten einer Grafana-Umgebung auf Cloud9 und Amazon EC2 auf einem öffentlichen Endpunkt.

Sie müssen bestimmen, welchen Authentifizierungsanbieter Sie für die Konfiguration von TwinMaker Datenquellen verwenden werden. Sie konfigurieren die Anmeldeinformationen für die Umgebung basierend auf der Standard-Anmeldeinformationskette (siehe [Verwenden der Standard-Anbieterkette für Anmeldeinformationen](#)). Die Standardanmeldeinformationen können die dauerhaften Anmeldeinformationen eines beliebigen Benutzers oder einer beliebigen Rolle sein. Wenn Sie beispielsweise Grafana auf Amazon EC2 ausführen, hat die standardmäßige Anmeldeinformationskette Zugriff auf die [Amazon EC2-Ausführungsrolle](#), die dann Ihr Authentifizierungsanbieter wäre. Der IAM-Amazon-Ressourcenname (ARN) des Authentifizierungsanbieters ist in den Schritten bis erforderlich [Erstellen einer Dashboard-IAM-Rolle](#).

## Erstellen einer Dashboard-IAM-Rolle

Mit können AWS IoT TwinMaker Sie den Datenzugriff auf Ihre Grafana-Dashboards steuern. Grafana-Dashboard-Benutzer sollten unterschiedliche Berechtigungsbereiche haben, um Daten anzuzeigen und in einigen Fällen Daten zu schreiben. Beispielsweise verfügt ein Alarmoperator möglicherweise nicht über die Berechtigung zum Anzeigen von Videos, während ein Administrator über die Berechtigung für alle Ressourcen verfügt. 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 um reduziert werden kann AWS IoT TwinMaker.

Um Ihre AWS IoT TwinMaker Dashboards in Grafana zu verwenden, erstellen Sie eine IAM-Rolle und fügen Richtlinien an. Sie können die folgenden Vorlagen verwenden, um diese Richtlinien zu erstellen.

### Eine IAM-Richtlinie erstellen

Erstellen Sie eine IAM-Richtlinie mit dem Namen *YourWorkspaceId*DashboardPolicy in der IAM-Konsole. Diese Richtlinie gewährt Ihren Workspaces Zugriff auf Amazon S3-Bucket und -AWS IoT TwinMaker Ressourcen. Sie können auch den [AWS IoT Greengrass Edge Connector für Amazon Kinesis Video Streams](#) verwenden, der Berechtigungen für die Kinesis Video Streams und

für die Komponente konfigurierten AWS IoT SiteWise Komponenten erfordert. Wählen Sie eine der folgenden Richtlinienvorlagen aus, um sie an Ihren Anwendungsfall anzupassen.

## 1. Keine Richtlinie für Videoberechtigungen

Wenn Sie den Grafana-[Video-Player-Bereich](#) nicht verwenden möchten, erstellen Sie die Richtlinie mit der folgenden Vorlage.

```
{
  "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-Bucket erstellt. Es enthält die 3D-Modelle und Szenen, die auf einem Dashboard angezeigt werden sollen. Das [SceneViewer](#) Panel lädt Elemente aus diesem Bucket.

## 2. Eingrenzung der Richtlinie für Videoberechtigungen

Um den Zugriff auf den Video-Player-Bereich in Grafana einzuschränken, gruppieren Sie Ihren AWS IoT Greengrass Edge Connector für Amazon Kinesis Video Streams-Ressourcen nach Tags. Weitere Informationen zum Heruntergrenzen von Berechtigungen für Ihre Videoressourcen finden Sie unter [Erstellen einer AWS IoT TwinMaker -Videoplayer-Richtlinie](#).

## 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 schreibgeschützten Zugriff auf jede AWS IoT SiteWise Komponente haben. Dies schließt alle Ressourcen ein, die in der Zukunft erstellt werden.

Erstellen Sie die Richtlinie mit der folgenden Vorlage:

```
{
  "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 zum Streamen aller Videos von Kinesis Video Streams in Ihrem Konto.

- Schreibgeschützter Zugriff auf den Eigenschaftswertverlauf aller AWS IoT SiteWise Komponenten in Ihrem Konto.
- Datenaufnahme in eine beliebige Eigenschaft einer AWS IoT SiteWise Komponente, die mit dem Schlüssel `EdgeConnectorForKVS` und dem Wert gekennzeichnet ist `workspaceId`.

## Markieren des Video-Uploads Ihrer AWS IoT SiteWise Kamerakomponente von Edge

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 jede AWS IoT SiteWise Komponente aktivieren, die Ihrem AWS IoT Greengrass Edge Connector für Amazon Kinesis Video Streams zugeordnet ist und mit dem Schlüssel gekennzeichnet ist `istEdgeConnectorForKVS`.

Der Tag-Wert kann eine Liste von `workspacelds` sein, die durch eines der folgenden Zeichen getrennt sind: `.` `:` `+` `=` `@` `_` `/` `-`. Wenn Sie beispielsweise eine AWS IoT SiteWise Komponente verwenden möchten, die einem AWS IoT Greengrass Edge Connector für Amazon Kinesis Video Streams über AWS IoT TwinMaker Workspaces hinweg zugeordnet ist, können Sie ein Tag verwenden, das diesem Muster folgt: `WorkspaceA/WorkspaceB/WorkspaceC`. Das Grafana-Plugin erzwingt, dass die AWS IoT TwinMaker `workspaceld` verwendet wird, um die Aufnahme von AWS IoT SiteWise Komponentendaten zu gruppieren.

## Fügen Sie Ihrer Dashboard-Richtlinie weitere Berechtigungen hinzu

Das AWS IoT TwinMaker Grafana-Plugin verwendet Ihren Authentifizierungsanbieter, um `AssumeRole` für die von Ihnen erstellte Dashboard-Rolle aufzurufen. Intern schränkt das Plugin den höchsten Umfang von Berechtigungen ein, auf die Sie zugreifen können, indem es eine Sitzungsrichtlinie im `AssumeRole` Aufruf verwendet. Weitere Informationen zu Sitzungsrichtlinien finden Sie unter [Sitzungsrichtlinien](#).

Dies ist die maximal zulässige Richtlinie, die Sie für Ihre Dashboard-Rolle für einen `-AWS IoT TwinMakerWorkspace` haben können:

```
{
  "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 Sie Anweisungen hinzufügen, die Allow mehr Berechtigungen enthalten, funktionieren diese nicht auf dem AWS IoT TwinMaker Plugin. Dies ist vorgesehen, um sicherzustellen, dass das Plug-In die erforderlichen Mindestberechtigungen verwendet.

Sie können die Berechtigungen jedoch weiter einschränken. Weitere Informationen finden Sie unter [Erstellen einer AWS IoT TwinMaker -Videoplayer-Richtlinie](#).

## Erstellen der Grafana-Dashboard-IAM-Rolle

Erstellen Sie in der IAM-Konsole eine IAM-Rolle mit dem Namen *YourWorkspaceIdDashboardRole*. Hängen Sie die *YourWorkspaceIdDashboardPolicy* an die Rolle an.

Um die Vertrauensrichtlinie der Dashboard-Rolle zu bearbeiten, müssen Sie dem Grafana-Authentifizierungsanbieter die Berechtigung erteilen, für AssumeRole die Dashboard-Rolle aufzurufen. Aktualisieren Sie die Vertrauensrichtlinie mit der folgenden Vorlage:

```

{
  "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 Suchen Ihres Authentifizierungsanbieters finden Sie unter [Einrichten Ihrer Grafana-Umgebung](#).

## Erstellen einer AWS IoT TwinMaker -Videoplayer-Richtlinie

Im Folgenden finden Sie eine Richtlinienvorlage mit allen Videoberechtigungen, die Sie für das AWS IoT TwinMaker Plugin in Grafana benötigen:

```
{
  "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 Alle Videoberechtigungsrichtlinien im [Eine IAM-Richtlinie erstellen](#) Thema .

## Einschränken des Zugriffs auf Ihre -Ressourcen

Das Video-Player-Bedienfeld in Grafana ruft Kinesis Video Streams und IoT SiteWise direkt auf, um ein vollständiges Videowiedergabeerlebnis zu bieten. Um unbefugten Zugriff auf Ressourcen zu vermeiden, die nicht mit Ihrem AWS IoT TwinMaker Workspace verknüpft sind, fügen Sie der IAM-Richtlinie für Ihre Workspace-Dashboard-Rolle Bedingungen hinzu.

## Einschränken von GET-Berechtigungen

Sie können den Zugriff auf Ihre Amazon Kinesis Video Streams und AWS IoT SiteWise Ressourcen einschränken, indem Sie Ressourcen markieren. Möglicherweise haben Sie Ihre AWS IoT SiteWise Kamerakomponente bereits basierend auf der AWS IoT TwinMaker workspaceId markiert, um die Funktion für die Video-Upload-Anforderung zu aktivieren. Weitere Informationen finden Sie im Thema [Video aus dem Edge hochladen](#). Sie können dasselbe Tag-Schlüssel-Wert-Paar verwenden, um den

GET-Zugriff auf AWS IoT SiteWise Komponenten einzuschränken und Ihre Kinesis Video Streams auf die gleiche Weise zu markieren.

Sie können diese Bedingung dann zu den Anweisungen `kinesisvideo` und `iotsitewise` in der `YourWorkspaceIdDashboardPolicy` hinzufügen:

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

## Realer Anwendungsfall: Gruppieren von Kameras

In diesem Szenario haben Sie eine große Reihe von Kameras, die den Prozess von Back-Cookies in einer Fabrik überwachen. Im Batterroom werden Cookie-Batter-Batches erstellt, der Batter wird im Freezer-Raum eingefroren und die Cookies werden im Baking-Raum gebacken. In jedem dieser Räume gibt es Kameras mit unterschiedlichen Teams von Operatoren, die jeden Prozess separat überwachen. Sie möchten, dass jede Gruppe von Operatoren für ihren jeweiligen Raum autorisiert wird. Beim Erstellen eines digitalen Trichters für die Cookie-Fabrik wird ein einziger Workspace verwendet, aber die Kameraberechtigungen müssen nach Raum unterteilt werden.

Sie können diese Trennung von Berechtigungen erreichen, indem Sie Gruppen von Kameras basierend auf ihrer `groupingId` markieren. In diesem Szenario sind die `groupingIds` `BatterRoom`, `FreezerRoom`, und `BakingRoom`. Die Kamera in jedem Raum ist mit Kinesis Video Streams verbunden und sollte ein Tag haben mit: Schlüssel = `EdgeConnectorForKVS`, Wert = `BatterRoom`. Der Wert kann eine Liste von Gruppierungen sein, die durch eines der folgenden Zeichen getrennt sind: `.` `:` `+` `=` `@` `_` `/` `-`

Verwenden Sie die folgenden Richtlinienanweisungen `YourWorkspaceIdDashboardPolicy`, um die 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 Anweisungen beschränken den Zugriff auf die Streaming-Videowiedergabe und den AWS IoT SiteWise Eigenschaftsverlauf auf bestimmte Ressourcen in einer Gruppierung. Die *groupingId* wird durch Ihren Anwendungsfall definiert. In unserem Szenario wäre es die *roomId*.

## AWS IoT SiteWise BatchPutAssetPropertyValue Eingrenzungsberechtigung

Wenn Sie diese Berechtigung erteilen, wird die [Funktion Video-Upload-Anforderung im Video Player](#) aktiviert. Wenn Sie ein Video hochladen, können Sie einen Zeitraum angeben und die Anfrage von senden, indem Sie im Bereich im Grafana-Dashboard auf Senden klicken.

Um `iotsitewise:BatchPutAssetPropertyValue` permissions zu erteilen, verwenden Sie die Standardrichtlinie:

```

...,
{
  "Effect": "Allow",
  "Action": [
    "iotsitewise:BatchPutAssetPropertyValue"
  ],
  "Resource": "*",
  "Condition": {

```

```

    "StringLike": {
      "aws:ResourceTag/EdgeConnectorForKVS": "*workspaceId*"
    }
  },
  ...

```

Mithilfe dieser Richtlinie können Benutzer BatchPutAssetPropertyValue für jede Eigenschaft auf der AWS IoT SiteWise Kamerakomponente 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 Messungseigenschaft namens 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 SiteWiseKonsole. Verwenden Sie die folgende Richtlinienanweisung *YourWorkspaceId*DashboardPolicy, um die zu ändern:

```

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

```

```
},  
...
```

Diese Anweisung schränkt die Aufnahme von Daten auf eine bestimmte Eigenschaft Ihrer markierten AWS IoT SiteWise Kamerakomponente ein. Weitere Informationen finden Sie unter [Funktionsweise AWS IoT SiteWise von 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.

## AWS IoT SiteWise Voraussetzungen für die Alarmkonfiguration

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 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 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.



## Add component

### Component information

Name

Type

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

Asset Model

Choose an asset model.

Asset

Choose an asset.

Alarm Model

Choose an alarm model.

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.

```
{
  "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 an anzunehmen. Aktivieren Sie das Kontrollkästchen Schreibberechtigungen für das Alarm Configuration Panel definieren und kopieren Sie den ARN für die Rolle Write.

The screenshot shows the Grafana Settings page for a data source named "AWS IoT TwinMaker-4". The "Connection Details" section includes fields for Authentication Provider (AWS SDK Default), Assume Role ARN (arn:aws:iam:\*), External ID (External ID), Endpoint (https://{service}.{region}.amazonaws.com), and Default Region (us-east-1). A warning box titled "Assume Role ARN" provides instructions on specifying an IAM role. The "TwinMaker settings" section includes a "Workspace" field (enter workspace ID), a checked checkbox for "Define write permissions for Alarm Configuration Panel", and an "Assume Role ARN Write" field containing "arn:aws:iam:\*", which is circled in red. At the bottom, there are buttons for "Back", "Explore", "Delete", and "Save & test".

**Settings**   **Dashboards**

Name    Default

### Connection Details

Authentication Provider	<input type="text" value="AWS SDK Default"/>
Assume Role ARN	<input type="text" value="arn:aws:iam:*"/>
External ID	<input type="text" value="External ID"/>
Endpoint	<input type="text" value="https://{service}.{region}.amazonaws.com"/>
Default Region	<input type="text" value="us-east-1"/>

**Assume Role ARN**

Specify an IAM role to narrow the permission scope of this datasource. Follow the documentation [here](#) to create policies and a role with minimal permissions for your TwinMaker workspace.

### TwinMaker settings

Workspace

Define write permissions for Alarm Configuration Panel

Assume Role ARN Write

## Verwenden Sie das Grafana-Dashboard zur Alarmvisualisierung

Gehen Sie wie folgt vor, um Ihrem Dashboard ein Alarmkonfigurationsfeld hinzuzufügen und es zu konfigurieren:

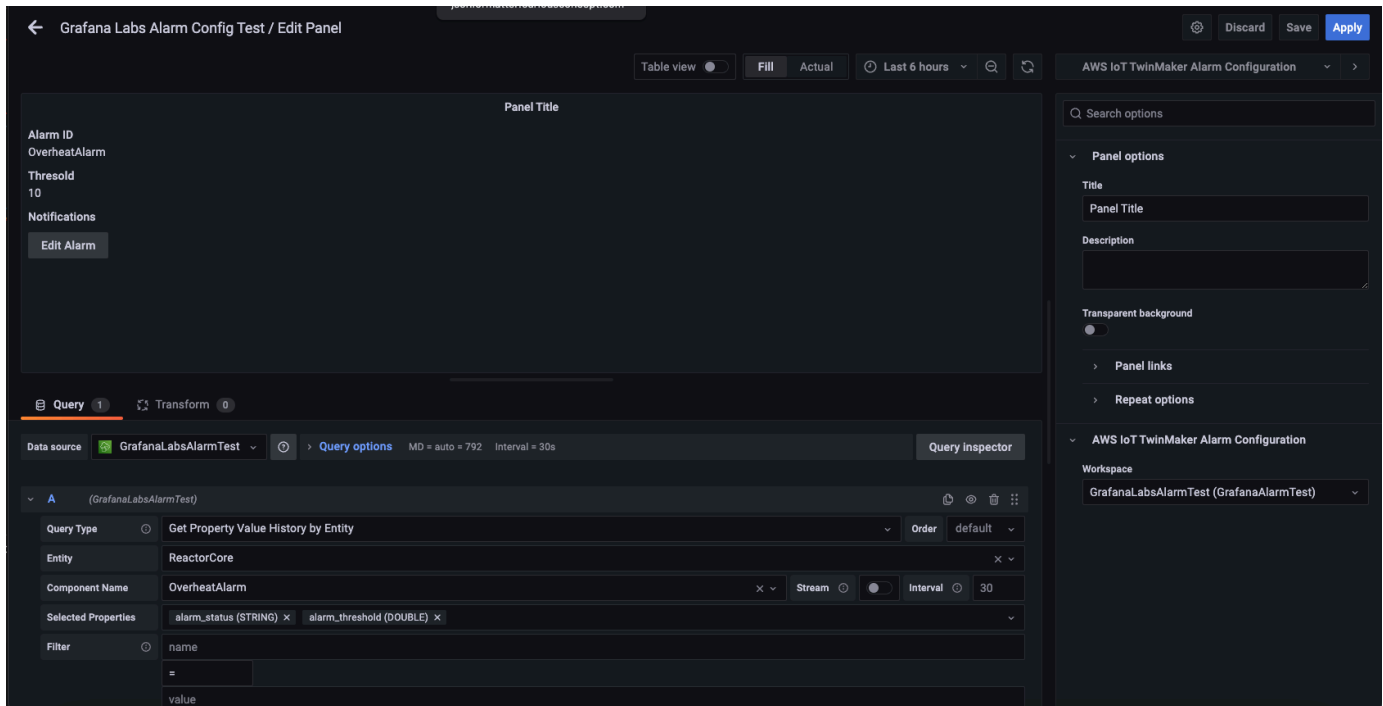
1. Wählen Sie den Arbeitsbereich in den Panel-Optionen aus.
2. Stellen Sie Ihre Datenquelle in der Abfragekonfiguration ein.
3. Verwenden Sie den folgenden Abfragetyp: `Get Property Value History by Entity`
4. Wählen Sie eine Entität oder Entitätsvariable aus, der Sie einen Alarm hinzufügen möchten.
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.

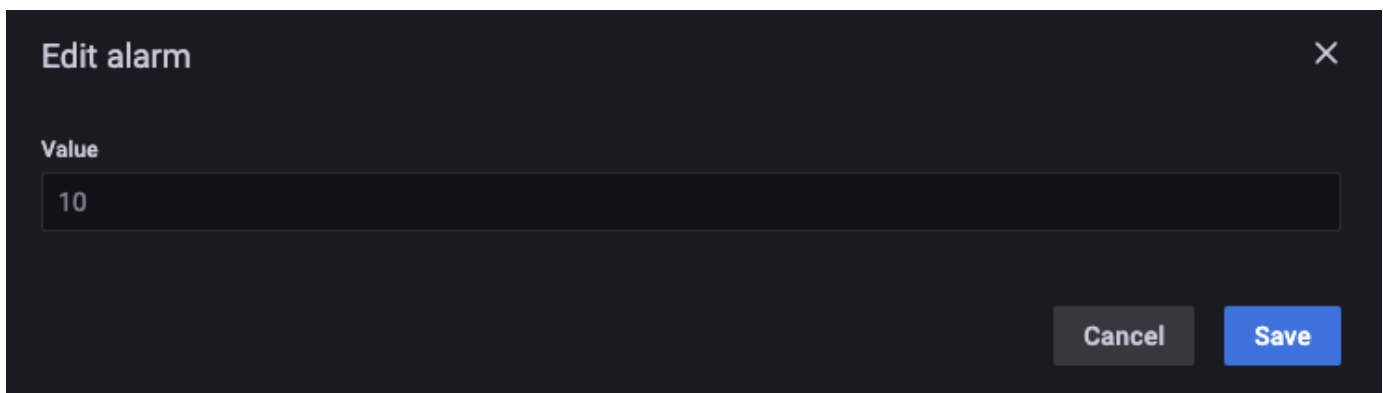
### Note

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
- Lassen Sie automatisch Ihren Matterport-Bereich und Ihre Tag-Änderungen in Ihrer AWS IoT TwinMaker Szene anzeigen 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.



The screenshot shows the AWS IoT TwinMaker console interface. The top navigation bar includes the AWS logo, 'Services', 'Resource Groups', a search icon, a notification bell, 'Customer Account', 'N. Virginia', and 'Support'. The left sidebar contains navigation options: 'AWS IoT TwinMaker', 'How it works', 'Workspaces', 'Component types', 'Entities', 'Resource library', 'Scenes', 'Query editor', 'Settings', 'What's new', 'Documentation', 'FAQ', and 'Pricing'. The main content area is titled 'Settings' and 'Settings for your AWS IoT TwinMaker account'. It has three tabs: 'Pricing options', 'Model sources', and '3rd party resources'. Below the tabs, a heading reads '3rd party software integration. You can configure AWS IoT TwinMaker to work with 3rd party software. This pages lists which software is available for integration'. A section titled 'Matterport integration (1) Info' is expanded. It contains a 'How it works' section with four steps: 1. Contact Matterport, 2. Record your Matterport SDK credentials, 3. Add your Matterport credentials into AWS secrets manager, and 4. Select your Matterport account in a scene composer scene. Each step includes a brief description and a 'Go to' button. Below this is a 'Connected accounts' section with a table showing no connections. A message states 'Copy you Matterport SDK credentials key into AWS Secret Manager.' with an 'AWS Secret Manager' button.

## 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 Kontakthanfrage 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:

Key/value	Plaintext	
application_key	matterport_application_key	Remove
client_id	matterport_oauth_app_client_id	Remove
client_secret	matterport_oauth_app_client_secret	Remove
<a href="#">+ Add row</a>		

5. 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:

AWS Secrets Manager > Secrets > Store a new secret

Step 1  
Choose secret type

Step 2  
**Configure secret**

Step 3  
Configure rotation - optional

Step 4  
Review

## Configure secret

### Secret name and description [Info](#)

**Secret name**  
A descriptive name that helps you find your secret later.

Secret name must contain only alphanumeric characters and the characters /\_+@-

**Description - optional**

Maximum 250 characters.

### Tags - optional

Key	Value - optional	
<input type="text" value="AWSIoTtwinMaker_Matterport"/>	<input type="text" value="Enter value"/>	<input type="button" value="Remove"/>
<input type="button" value="Add"/>		

### Note

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.

- 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](#).
- 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:

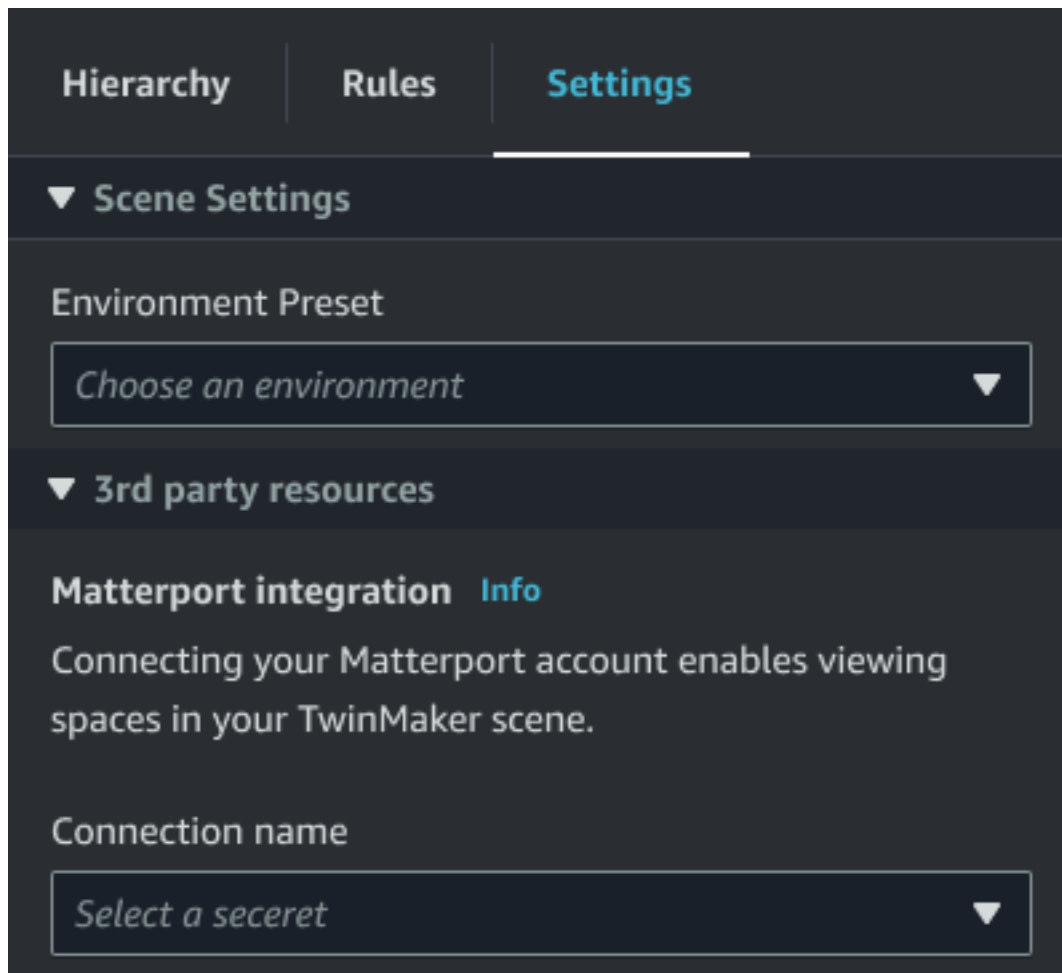
- [Geheimnisse erstellen und verwalten mit AWS Secrets Manager](#)
- [Was ist AWS Secrets Manager?](#)
- [AWS Secrets Manager Geheimnisse rotieren](#)

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:

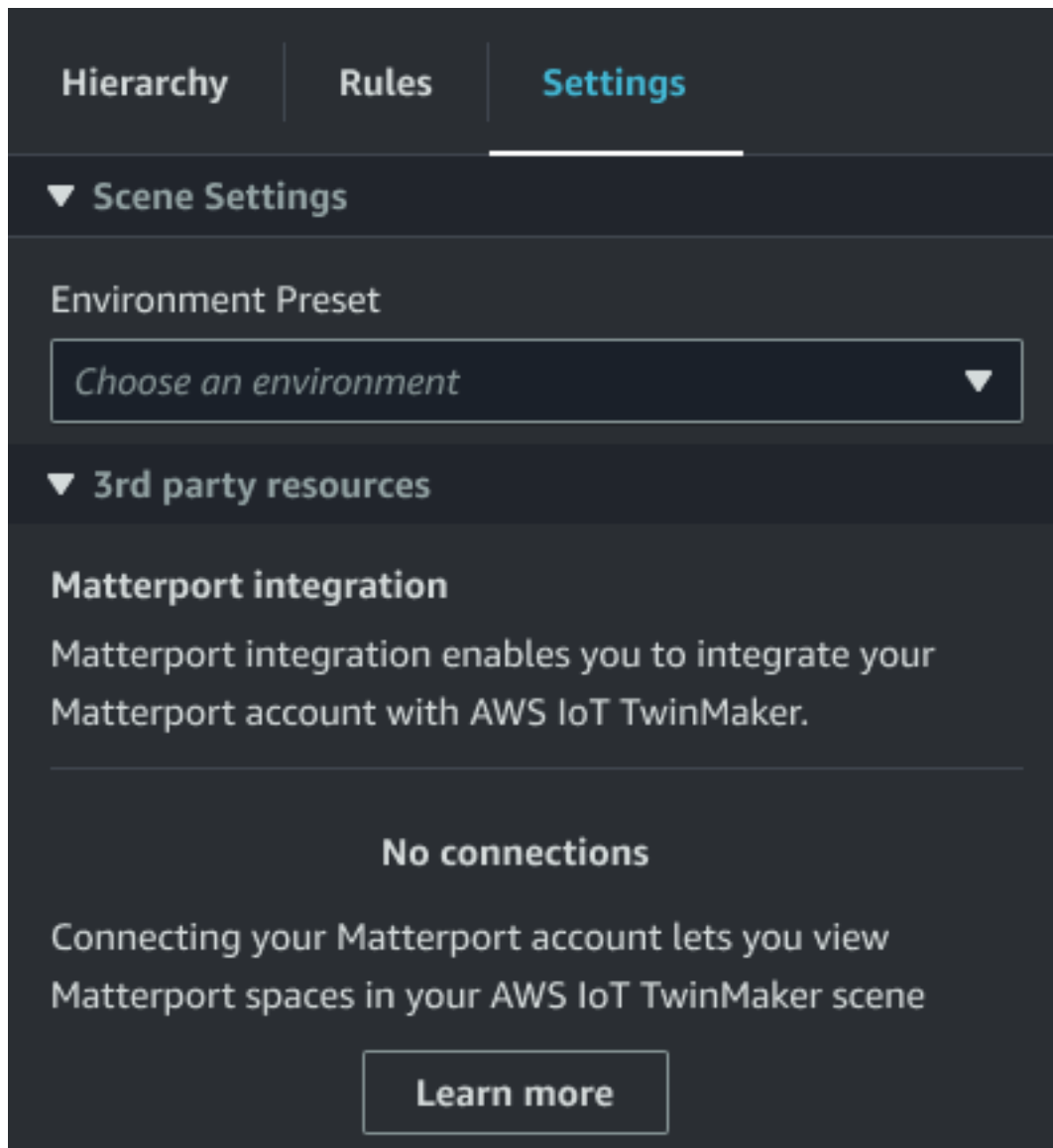
1. Melden Sie sich bei der [AWS IoT TwinMaker -Konsole](#) an.
2. Erstellen oder öffnen Sie eine bestehende AWS IoT TwinMaker Szene, in der Sie einen Matterport-Raum verwenden möchten.
3. Sobald die Szene geöffnet wurde, navigieren Sie zur Registerkarte Einstellungen.
4. Suchen Sie in den Einstellungen unter Ressourcen von Drittanbietern nach dem Verbindungsnamen und geben Sie den geheimen Schlüssel ein, den Sie in der Prozedur von erstellt haben [Speichern Sie Ihre Matterport-Anmeldeinformationen in AWS Secrets Manager](#).



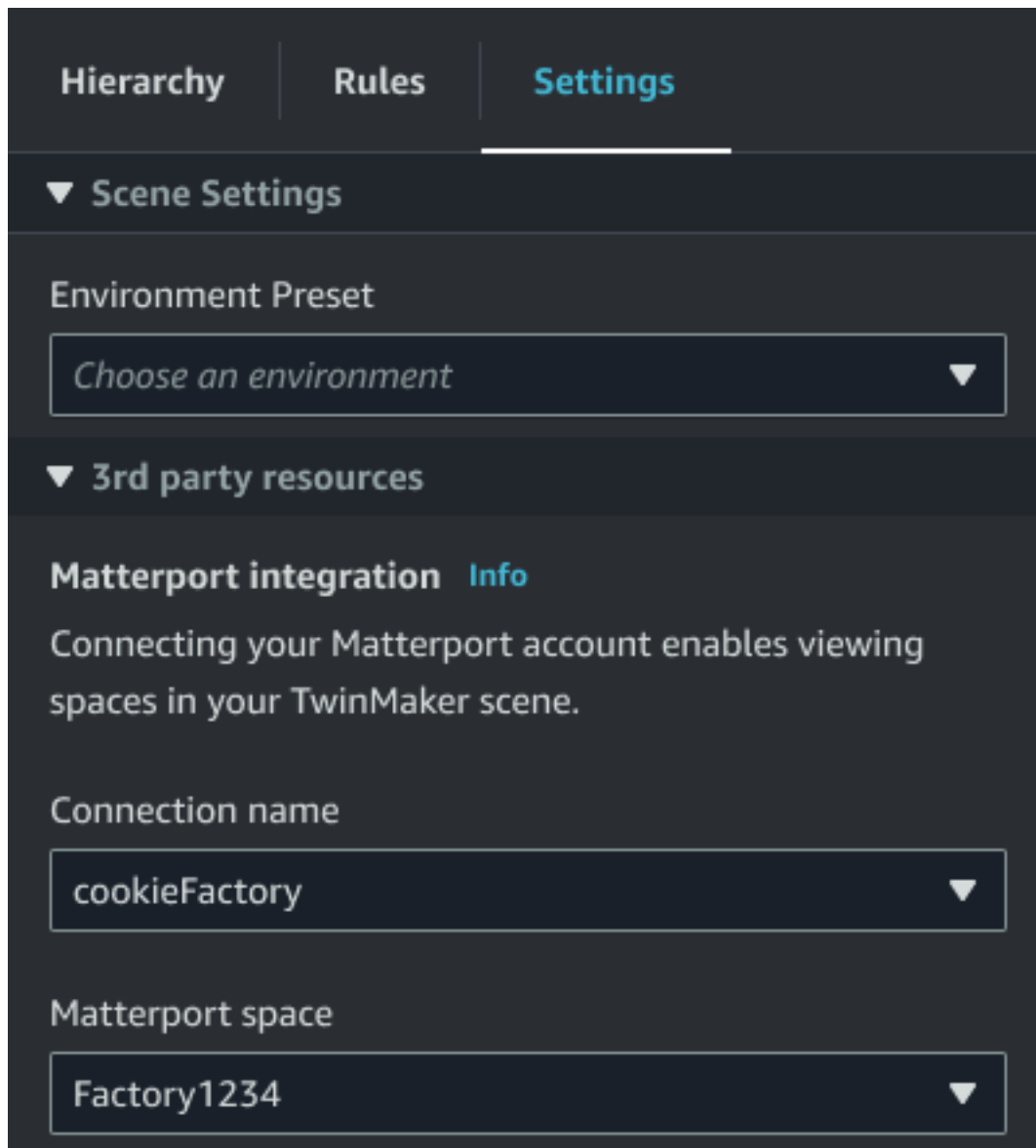
**Note**

Wenn Sie die Meldung „Keine Verbindungen“ sehen, navigieren Sie zur Seite mit den [AWS IoT TwinMaker Konsoleneinstellungen](#), um mit der Matterport-Integration zu beginnen.

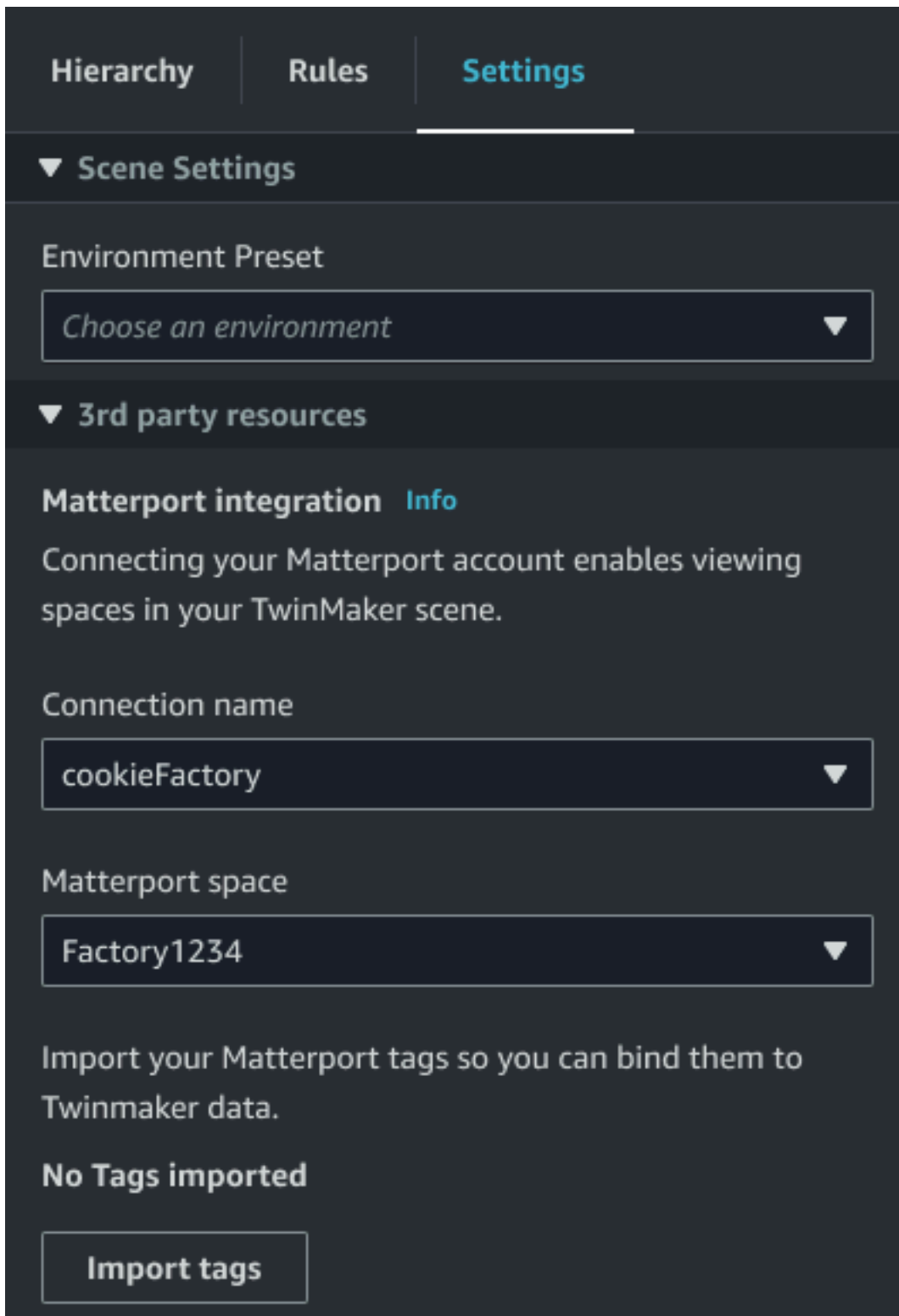




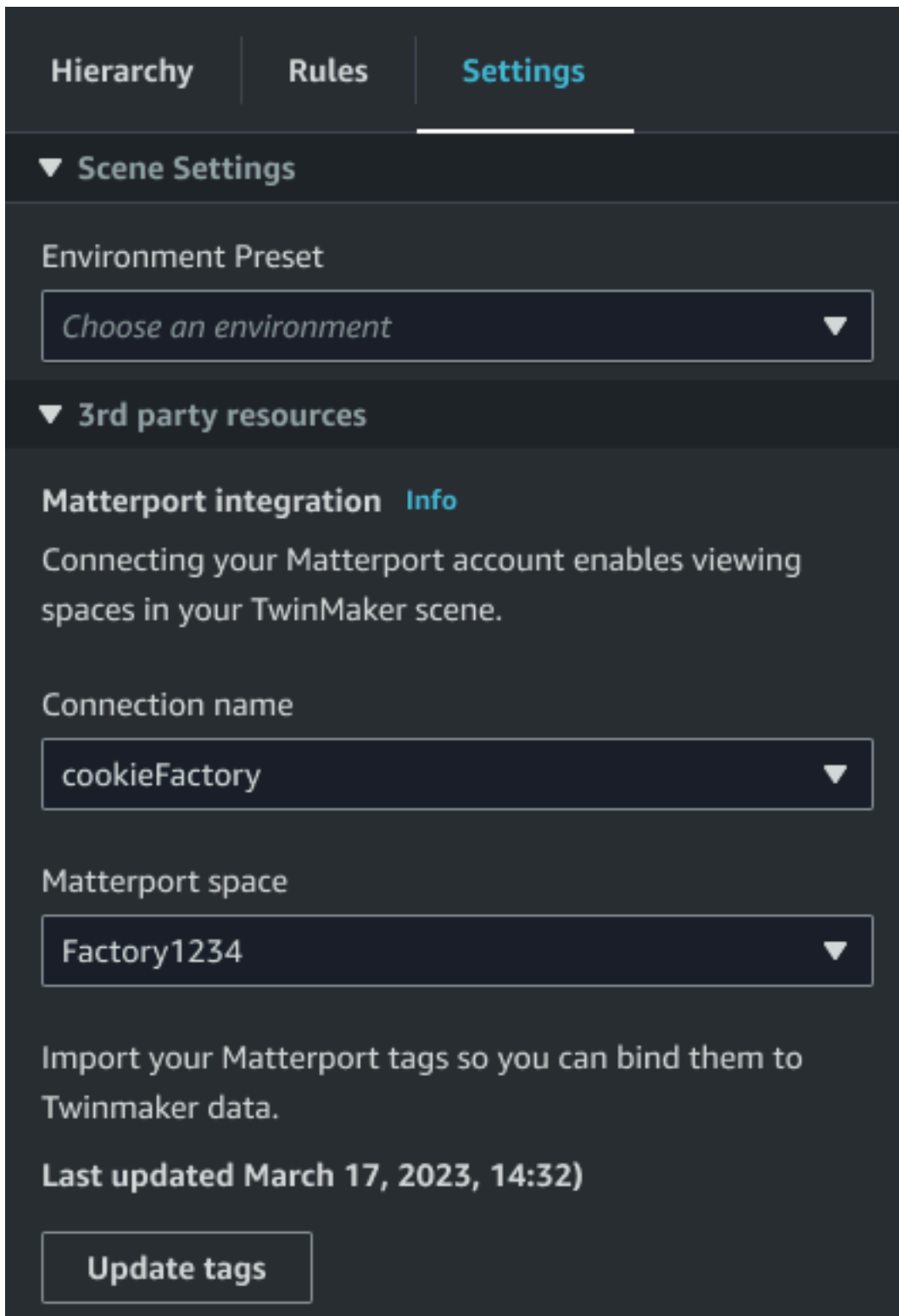
5. Wählen Sie als Nächstes den Matterport-Raum aus, den Sie in Ihrer Szene verwenden möchten, indem Sie ihn im Drop-down-Menü „Matterport-Raum“ auswählen.



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 unter [Erstellen einer benutzerdefinierten 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 Gelegenheit für die Simulation digitaler Zwang. Sie können verwenden AWS IoT TwinMaker, um den Standort und die physischen Bedingungen Ihrer Kamera zu simulieren. Erstellen Sie Entitäten in AWS IoT TwinMaker für Ihre Vor-Ort-Kameras und verwenden Sie Videokomponenten, um Live-Videos und Metadaten von Ihrer Website 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-Konnektor für den Kinesis-Videostream streamen oder Videos auf dem Edge-Gerät speichern und das Hochladen von Videos mit MQTT-Nachrichten initiieren. Verwenden Sie diese Komponente, um Videodaten von Ihren Geräten zur Verwendung mit -AWS IoT Services zu streamen. Informationen zum Generieren der erforderlichen Ressourcen und Bereitstellen des Edge-Konnektors für Kinesis Video Streams finden Sie unter [Erste Schritte mit dem Edge-Konnektor für Kinesis Video Stream](#) auf GitHub. Weitere Informationen zur AWS IoT Greengrass Komponente finden Sie in der AWS IoT Greengrass Dokumentation zum [Edge-Konnektor 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 Digital-Ensemble-Anwendung in der AWS IoT TwinMaker Konsole streamen oder aufzeichnen. Sie können Livestreams und Metadaten von Ihren Geräten auch in einem Grafana-Dashboard anzeigen. Weitere Informationen zur Integration von Grafana und AWS IoT TwinMaker finden Sie unter [AWS IoT TwinMaker Grafana-Dashboard-Integration](#).

## Verwenden des Edge-Konnektors für den Kinesis-Videostream zum Streamen von Videos in AWS IoT TwinMaker

Mit dem Edge-Konnektor für den Kinesis-Videostream können Sie Videos 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 für die 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 Komponenten für den Edge-Konnektor für den Kinesis-Videostream wurden erstellt. Weitere Informationen zum Erstellen der AWS IoT SiteWise Komponenten für den Konnektor finden Sie unter [Erste Schritte mit dem Edge-Konnektor für den Kinesis-Videostream](#).
- Der Edge-Konnektor des Kinesis-Videostroms wurde auf Ihrem AWS IoT Greengrass Gerät bereitgestellt. Weitere Informationen zur Bereitstellung der Edge-Konnektor-Komponente des Kinesis-Videostroms finden Sie unter Bereitstellung [README](#).

## Erstellen von Videokomponenten für AWS IoT TwinMaker Szenen

Führen Sie die folgenden Schritte aus, um den Edge-Konnektor für die Kinesis-Videostreamkomponente für Ihre Szene zu erstellen.

1. Öffnen Sie in der AWS IoT TwinMaker -Konsole die Szene, der Sie die Videokomponente hinzufügen möchten.
2. Nachdem sich die Szene geöffnet hat, wählen Sie eine vorhandene Entität aus oder erstellen Sie die Entität, der Sie die Komponente hinzufügen möchten, und wählen Sie dann Komponente hinzufügen aus.
3. Geben Sie im Bereich Komponente hinzufügen einen Namen für die Komponente ein und wählen Sie für Typ die Option `com.amazon.iotsitewise.connector.Edgevideo` aus.
4. Wählen Sie ein Asset-Modell aus, indem Sie den Namen des von Ihnen erstellten AWS IoT SiteWise Kameramodells auswählen. Dieser Name sollte das folgende Format haben: `EdgeConnectorForKVSCameraModel-0abc`, wobei die Zeichenfolge aus Buchstaben und Zahlen am Ende mit Ihrem eigenen Komponentennamen übereinstimmt.
5. Wählen Sie für Asset die AWS IoT SiteWise Kameraressourcen aus, von denen Sie Videos streamen möchten. Es wird ein kleines Fenster angezeigt, in dem Sie eine Vorschau des aktuellen Videostreams sehen.

### Note

Um Ihr Video-Streaming zu testen, wählen Sie Testen aus. Dieser Test sendet ein MQTT-Ereignis, um das Video-Live-Streaming zu initiieren. Warten Sie einen Moment, um zu sehen, wie das Video im Player angezeigt wird.

6. Um die Videokomponente zu Ihrer Entität hinzuzufügen, wählen Sie Komponente hinzufügen aus.



# Video und Metadaten aus dem Kinesis-Videostrom zu einem Grafana-Dashboard hinzufügen

Nachdem Sie eine Videokomponente für Ihre Entität in Ihrer AWS IoT TwinMaker Szene erstellt haben, können Sie das Videofenster in Grafana so konfigurieren, dass Livestreams angezeigt werden. Stellen Sie sicher, dass Sie ordnungsgemäß AWS IoT TwinMaker in Grafana integriert haben. Weitere Informationen finden Sie unter [AWS IoT TwinMaker Grafana-Dashboard-Integration](#).

## Important

Um Videos in Ihrem Grafana-Dashboard anzuzeigen, müssen Sie sicherstellen, dass die Grafana-Datenquellen über die richtigen IAM-Berechtigungen verfügen. Informationen zum Erstellen der erforderlichen Rolle und Richtlinie finden Sie unter [Erstellen einer Dashboard-IAM-Rolle](#).

Führen Sie die folgenden Schritte aus, um Kinesis Video Streams und Metadaten in Ihrem Grafana-Dashboard anzuzeigen.

1. Öffnen Sie das AWS IoT TwinMaker-Dashboard.
2. Wählen Sie Bereich hinzufügen und dann Einen leeren Bereich hinzufügen aus.
3. Wählen Sie in der Liste der Bereiche den Bereich AWS IoT TwinMaker Videoplayer aus.
4. Geben Sie im AWS IoT TwinMaker Videoplayer-Bereich den Streamnamen des mit dem Namen des Kinesis-Videostreams einKinesisVideoStreamName, aus dem Sie Videos streamen möchten.

## Note

Um Metadaten in den Grafana-Videobereich zu streamen, müssen Sie zunächst eine Entität mit einer Videostreaming-Komponente erstellt haben.

5. Optional: Um Metadaten von AWS IoT SiteWise Komponenten an den Videoplayer zu streamen, wählen Sie für Entität die AWS IoT TwinMaker Entität aus, die Sie in Ihrer AWS IoT TwinMaker Szene erstellt haben. Wählen Sie für den Komponentennamen die Videokomponente aus, die Sie für die Entität in Ihrer AWS IoT TwinMaker Szene erstellt haben.

# Verwenden der AWS IoT TwinMaker Flink-Bibliothek

AWS IoT TwinMaker stellt eine Flink-Bibliothek bereit, mit der Sie Daten in externe Datenspeicher lesen und schreiben können, die in Ihren digitalen Komprimierungen verwendet werden.

Sie verwenden die AWS IoT TwinMaker Flink-Bibliothek, indem Sie sie als benutzerdefinierten Connector in Managed Service für Apache Flink installieren und Flink-SQL-Abfragen in einem Zeppelin-Notebook in Managed Service für Apache Flink ausführen. Das Notebook kann zu einer kontinuierlich ausgeführten Stream-Verarbeitungsanwendung hochgestuft 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 gefüllter Workspace mit Szenen und Komponenten. Verwenden Sie die integrierten Komponententypen für Daten von -AWSServices (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 werden von [Apache Zeppelin](#) unterstützt 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 [AWS IoT TwinMakerBenutzerhandbuch für die Flink-Bibliothek](#) .

Anweisungen zum Einrichten AWS IoT TwinMaker mit dem Schnellstart in den [AWS IoT TwinMaker Beispielen](#) finden Sie unter [README-Datei für die Beispiel-Insights-Anwendung](#) .

# Protokollieren und Überwachen in AWS IoT TwinMaker

Die Überwachung ist ein wichtiger Faktor für die Aufrechterhaltung der Zuverlässigkeit, Verfügbarkeit und Leistung von AWS IoT TwinMaker und anderer AWS-Lösungen. AWS IoT TwinMaker unterstützt die folgenden Überwachungs-Tools, mit denen Sie den Service beobachten, Fehler melden und automatische Aktionen ausführen können, wenn notwendig:

- Amazon CloudWatch überwacht in Echtzeit Ihre AWS Ressourcen und die in ausgeführten Anwendungen AWS. Sie können Metriken erfassen und verfolgen, benutzerdefinierte Dashboards erstellen und Benachrichtigungen festlegen, die Sie benachrichtigen oder Maßnahmen ergreifen, wenn eine bestimmte Metrik einen von Ihnen festgelegten Schwellenwert erreicht. Beispielsweise können Sie mit der CPU-Auslastung oder andere Metriken für Ihre Amazon EC2 EC2-Instances CloudWatch erfassen und bei Bedarf automatisch neue Instances starten. Weitere Informationen finden Sie im [CloudWatch Amazon-Benutzerhandbuch](#).
- Amazon CloudWatch Logs überwacht, speichert und ermöglicht den Zugriff auf Ihre Protokolldateien von AWS IoT TwinMaker Gateways und anderen Quellen aus. CloudTrail CloudWatch Mit Protokollen können Sie Informationen in den Protokolldateien überwachen und Sie benachrichtigen, wenn bestimmte Schwellenwerte erreicht. Sie können Ihre Protokolldaten auch in einem sehr robusten Speicher archivieren. Weitere Informationen finden Sie im [Amazon CloudWatch Logs-Benutzerhandbuch](#).
- AWS CloudTrail erfasst API-Aufrufe und zugehörige Ereignisse, die von oder im Namen Ihres AWS-Kontos erfolgten, und übermittelt die Protokolldateien an einen von Ihnen angegebenen Amazon-S3-Bucket. Sie können die Benutzer und Konten, die AWS aufgerufen haben, identifizieren, sowie die Quell-IP-Adresse, von der diese Aufrufe stammen, und den Zeitpunkt der Aufrufe ermitteln. Weitere Informationen finden Sie im [AWS CloudTrail-Benutzerhandbuch](#).

## Themen

- [Überwachung AWS IoT TwinMaker mit CloudWatch Amazon-Metriken](#)
- [Protokollierung von AWS IoT TwinMaker-API-Aufrufen mit AWS CloudTrail](#)

# Überwachung AWS IoT TwinMaker mit CloudWatch Amazon-Metriken

Sie können mit AWS IoT TwinMaker CloudWatch, wobei Rohdaten erfasst und in lesbare Metriken verarbeitet, die nahezu in Echtzeit bereitgestellt werden. Diese Statistiken werden 15 Monate gespeichert, damit Sie auf Verlaufsdaten 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 aufgeführten Metriken und Dimensionen im `AWS/IoTTwinMaker` Namespace.

## Tip

AWS IoT TwinMaker veröffentlicht Metriken in Intervallen von einer Minute. Wenn Sie diese Metriken in der CloudWatch Konsole in Diagrammen anzeigen, empfehlen wir, 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
<code>ComponentTypeCreationFailure</code>	<p>Diese Metrik gibt an, ob die Erstellung des Komponententyps erfolgreich war.</p> <p>Die Metrik wird veröffentlicht, wenn sich ein Komponententyp im <code>CREATING</code> Status befindet.</p>

Metrik	Beschreibung
	<p>Dies passiert, wenn ein Komponententyp mit den erforderlichen Eigenschaften im Schemainitialisierer erstellt wird und diese Eigenschaften mit Standardwerten instanziiert werden.</p> <p>Der Metrikwert kann entweder 0 für Erfolg oder 1 für Misserfolg stehen.</p> <p>Abmessungen: ComponentTypeId, Workspace Id.</p> <p>Einheiten: Anzahl</p>
ComponentTypeUpdateFailure	<p>Diese Metrik gibt an, ob die Aktualisierung des Komponententyps erfolgreich war.</p> <p>Die Metrik wird veröffentlicht, wenn sich ein Komponententyp im UPDATING Status befindet. Dies passiert, wenn ein Komponententyp mit den erforderlichen Eigenschaften im Schemainitialisierer aktualisiert wird und diese Eigenschaften mit Standardwerten instanziiert werden.</p> <p>Der Metrikwert kann entweder 0 für Erfolg oder 1 für Misserfolg stehen.</p> <p>Abmessungen: ComponentTypeId, Workspace Id.</p> <p>Einheiten: Anzahl</p>

Metrik	Beschreibung
EntityCreationFailure	<p>Diese Metrik gibt an, ob die Erstellung der Entität erfolgreich war. Die Metrik wird veröffentlicht, wenn sich eine Entität imCREATING Bundesstaat befindet. Dies passiert, wenn eine Entität mit einer Komponente erstellt wird.</p> <p>Der Metrikwert kann entweder0 für Erfolg oder1 für Misserfolg stehen.</p> <p>Abmessungen: EntityName, EntityId, Workspaceld.</p> <p>Einheiten: Anzahl</p>
EntityUpdateFailure	<p>Diese Metrik gibt an, ob die Entitätsaktualisierung erfolgreich war. Die Metrik wird veröffentlicht, wenn sich eine Entität imUPDATING Bundesstaat befindet. Dies passiert, wenn eine Entität aktualisiert wird.</p> <p>Der Metrikwert kann entweder0 für Erfolg oder1 für Misserfolg stehen.</p> <p>Abmessungen: EntityName, EntityId, Workspaceld.</p> <p>Einheiten: Anzahl</p>

Metrik	Beschreibung
EntityDeletionFailure	<p>Diese Metrik gibt an, ob das Löschen der Entität erfolgreich war. Die Metrik wird veröffentlicht, wenn sich eine Entität imDELETING Bundesstaat befindet. Dies passiert, wenn eine Entität gelöscht wird.</p> <p>Der Metrikwert kann entweder0 für Erfolg oder1 für Misserfolg stehen.</p> <p>Abmessungen: EntityName, EntityId, Workspaceld.</p> <p>Einheiten: Anzahl</p>

 Tip

Alle Metriken werden imAWS/IoTTwinMaker -Namespace veröffentlicht.

## Protokollierung von AWS IoT TwinMaker-API-Aufrufen mit AWS CloudTrail

AWS IoT TwinMakerist in integriertAWS CloudTrail, einem Service, der eine Aufzeichnung der von einem Benutzer, einer Rolle oder einemAWS -Service durchgeführten Aktionen bereitstelltAWS IoT TwinMaker. CloudTrail erfasst alle API-Aufrufe fürAWS IoT TwinMaker als Ereignisse. Zu den erfassten Aufrufen gehören Aufrufe von der AWS IoT TwinMaker-Konsole und Code-Aufrufe der AWS IoT TwinMaker-API-Operationen. Wenn Sie einen Trail erstellen, können Sie die kontinuierliche Bereitstellung von CloudTrail Ereignissen für einen Amazon S3 S3-Bucket aktivieren, einschließlich Ereignisse fürAWS IoT TwinMaker. Auch wenn Sie keinen Trail konfigurieren, können Sie die neuesten Ereignisse in der CloudTrail Konsole in Event history (Ereignisverlauf) anzeigen. Mit den von CloudTrail gesammelten Informationen können Sie die an gestellte AnfrageAWS IoT TwinMaker, die IP-Adresse, von der die Anforderung gestellt wurde, den Initiator der Anfrage, den Zeitpunkt der Anfrage und weitere Angaben bestimmen.

Weitere Informationen CloudTrail dazu finden Sie im [AWS CloudTrailBenutzerhandbuch](#).

## AWS IoT TwinMakerInformationen in CloudTrail

Wenn Sie IhrAWS Konto erstellen, CloudTrail wird es automatisch aktiviert. CloudTrail Aufzeichnungen unterstützen Ereignisaktivitäten, die zusammen mit anderenAWS Serviceereignissen in der Ereignishistorie auftreten.AWS IoT TwinMaker Sie können die neusten Ereignisse in Ihr AWS-Konto herunterladen und dort suchen und anzeigen. Weitere Informationen finden Sie unter [Anzeigen von Ereignissen mit dem CloudTrail -Ereignisverlauf](#).

Zur kontinuierlichen Aufzeichnung von Ereignissen in Ihrem AWS-Konto, einschließlich Ereignissen für AWS IoT TwinMaker, erstellen Sie einen Trail. Ein Trail ermöglicht es CloudTrail , Protokolldateien in einem Amazon S3 S3-Bucket bereitzustellen. Wenn Sie einen Pfad in der Konsole anlegen, gilt dieser standardmäßig für alleAWS Regionen. CloudTrailprotokolliert Ereignisse aus allen Regionen in derAWS -Partition und stellt die Protokolldateien in dem von Ihnen angegebenen Amazon S3 S3-Bucket bereit. Darüber hinaus können Sie andereAWS -Services konfigurieren, um die in den CloudTrail Protokollen erfassten Ereignisdaten weiter zu analysieren und entsprechend zu agieren. Weitere Informationen finden Sie unter:

- [Übersicht zum Erstellen eines Trails](#)
- [CloudTrail unterstützte Dienste und Integrationen](#)
- [Amazon SNS von CloudTrail](#)
- [Empfangen CloudTrail von CloudTrail Protokolldateien aus mehreren Konten](#)

Die meistenAWS IoT TwinMaker Operationen werden von der [AWS IoT TwinMakerAPI-Referenz](#) protokolliert CloudTrail und sind in dieser dokumentiert.

Die folgenden Vorgänge werden nicht von protokolliert 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 verbundenen Benutzer gesendet wurde.



- Gibt an, ob die Anforderung aus einem anderen AWS-Service gesendet 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).
- Verwenden Sie SSL/TLS, um mit Ressourcen zu kommunizieren. AWS Wir benötigen TLS 1.2 und empfehlen TLS 1.3.
- Richten Sie die API und die Protokollierung von Benutzeraktivitäten mit ein. AWS CloudTrail
- 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 140-2-validierte kryptografische Module benötigen, verwenden Sie einen FIPS-Endpunkt. Weitere Informationen über verfügbare FIPS-Endpunkte finden Sie unter [Federal Information Processing Standard \(FIPS\) 140-2](#).

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 API AWS IoT TwinMaker oder den SDKs arbeiten oder diese anderweitig AWS-Services verwenden. AWS CLI AWS 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. Sie müssen dann Anträge an Ihren IAM-Administrator stellen, um die Berechtigungen Ihrer Servicenutzer 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 unter [Wie 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, mit denen Sie Ihre Anfragen mithilfe Ihrer Anmeldeinformationen kryptografisch signieren können. Wenn Sie keine AWS Tools verwenden, müssen Sie Anfragen selbst signieren. Weitere Informationen zur Verwendung der empfohlenen Methode, um Anfragen selbst zu [signieren, finden Sie im IAM-Benutzerhandbuch unter AWS API-Anfragen](#) signieren.

Unabhängig von der verwendeten Authentifizierungsmethode müssen Sie möglicherweise zusätzliche Sicherheitsinformationen angeben. 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 [Verwenden der Multi-Faktor-Authentifizierung \(MFA\) in AWS](#) im IAM-Benutzerhandbuch.

### AWS-Konto Root-Benutzer

Wenn Sie ein neues AWS-Konto erstellen, 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

und verwenden Sie diese, 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 einer Gruppe mit dem Namen IAMAdmins Berechtigungen zum Verwalten 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 [Erstellen eines IAM-Benutzers \(anstatt einer Rolle\)](#) im IAM-Benutzerhandbuch.

## IAM-Rollen

Eine [IAM-Rolle](#) ist eine Identität innerhalb Ihres Unternehmens AWS-Konto, die über bestimmte Berechtigungen verfügt. Sie ist einem IAM-Benutzer vergleichbar, ist aber nicht mit einer bestimmten Person verknüpft. Sie können vorübergehend eine IAM-Rolle in der übernehmen, AWS Management Console indem Sie die Rollen [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 [Verwenden von IAM-Rollen](#) 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](#) 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 [So unterscheiden sich IAM-Rollen von ressourcenbasierten Richtlinien](#) im IAM-Benutzerhandbuch.



- **Serviceübergreifender Zugriff** — Einige AWS-Services verwenden Funktionen in anderen AWS-Services. Wenn Sie beispielsweise einen Aufruf in einem Service tätigen, führt dieser Service häufig Anwendungen in Amazon-EC2 aus oder speichert Objekte in Amazon-S3. 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, kombiniert 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.
- **Anwendungen, die auf Amazon EC2 ausgeführt werden** — Sie können eine IAM-Rolle verwenden, um temporäre Anmeldeinformationen für Anwendungen zu verwalten, die auf einer EC2-Instance ausgeführt werden und API-Anfragen stellen AWS CLI . AWS Das ist eher zu empfehlen, als Zugriffsschlüssel innerhalb der EC2-Instance zu speichern. Um einer EC2-Instance eine AWS Rolle zuzuweisen und sie allen ihren Anwendungen zur Verfügung zu stellen, erstellen Sie ein Instance-Profil, das an die Instance angehängt ist. Ein Instance-Profil enthält die Rolle und ermöglicht, dass Programme, die in der EC2-Instance ausgeführt werden, temporäre Anmeldeinformationen erhalten. Weitere Informationen finden Sie unter [Verwenden einer IAM-Rolle zum Erteilen von Berechtigungen für Anwendungen, die auf Amazon-EC2-Instances ausgeführt werden](#) im IAM-Benutzerhandbuch.

Informationen dazu, wann Sie IAM-Rollen oder IAM-Benutzer verwenden sollten, finden Sie unter [Erstellen einer IAM-Rolle \(anstatt eines Benutzers\)](#) im IAM-Benutzerhandbuch.

## 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. Berechtigungen in den Richtlinien bestimmen, ob die 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 bedeutet, welcher Prinzipal kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

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 [Erstellen von IAM-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 eingebundenen Richtlinie wählen, finden Sie unter [Auswahl 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.

## Zugriffssteuerungslisten (ACLs)

Zugriffssteuerungslisten (ACLs) steuern, welche Prinzipale (Kontomitglieder, Benutzer oder Rollen) auf eine Ressource zugreifen können. ACLs sind ähnlich wie ressourcenbasierte Richtlinien, verwenden jedoch nicht das JSON-Richtliniendokumentformat.

Amazon S3 und Amazon VPC sind Beispiele für Services, die ACLs unterstützen. AWS WAF Weitere Informationen“ zu ACLs finden Sie unter [Zugriffskontrollliste \(ACL\) – Übersicht](#) (Access Control List) im Amazon-Simple-Storage-Service-Entwicklerhandbuch.

## 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.

- **Service Control Policies (SCPs)** — SCPs sind JSON-Richtlinien, die die maximalen Berechtigungen für eine Organisation oder Organisationseinheit (OU) festlegen. AWS Organizations ist ein Dienst zur Gruppierung und zentralen Verwaltung mehrerer AWS-Konten, die Ihrem Unternehmen gehören. Wenn Sie innerhalb einer Organisation alle Features aktivieren, können Sie Service-Kontrollrichtlinien (SCPs) auf alle oder einzelne Ihrer Konten anwenden. Das SCP schränkt die Berechtigungen für Entitäten in Mitgliedskonten ein, einschließlich der einzelnen Entitäten. Root-Benutzer des AWS-Kontos Weitere Informationen zu Organizations und SCPs finden Sie unter [Funktionsweise von SCPs](#) 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
<a href="#">Identitätsbasierte Richtlinien</a>	Ja
<a href="#">Ressourcenbasierte Richtlinien</a>	Nein
<a href="#">Richtlinienaktionen</a>	Ja
<a href="#">Richtlinienressourcen</a>	Ja
<a href="#">Bedingungsschlüssel für die Richtlinie</a>	Ja
<a href="#">ACLs</a>	Nein
<a href="#">ABAC (Tags in Richtlinien)</a>	Teilweise
<a href="#">Temporäre Anmeldeinformationen</a>	Ja
<a href="#">Hauptberechtigungen</a>	Ja
<a href="#">Servicerollen</a>	Ja
<a href="#">Service-verknüpfte Rollen</a>	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 [Erstellen von IAM-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 [Wie sich IAM-Rollen von ressourcenbasierten Richtlinien unterscheiden](#) 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 kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

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 bedeutet die Festlegung, welcher Prinzipal Aktionen für welche Ressourcen 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 finden Sie unter [Ressourcen definiert von 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 Richtlini enbedingungsschlüssel	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 Bedingungs Schlüssel angeben, 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 Bedingungs Schlüssel und dienstspezifische Bedingungs Schlüssel. Eine Übersicht aller AWS globalen Bedingungs Schlüssel finden Sie unter [Kontextschlüssel für AWS globale Bedingungen](#) im IAM-Benutzerhandbuch.

Eine Liste der AWS IoT TwinMaker Bedingungs Schlüssel finden Sie unter [Bedingungs Schlüssel für AWS IoT TwinMaker](#) in der Service Authorization Reference. Informationen zu den Aktionen und Ressourcen, mit denen Sie einen Bedingungs Schlüssel 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](#)

## Zugriffssteuerungslisten (ACLs) in AWS IoT TwinMaker

Unterstützt ACLs

Nein

Zugriffssteuerungslisten (ACLs) steuern, welche Prinzipale (Kontomitglieder, Benutzer oder Rollen) auf eine Ressource zugreifen können. ACLs sind ähnlich wie ressourcenbasierte 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 Bedingungelement 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 [Was ist ABAC?](#) 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 zu einer 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	Nein
--------------------------------------	------

Eine dienstbezogene Rolle ist eine Art von Servicerolle, die mit einer Service-Verknüpfung 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](#) im IAM-Benutzerhandbuch.

Einzelheiten zu Aktionen und Ressourcentypen, die von definiert wurden AWS IoT TwinMaker, einschließlich des Formats der 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 verursauchen 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 [Richtlinienvvalidierung zum 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 [Konfigurieren eines MFA-geschützten API-Zugriffs](#) 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:ListGroupsWithUser",
        "iam:ListAttachedUserPolicies",
        "iam:ListUserPolicies",
        "iam:GetUser"
      ],
      "Resource": ["arn:aws:iam::*:user/${aws:username}"]
    },
    {
      "Sid": "NavigateInConsole",
      "Effect": "Allow",
      "Action": [
        "iam:GetGroupPolicy",
        "iam:GetPolicyVersion",
        "iam:GetPolicy",
        "iam:ListAttachedGroupPolicies",
        "iam:ListGroupPolicies",
        "iam:ListPolicyVersions",
        "iam:ListPolicies",
        "iam:ListUsers"
      ],
      "Resource": "*"
    }
  ]
}
```

## 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 TwinMaker übergeben 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. Im Fall von Diensten, die ressourcenbasierte Richtlinien oder Zugriffskontrolllisten (Access Control Lists, ACLs) verwenden, 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 [So unterscheiden sich IAM-Rollen von ressourcenbasierten Richtlinien](#) 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 Services, die serviceorientierte Rollen unterstützen, finden Sie unter [AWS services that work with IAM](#) (-Services, die mit IAM funktionieren). Suchen Sie nach den Services, für die Yes (Ja) in der Spalte Service-linked roles (Serviceorientierte Rollen) angegeben ist. Wählen Sie über einen Link Ja aus, um die Dokumentation zu einer serviceverknüpften Rolle für diesen Service anzuzeigen.

### Mit dem Dienst verknüpfte Rollenberechtigungen für AWS IoT TwinMaker

AWS IoT TwinMaker verwendet die dienstbezogene Rolle mit dem Namen `AWSServiceRoleForIoTtwinMaker`— Ermöglicht AWS IoT TwinMaker das Aufrufen anderer AWS Dienste und das Synchronisieren ihrer Ressourcen in Ihrem Namen.

Die `AWSServiceRoleForIoTtwinMaker` dienstgebundene Rolle vertraut darauf, dass die folgenden Dienste die Rolle übernehmen:

- `iottwinmaker.amazonaws.com`

Die genannte Rollenberechtigungsrichtlinie `AWSIoTtwinMakerServiceRolePolicy` 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 erlaubt es Ihnen nicht, die `AWSServiceRoleForIoT` dienstbezogene 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 serviceverknüpften Rolle für AWS IoT TwinMaker

Wenn Sie ein Feature oder einen Dienst, die bzw. der eine serviceverknüpfte 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 AWSServiceRoleForIoT TwinMaker dienstverknüpfte 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 AWS verwaltete ReadOnlyAccess-Richtlinie bietet beispielsweise Lesezugriff auf alle AWS Dienste und Ressourcen. Wenn ein Dienst ein neues Feature 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: AWSIoTtwinMakerServiceRolePolicy

Sie können keine Verbindungen AWSIoTtwinMakerServiceRolePolicy 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 [Mit dem Dienst verknüpfte Rollenberechtigungen für AWS IoT TwinMaker](#).

Die genannte Richtlinie für Rollenberechtigungen AWSIoTtwinMakerServiceRolePolicy 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

```

{
  "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",

```



Änderung	Beschreibung	Datum
	<p>n ausgeführt werden können AWS IoT TwinMaker :</p> <ul style="list-style-type: none"> <li>• Aktion: <code>iotsitewise:DescribeAsset</code>, <code>iotsitewise:ListAssets</code>, <code>iotsitewise:DescribeAssetModel</code>, and <code>iotsitewise:ListAssetModels</code>, <code>iottwinmaker:GetEntity</code>, <code>iottwinmaker&gt;CreateEntity</code>, <code>iottwinmaker:UpdateEntity</code>, <code>iottwinmaker&gt;DeleteEntity</code>, <code>iottwinmaker&gt;ListEntities</code>, <code>iottwinmaker:GetComponentType</code>, <code>iottwinmaker&gt;CreateComponentType</code>, <code>iottwinmaker:UpdateComponentType</code>, <code>iottwinmaker&gt;DeleteComponentType</code>, <code>iottwinmaker&gt;ListComponentTypes</code> für all your <code>iotsitewise</code> asset and asset-model resources</li> </ul> <p>Weitere Informationen finden Sie unter <a href="#">Mit dem Dienst</a></p>	



Änderung	Beschreibung	Datum
	<a href="#">verknüpfte Rollenberechtigungen für AWS IoT TwinMaker.</a>	
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 privat auf AWS IoT TwinMaker APIs zugreifen können, ohne dass ein Internet-Gateway, ein NAT-Gerät (Network Address Translation), eine VPN-Verbindung oder eine AWS Direct Connect-Verbindung erforderlich ist. Instances in Ihrer VPC benötigen keine öffentlichen IP-Adressen, um mit AWS IoT TwinMaker APIs zu kommunizieren. Datenverkehr zwischen Ihrer VPC und dem, der das Amazon-Netzwerk AWS IoT TwinMaker nicht verlässt.

Jeder Schnittstellenendpunkt wird durch eine oder mehrere [Elastic-Netzwerk-Schnittstellen](#) in Ihren Subnetzen dargestellt.

Weitere Informationen finden Sie unter [Interface VPC Endpoints \(AWS PrivateLink\)](#) 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:

- [GetPropertyWert](#)
  - [GetPropertyValueHistory](#)
  - [BatchPutPropertyValues](#)
- Verwenden Sie für die API-Operationen auf der Kontrollebene den folgenden Endpunkt:

```
api.iottwinmaker.region.amazonaws.com
```

Zu den unterstützten API-Vorgängen auf der Kontrollebene gehören:

- [CreateComponentTyp](#)
- [CreateEntity](#)
- [CreateScene](#)
- [CreateWorkspace](#)
- [DeleteComponentTyp](#)
- [DeleteEntity](#)
- [DeleteScene](#)
- [DeleteWorkspace](#)
- [GetComponentTyp](#)
- [GetEntity](#)
- [GetScene](#)
- [GetWorkspace](#)
- [ListComponentTypen](#)
- [ListComponentTypen](#)
- [ListEntities](#)
- [ListScenes](#)
- [ListTagsForResource](#)
- [ListWorkspaces](#)
- [TagResource](#)
- [UntagResource](#)
- [UpdateComponentTyp](#)
- [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, unduse1-az6.

Der DataPlane Dienst wird in den folgenden Verfügbarkeitszonen unterstützt: use1-az1use1-az2, unduse1-az4.

- us-west-2

Die DataPlane Dienste ControlPlane und werden in den folgenden Verfügbarkeitszonen unterstützt: usw2-az1 usw2-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 für Ihre AWS Ressourcen — 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. Ersetzen Sie *Region* durch Ihre AWS Region.

```
data.iottwinmaker.region.amazonaws.com
```

- Verwenden Sie für die API-Operationen auf der Kontrollebene den folgenden Endpunkt. Ersetzen Sie *Region* 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-Endpoint-ID und Region.

```
vpc-endpoint-id.api.iottwinmaker.region.vpce.amazonaws.com
```

- Deaktivieren Sie die Host-Präfix-Injektion. Die AWS SDKs AWS CLI und stellen dem Service-Endpoint verschiedene Host-Präfixe voran, wenn Sie die einzelnen API-Operationen aufrufen. Dies führt dazu, dass die AWS SDKs AWS CLI und ungültige URLs erzeugen, AWS IoT TwinMaker wenn Sie einen VPC-Endpoint angeben.

#### Important

Sie können die Hostpräfixinjektion in AWS CLI oder nicht deaktivieren. AWS Tools for PowerShell Das bedeutet, dass Sie, wenn Sie `privates DNS` deaktiviert haben, den VPC-Endpoint nicht verwenden AWS CLI oder AWS IoT TwinMaker über ihn AWS Tools for PowerShell darauf zugreifen können. Wenn Sie diese Tools für den Zugriff AWS IoT TwinMaker über den Endpoint verwenden möchten, aktivieren Sie `privates DNS`.

Weitere Informationen zur Deaktivierung der Hostpräfixinjektion in den AWS SDKs finden Sie in den folgenden Dokumentationsabschnitten für jedes SDK:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java](#)
- [AWS SDK for Java 2.x](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for .NET](#)
- [AWS SDK for PHP](#)
- [AWS SDK for Python \(Boto3\)](#)
- [AWS SDK for Ruby](#)

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 heruntergeladen AWS Artifact. Weitere Informationen finden Sie unter [Berichte heruntergeladen 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:

- [Schnellstartanleitungen zu Sicherheit und Compliance](#) — In diesen Bereitstellungsleitfäden werden architektonische Überlegungen erörtert und Schritte für die Implementierung von Basisumgebungen beschrieben AWS , bei denen Sicherheit und Compliance im Mittelpunkt stehen.
- [Architecting for HIPAA Security and Compliance on Amazon Web Services](#) — In diesem Whitepaper wird beschrieben, wie Unternehmen HIPAA-fähige Anwendungen erstellen AWS können.

## Note

AWS-Services Nicht alle sind HIPAA-fähig. Weitere Informationen finden Sie in der [Referenz für HIPAA-berechtigte Services](#).

- [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](#)— Dies AWS-Service bietet 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. Eine Liste der unterstützten Services und Kontrollen finden Sie in der [Security-Hub-Steuerungsreferenz](#).
- [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 für den Zugriff AWS IoT TwinMaker über das Netzwerk. 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 den APIs der Steuerungsebene und der Datenebene 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 APIs auf Steuerungsebene. z. B. `api.iottwinmaker.us-west-1.amazonaws.com`.
- Verwenden Sie das `data` Präfix für Datenebenen-APIs. Zum Beispiel `data.iottwinmaker.us-west-1.amazonaws.com`.

# Dokumentverlauf für das AWS IoT TwinMaker-Benutzerhandbuch

In der folgenden Tabelle werden die Dokumentationsversionen für beschrieben AWS IoT TwinMaker.

Änderung	Beschreibung	Datum
<a href="#">Neue serviceverknüpfte Rolle und neue IAM-Richtlinie</a>	AWS IoT TwinMaker hat eine neue serviceverknüpfte Rolle namens hinzugefügt <a href="#">AWSServiceRoleForIoT TwinMaker</a> . hat diese neue serviceverknüpfte Rolle AWS IoT TwinMaker hinzugefügt, damit andere -AWSServices AWS IoT TwinMaker aufrufen und ihre Ressourcen in Ihrem Namen synchronisieren kann. Die neue AWS IoT TwinMakerServiceRolePolicy IAM-Richtlinie ist dieser Rolle zugeordnet, und die Richtlinie erteilt die Berechtigung AWS IoT TwinMaker zum Aufrufen anderer -AWSServices und zum Synchronisieren ihrer Ressourcen in Ihrem Namen.	17. November 2023
<a href="#">Erstversion</a>	Erstveröffentlichung des AWS IoT TwinMaker-Benutzerhandbuchs	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.