



Guide de l'utilisateur

# AWS IoT TwinMaker



# AWS IoT TwinMaker: Guide de l'utilisateur

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Les marques et la présentation commerciale d'Amazon ne peuvent être utilisées en relation avec un produit ou un service qui n'est pas d'Amazon, d'une manière susceptible de créer une confusion parmi les clients, ou d'une manière qui dénigre ou discrédite Amazon. Toutes les autres marques commerciales qui ne sont pas la propriété d'Amazon appartiennent à leurs propriétaires respectifs, qui peuvent ou non être affiliés ou connectés à Amazon, ou sponsorisés par Amazon.

---

# Table of Contents

|  |    |
|--|----|
| Qu'est-ce que c'est AWS IoT TwinMaker ? .....  | 1  |
| Comment ça marche .....  | 1  |
| Concepts et composants clés .....  | 2  |
| Espace de travail .....  | 3  |
| Modèle entité-composant .....  | 4  |
| Visualisation .....  | 6  |
| Commencer avec AWS IoT TwinMaker .....   | 8  |
| Création et gestion d'un rôle de service pour AWS IoT TwinMaker .....                                  | 9  |
| Attribuez la confiance .....   | 9  |
| Autorisations Amazon S3 .....  | 10 |
| Attribuer des autorisations à un compartiment Amazon S3 spécifique .....                               | 11 |
| Autorisations pour les connecteurs intégrés .....  | 12 |
| Autorisations pour un connecteur vers une source de données externe .....                              | 16 |
| Modifiez le rôle IAM de votre espace de travail pour utiliser le connecteur de données<br>Athena ..... | 17 |
| Création d'un espace de travail .....  | 19 |
| Créez votre première entité .....  | 21 |
| Création d'un AWS compte .....   | 24 |
| Inscrivez-vous pour un Compte AWS .....  | 24 |
| Création d'un utilisateur doté d'un accès administratif .....  | 25 |
| Utilisation et création de types de composants .....   | 27 |
| Types de composants intégrés .....   | 27 |
| Caractéristiques principales des types de AWS IoT TwinMaker composants .....                           | 28 |
| Création de définitions de propriétés .....  | 30 |
| Création de fonctions .....  | 31 |
| Exemples de types de composants .....  | 32 |
| Alarme (résumé) .....  | 32 |
| Télémetrie Timestream .....  | 33 |
| Alarme (hérite d'une alarme abstraite) .....   | 34 |
| Exemples d'équipements .....   | 35 |
| Opérations en vrac .....   | 38 |
| Concepts clés et terminologie .....  | 38 |
| AWS IoT TwinMaker metadataTransferJob fonctionnalité .....   | 39 |
| Réalisation d'opérations d'importation et d'exportation en vrac .....                                  | 40 |

|   |     |
|---|-----|
| metadataTransferJob prérequis .....   | 41  |
| Autorisations IAM .....   | 41  |
| Exécuter une opération groupée .....  | 45  |
| Gestion des erreurs .....   | 48  |
| Modèles de métadonnées d'importation .....  | 49  |
| AWS IoT TwinMaker metadataTransferJob exemples .....                              | 53  |
| AWS IoT TwinMaker schéma de tâche de transfert de métadonnées .....               | 54  |
| Connecteurs de données .....  | 72  |
| Connecteurs de données .....  | 72  |
| Connecteur d'initialisation de schéma .....                                       | 73  |
| DataReaderByEntity .....  | 74  |
| DataReaderByComponentType .....   | 75  |
| DataReader .....  | 77  |
| AttributePropertyValueReaderByEntity .....  | 78  |
| DataWriter .....  | 79  |
| Exemples .....  | 80  |
| Connecteur de données tabulaires Athena .....                                     | 89  |
| AWS IoT TwinMaker Conditions requises pour le connecteur de données Athena .....  | 89  |
| Utilisation du connecteur de données Athena .....                                 | 90  |
| Utilisation de la référence JSON du connecteur de données tabulaires Athena ..... | 94  |
| Utilisation du connecteur de données Athena .....                                 | 95  |
| Visualisez les données tabulaires d'Athena dans Grafana .....                     | 96  |
| AWS IoT TwinMaker connecteur de données chronologiques .....                      | 98  |
| AWS IoT TwinMaker prérequis pour les connecteurs de données chronologiques .....  | 98  |
| Contexte du connecteur de données chronologiques .....                            | 99  |
| Développement d'un connecteur de données chronologiques .....                     | 100 |
| Améliorer votre connecteur de données .....                                       | 109 |
| Test de votre connecteur .....  | 110 |
| Sécurité .....  | 110 |
| Création de AWS IoT TwinMaker ressources .....                                    | 111 |
| Quelle est la prochaine étape .....   | 112 |
| AWS IoT TwinMaker connecteur de données Cookie Factory .....                      | 112 |
| Création de AWS IoT TwinMaker scènes .....  | 118 |
| Avant de créer des scènes .....   | 118 |
| Optimisez vos ressources avant de les importer dans AWS IoT TwinMaker .....       | 118 |
| Les meilleures pratiques en matière de performance dans AWS IoT TwinMaker .....   | 119 |

|  |     |
|--|-----|
| En savoir plus .....   | 120 |
| Téléchargement de ressources dans AWS IoT TwinMaker .....                                | 120 |
| Téléchargez des fichiers dans la bibliothèque de ressources à l'aide de la console ..... | 120 |
| Créer vos scènes .....   | 121 |
| Utilisez la navigation 3D AWS IoT TwinMaker dans vos scènes .....                        | 122 |
| Ajouter des caméras fixes .....  | 124 |
| Édition améliorée .....  | 124 |
| Placement ciblé des objets de scène .....  | 125 |
| Sélection du sous-modèle .....   | 125 |
| Modifier les entités dans la hiérarchie des scènes .....                                 | 126 |
| Ajouter des annotations aux entités .....  | 127 |
| Ajouter des superpositions aux balises .....   | 132 |
| Modifiez vos scènes .....  | 140 |
| Ajouter des modèles .....  | 140 |
| Ajouter des widgets .....  | 141 |
| Ajout de balises .....   | 145 |
| Optimisez votre modèle 3D .....  | 145 |
| Utilisation de tuiles 3D dans votre scène .....  | 145 |
| Scènes dynamiques .....  | 148 |
| Scènes statiques ou dynamiques .....   | 148 |
| Types de composants et entités de scène .....  | 149 |
| Concepts de scène dynamiques .....   | 150 |
| AWS IoT TwinMaker intégration du kit d'applications .....                                | 151 |
| Changer de mode de AWS IoT TwinMaker tarification .....                                  | 152 |
| Graphe de connaissances .....  | 154 |
| AWS IoT TwinMaker concepts de base du graphe de connaissances .....                      | 154 |
| Utilisation du graphe de connaissances .....   | 155 |
| Génération d'un graphe de scène .....  | 158 |
| AWS IoT TwinMaker prérequis pour les graphes de scène .....                              | 159 |
| Liez des nœuds 3D dans votre scène .....   | 159 |
| Création d'une application Web .....   | 162 |
| Graphe de connaissances : panneau Grafana .....  | 164 |
| AWS IoT TwinMaker prérequis pour l'éditeur de requêtes .....                             | 164 |
| Graphe de connaissances : autorisations Grafana .....                                    | 164 |
| Ressources supplémentaires sur le graphe de connaissances .....                          | 170 |
| Synchronisation des actifs avec AWS IoT SiteWise .....                                   | 183 |

|   |     |
|---|-----|
| Utilisation de la synchronisation des actifs avec AWS IoT SiteWise .....                          | 183 |
| Utilisation d'un espace de travail personnalisé .....   | 184 |
| Utilisation de l'lo TSite WiseDefaultWorkspace .....  | 189 |
| Différences entre les espaces de travail personnalisés et les espaces de travail par défaut ..... | 190 |
| Ressources synchronisées depuis AWS IoT SiteWise .....  | 191 |
| Espaces de travail personnalisés et par défaut .....  | 191 |
| Espace de travail par défaut uniquement .....   | 192 |
| Ressources non synchronisées .....  | 193 |
| Utiliser des entités et des types de composants synchronisés dans AWS IoT TwinMaker ....          | 193 |
| Analyser l'état et les erreurs de synchronisation .....   | 194 |
| Synchroniser les statuts des tâches .....   | 194 |
| Supprimer une tâche de synchronisation .....  | 196 |
| Limites de synchronisation des actifs .....   | 198 |
| Configuration des tableaux de bord Grafana .....  | 200 |
| Configuration CORS .....  | 201 |
| Configuration de votre environnement Grafana .....  | 202 |
| Amazon Managed Grafana .....  | 202 |
| Grafana autogéré .....  | 203 |
| Création d'un rôle de tableau de bord .....   | 204 |
| Créer une politique IAM .....   | 204 |
| Téléchargez une vidéo depuis le périphérique .....  | 208 |
| Ajouter d'autres autorisations .....  | 208 |
| Création du rôle IAM du tableau de bord Grafana .....   | 210 |
| Politique de création d'un lecteur AWS IoT TwinMaker vidéo .....                                  | 211 |
| Limitez l'accès à vos ressources .....  | 213 |
| Limitez la portée des autorisations GET .....   | 213 |
| Réduisez la portée de AWS IoT SiteWise BatchPutAssetPropertyValue l'autorisation .....            | 215 |
| Connect les alarmes aux tableaux de bord Grafana .....  | 217 |
| AWS IoT SiteWise prérequis de configuration des alarmes .....                                     | 217 |
| Définir le rôle IAM du composant d' AWS IoT SiteWise alarme .....                                 | 218 |
| Interrogation et mise à jour via l' AWS IoT TwinMaker API .....                                   | 219 |
| Configurez votre tableau de bord Grafana pour les alarmes .....                                   | 221 |
| Utilisez le tableau de bord Grafana pour la visualisation des alarmes .....                       | 223 |
| Intégration de Matterport .....   | 226 |
| Présentation de l'intégration .....   | 227 |
| Conditions préalables à l'intégration de Matterport .....   | 228 |

|  |     |
|--|-----|
| Informations d'identification du SDK Matterport .....  | 230 |
| Stockez les informations d'identification Matterport dans AWS Secrets Manager .....                        | 231 |
| Matterport scanne des scènes AWS IoT TwinMaker .....   | 234 |
| Matterport dans votre tableau de bord Grafana AWS IoT TwinMaker .....                                      | 240 |
| Intégration de Matterport au kit d'applications AWS IoT .....  | 240 |
| Diffusion de vidéos vers AWS IoT TwinMaker .....   | 242 |
| Utilisez le connecteur Edge pour Kinesis Video Stream afin de diffuser de la vidéo AWS IoT TwinMaker ..... | 242 |
| Conditions préalables .....  | 242 |
| Création de composants vidéo pour les AWS IoT TwinMaker scènes .....                                       | 243 |
| Ajouter des vidéos et des métadonnées issues du flux vidéo Kinesis à un tableau de bord Grafana .....      | 244 |
| Utilisation de la AWS IoT TwinMaker bibliothèque Flink .....   | 246 |
| Journalisation et surveillance .....   | 247 |
| Surveillance à l'aide des CloudWatch métriques Amazon .....  | 248 |
| Métriques .....  | 248 |
| Journalisation des appels d'API avec AWS CloudTrail .....  | 251 |
| AWS IoT TwinMaker informations dans CloudTrail .....   | 251 |
| Sécurité .....   | 253 |
| Protection des données .....   | 254 |
| Chiffrement au repos .....   | 255 |
| Chiffrement en transit .....   | 255 |
| Gestion de l'identité et des accès .....   | 255 |
| Public ciblé .....   | 256 |
| Authentification par des identités .....   | 256 |
| Gestion de l'accès à l'aide de politiques .....  | 258 |
| Comment AWS IoT TwinMaker fonctionne avec IAM .....  | 260 |
| Exemples de stratégies basées sur l'identité .....   | 265 |
| Résolution des problèmes .....   | 269 |
| Utilisation des rôles liés à un service .....  | 271 |
| AWS politiques gérées .....  | 273 |
| Points de terminaison de VPC (AWS PrivateLink) .....   | 278 |
| Considérations relatives aux points de AWS IoT TwinMaker terminaison VPC .....                             | 278 |
| Création d'un point de terminaison VPC d'interface pour AWS IoT TwinMaker .....                            | 280 |
| Accès AWS IoT TwinMaker via un point de terminaison VPC d'interface .....                                  | 281 |
| Création d'une politique de point de terminaison VPC pour AWS IoT TwinMaker .....                          | 283 |

---

|   |            |
|---|------------|
| Validation de la conformité .....                       | 284        |
| Résilience .....  | 284        |
| Sécurité de l'infrastructure .....                      | 285        |
| Points de terminaison et quotas .....                   | 286        |
| AWS IoT TwinMaker points de terminaison et quotas ..... | 286        |
| Informations supplémentaires sur les terminaux .....    | 286        |
| Historique de la documentation .....                    | 287        |
| .....   | cclxxxviii |

# Qu'est-ce que c'est AWS IoT TwinMaker ?

AWS IoT TwinMaker est un AWS IoT service que vous pouvez utiliser pour créer des jumeaux numériques opérationnels de systèmes physiques et numériques. AWS IoT TwinMaker crée des visualisations numériques à l'aide de mesures et d'analyses issues de divers capteurs, caméras et applications d'entreprise du monde réel pour vous aider à suivre votre usine physique, votre bâtiment ou votre installation industrielle. Vous pouvez utiliser ces données réelles pour surveiller les opérations, diagnostiquer et corriger les erreurs, et optimiser les opérations.

Un jumeau numérique est une représentation numérique en direct d'un système et de tous ses composants physiques et numériques. Il est mis à jour dynamiquement avec des données afin d'imiter la structure, l'état et le comportement réels du système. Vous pouvez l'utiliser pour améliorer les résultats commerciaux.

Les utilisateurs finaux interagissent avec les données de votre jumeau numérique à l'aide d'une application d'interface utilisateur.

## Comment ça marche

Pour répondre aux exigences minimales relatives à la création d'un jumeau numérique, vous devez effectuer les opérations suivantes.

- Modélisez les appareils, les équipements, les espaces et les processus dans un lieu physique.
- Connectez ces modèles à des sources de données qui stockent des informations contextuelles importantes, telles que les flux de données des capteurs et des caméras.
- Créez des visualisations qui aident les utilisateurs à comprendre les données et les informations afin de prendre des décisions commerciales plus efficacement.
- Mettez les jumeaux numériques à la disposition des utilisateurs finaux afin d'améliorer les résultats commerciaux.

AWS IoT TwinMaker répond à ces défis en fournissant les fonctionnalités suivantes.

- Graphe de connaissances du système de composants d'entités : AWS IoT TwinMaker fournit des outils pour modéliser les dispositifs, les équipements, les espaces et les processus dans un graphe de connaissances.

Ce graphe de connaissances contient des métadonnées sur le système et peut se connecter à des données situées à différents emplacements. AWS IoT TwinMaker fournit des connecteurs intégrés pour les données stockées dans Kinesis Video AWS IoT SiteWise Streams et Kinesis Video Streams. Vous pouvez également créer des connecteurs personnalisés pour les données stockées dans d'autres emplacements.

Le graphe de connaissances et les connecteurs fournissent ensemble une interface unique pour interroger des données dans des emplacements disparates.

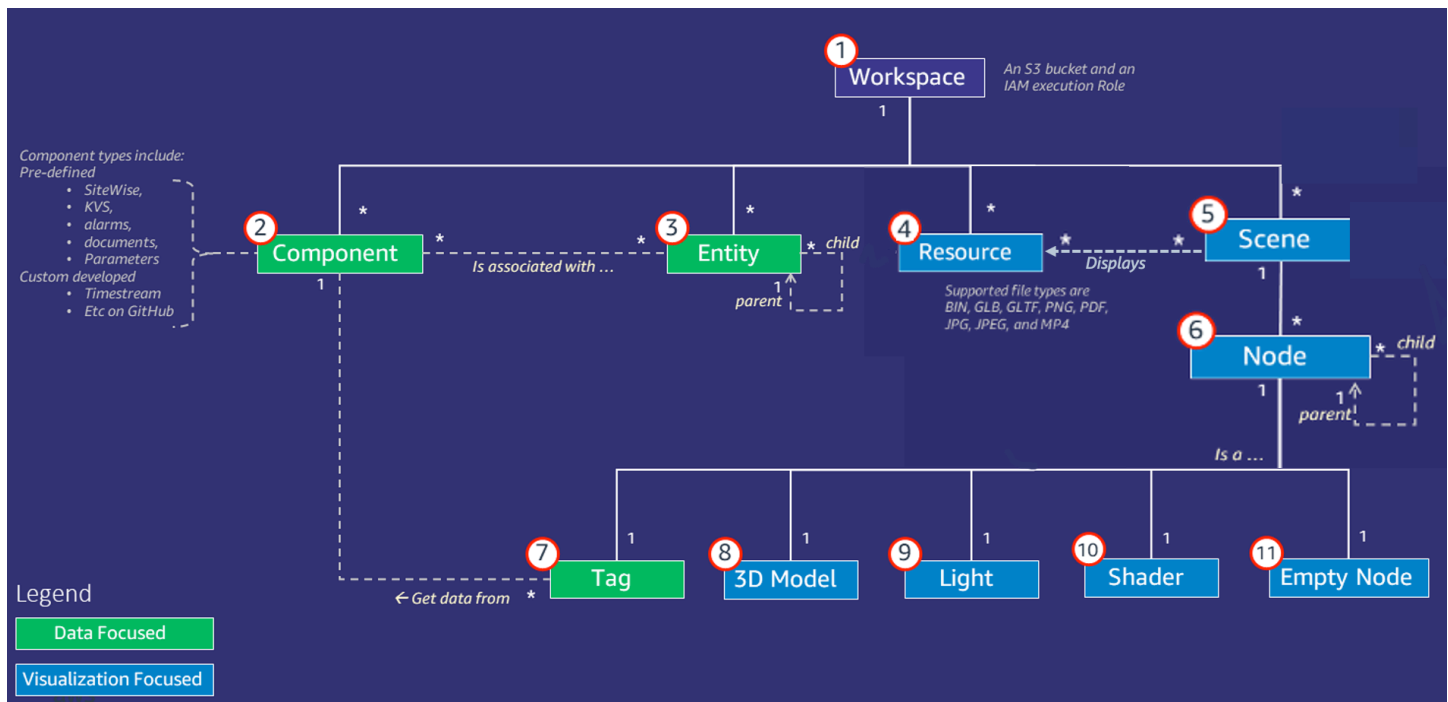
- Compositeur de scènes : la AWS IoT TwinMaker console fournit un outil de composition de scène pour créer des scènes en 3D. Vous téléchargez vos modèles 3D/CAD créés précédemment, optimisés pour l'affichage sur le Web et convertis au format .gltf ou .glb. Vous utilisez ensuite le compositeur de scène pour placer plusieurs modèles dans une seule scène, en créant des représentations visuelles de leurs opérations.

Vous pouvez également superposer des données dans la scène. Par exemple, vous pouvez créer une étiquette dans un emplacement de scène qui se connecte aux données de température d'un capteur. Cela associe les données à l'emplacement.

- Applications : AWS IoT TwinMaker fournit un plug-in pour Grafana et Amazon Managed Grafana que vous pouvez utiliser pour créer des applications de tableau de bord pour les utilisateurs finaux.
- Outils tiers : Mendix s'associe AWS IoT TwinMaker pour fournir des solutions complètes pour l'IoT industriel. Consultez l'atelier [Lean Daily Management Application with Mendix et AWS IoT TwinMaker](#) pour commencer à utiliser la plateforme de développement d'applications Low Code (LCAP) Mendix avec des AWS services tels que AWS IoT TwinMaker Kinesis Video Streams et AWS IoT SiteWise

## Concepts et composants clés

Le schéma suivant illustre la façon dont les concepts clés AWS IoT TwinMaker s'intègrent.



### Note

Les astérisques (\*) dans le diagramme indiquent one-to-many les relations. Pour les quotas pour chacune de ces relations, consultez la section [AWS IoT TwinMaker Points de terminaison et quotas](#).

Les sections suivantes décrivent les concepts illustrés dans le diagramme.

## Espace de travail

Un espace de travail est un conteneur de haut niveau pour votre application de jumelage numérique. Vous créez un ensemble logique d'entités, de composants, de ressources de scène et d'autres ressources pour votre jumeau numérique dans cet espace de travail. Il sert également de limite de sécurité pour gérer l'accès à l'application jumelle numérique et aux ressources qu'elle contient. Chaque espace de travail est lié au compartiment Amazon S3 dans lequel les données de votre espace de travail sont stockées. Vous utilisez les rôles IAM pour restreindre l'accès à votre espace de travail.

Un espace de travail peut contenir plusieurs composants, entités, scènes et ressources. Un type de composant, une entité, une scène ou une ressource n'existe que dans un seul espace de travail.

## Modèle entité-composant

AWS IoT TwinMaker fournit des outils que vous pouvez utiliser pour modéliser votre système à l'aide d'un graphe de entity-component-based connaissances. Vous pouvez utiliser l'architecture entité-composant pour créer une représentation de votre système physique. Ce modèle de composant d'entité comprend des entités, des composants et des relations. Pour plus d'informations sur les systèmes de composants d'entités, consultez la section Système de [composants d'entité](#).

### Entité

Les entités sont des représentations numériques des éléments d'un jumeau numérique qui capturent les capacités de cet élément. Cet élément peut être un équipement physique, un concept ou un processus. Les entités sont associées à des composants. Ces composants fournissent des données et un contexte à l'entité associée.

Vous pouvez AWS IoT TwinMaker ainsi organiser les entités dans des hiérarchies personnalisées pour une gestion plus efficace. La vue par défaut du système d'entités et de composants est hiérarchique.

### Composant

Les composants fournissent le contexte et les données pour les entités d'une scène. Vous ajoutez des composants aux entités. La durée de vie d'un composant est liée à la durée de vie d'une entité.

Les composants peuvent ajouter des données statiques, telles qu'une liste de documents ou les coordonnées d'un emplacement géographique. Ils peuvent également avoir des fonctions qui se connectent à d'autres systèmes, notamment des systèmes contenant des données de séries chronologiques, tels que d'autres AWS IoT SiteWise historiques de séries chronologiques sur le cloud.

Les composants sont définis par des documents JSON qui décrivent la connexion entre une source de données et AWS IoT TwinMaker. Les composants peuvent décrire des sources de données externes ou des sources de données intégrées à AWS IoT TwinMaker. Un composant accède à une source de données externe à l'aide d'une fonction Lambda spécifiée dans le document JSON. Un espace de travail peut contenir de nombreux composants. Les composants fournissent des données aux balises par le biais d'entités associées.

AWS IoT TwinMaker fournit plusieurs composants intégrés que vous pouvez ajouter depuis la console. Vous pouvez également créer vos propres composants personnalisés pour vous connecter à des sources de données telles que la télémétrie temporelle et les coordonnées géospatiales. La

TimeStream télémétrie, les composants géospatiaux et les connecteurs vers des sources de données tierces telles que Snowflake en sont des exemples.

AWS IoT TwinMaker fournit les types de composants intégrés suivants pour les cas d'utilisation courants :

- Document, tel que des manuels d'utilisation ou des images situés sur des URL spécifiées.
- Séries chronologiques, telles que les données des capteurs provenant de AWS IoT SiteWise.
- Alarmes, telles que les alarmes chronologiques provenant de sources de données externes.
- Vidéo, à partir de caméras IP connectées à Kinesis Video Streams.
- Composants personnalisés pour se connecter à des sources de données supplémentaires. Par exemple, vous pouvez créer un connecteur personnalisé pour connecter vos AWS IoT TwinMaker entités à des séries chronologiques stockées en externe.

## Sources de données

Une source de données est l'emplacement des données sources de votre jumeau numérique. AWS IoT TwinMaker prend en charge deux types de sources de données :

- Connecteurs hiérarchiques, qui vous permettent de synchroniser en permanence un modèle externe avec AWS IoT TwinMaker.
- Connecteurs de séries chronologiques, qui vous permettent de vous connecter à des bases de données de séries chronologiques telles que AWS IoT SiteWise

## Propriété

Les propriétés sont les valeurs, statiques ou basées sur des séries chronologiques, contenues dans les composants. Lorsque vous ajoutez des composants à des entités, les propriétés du composant décrivent en détail l'état actuel de l'entité.

AWS IoT TwinMaker prend en charge trois types de propriétés :

- Valeur unique, non-time-series propriétés — Ces propriétés sont généralement des paires clé-valeur statiques qui sont directement stockées AWS IoT TwinMaker avec les métadonnées de l'entité associée.
- Propriétés des séries chronologiques : AWS IoT TwinMaker stocke une référence au magasin des séries chronologiques pour ces propriétés. La valeur par défaut est la plus récente.

- Propriétés de relation : ces propriétés stockent une référence à une autre entité ou à un autre composant. Par exemple, `seen_by` il s'agit d'un composant relationnel qui peut relier une entité de caméra à une autre entité directement visualisée par cette caméra.

Vous pouvez interroger les valeurs de propriétés dans des sources de données hétérogènes à l'aide de l'interface de requête de données unifiée.

## Visualisation

Vous l'utilisez AWS IoT TwinMaker pour augmenter une représentation tridimensionnelle de votre jumeau numérique, puis vous la visualisez dans Grafana. Pour créer des scènes, utilisez des fichiers CAO ou d'autres types de fichiers 3D existants. Vous utilisez ensuite des superpositions de données pour ajouter des données pertinentes pour votre jumeau numérique.

## Scènes

Les scènes sont des représentations tridimensionnelles qui fournissent un contexte visuel aux données auxquelles elles sont connectées AWS IoT TwinMaker. Les scènes peuvent être créées en utilisant un seul modèle 3D gltf (GL Transmission Format) ou glb pour l'ensemble de l'environnement, ou en utilisant une composition de plusieurs modèles. Les scènes incluent également des balises pour indiquer les points d'intérêt de la scène.

Les scènes sont les conteneurs de premier niveau pour les visualisations. Une scène est composée d'un ou de plusieurs nœuds.

Un espace de travail peut contenir plusieurs scènes. Par exemple, un espace de travail peut contenir une scène pour chaque étage d'une installation.

## Ressources

Les scènes affichent des ressources, qui sont affichées sous forme de nœuds dans la AWS IoT TwinMaker console. Une scène peut contenir de nombreuses ressources.

Les ressources sont des images et des modèles tridimensionnels glTF basés sur des modèles tridimensionnels utilisés pour créer une scène. Une ressource peut représenter un seul équipement ou un site complet.

Vous placez des ressources dans une scène en téléchargeant un fichier .gltf ou .glb dans la bibliothèque de ressources de votre espace de travail, puis en les ajoutant à votre scène.

## Interface utilisateur augmentée

**AWS IoT TwinMaker** Vous pouvez ainsi compléter vos scènes avec des superpositions de données qui ajoutent du contexte et des informations importants, telles que les données des capteurs, aux emplacements de la scène.

**Nœuds** : les nœuds sont des instances de balises, de lumières et de modèles tridimensionnels. Ils peuvent également être vides pour ajouter de la structure à la hiérarchie de vos scènes. Par exemple, vous pouvez regrouper plusieurs nœuds sous un seul nœud vide.

**Balises** : une balise est un type de nœud qui représente les données d'un composant (via une entité). Un tag ne peut être associé qu'à un seul composant. Une balise est une annotation ajoutée à une position de  $x, y, z$  coordonnées spécifique d'une scène. La balise connecte cette partie de scène au graphe de connaissances à l'aide d'une propriété d'entité. Vous pouvez utiliser une balise pour configurer le comportement ou l'apparence visuelle d'un élément de la scène, comme une alarme.

**Lumières** : vous pouvez ajouter des lumières à une scène pour mettre en évidence certains objets, ou projeter des ombres sur des objets pour indiquer leur emplacement physique.

**Modèles tridimensionnels** : un modèle tridimensionnel est une représentation visuelle d'un fichier .gltf ou .glb importé en tant que ressource.

### Note

AWS IoT TwinMaker n'est pas destiné à être utilisé dans ou en association avec le fonctionnement d'environnements dangereux ou de systèmes critiques susceptibles d'entraîner des blessures corporelles graves ou la mort ou de causer des dommages environnementaux ou matériels.

L'exactitude des données collectées lors de votre utilisation de AWS IoT TwinMaker doit être évaluée en fonction de votre cas d'utilisation. AWS IoT TwinMaker ne doit pas être utilisé comme substitut à la surveillance humaine des systèmes physiques dans le but d'évaluer si ces systèmes fonctionnent en toute sécurité.

# Commencer avec AWS IoT TwinMaker

Les rubriques de cette section décrivent comment effectuer les opérations suivantes.

- Créez et configurez un nouvel espace de travail.
- Créez une entité et ajoutez-y un composant.

Prérequis :

Pour créer votre premier espace de travail et votre première scène, vous avez besoin des AWS ressources suivantes.

- Un [compte AWS](#).
- Un rôle de service IAM pour AWS IoT TwinMaker. Ce rôle est automatiquement généré par défaut lorsque vous créez un nouvel AWS IoT TwinMaker espace de travail dans la [AWS IoT TwinMaker console](#).

Si vous choisissez de ne pas autoriser la création AWS IoT TwinMaker automatique d'un nouveau rôle de service IAM, vous devez en spécifier un que vous avez déjà créé.

Pour obtenir des instructions sur la création et la gestion de ce rôle de service, consultez [???](#).

Pour plus d'informations sur les rôles de service IAM, voir [Création d'un rôle pour déléguer des autorisations à un Service AWS](#).

## Important

Ce rôle de service doit être associé à une politique autorisant le service à lire et à écrire dans un compartiment Amazon S3. AWS IoT TwinMaker utilise ce rôle pour accéder à d'autres services en votre nom. Vous devrez également établir une relation de confiance entre ce rôle AWS IoT TwinMaker afin que le service puisse assumer ce rôle. Si votre jumeau interagit avec d'autres AWS services, ajoutez également les autorisations nécessaires pour ces services.

Rubriques

- [Création et gestion d'un rôle de service pour AWS IoT TwinMaker](#)

- [Création d'un espace de travail](#)
- [Créez votre première entité](#)
- [Création d'un AWS compte](#)

## Création et gestion d'un rôle de service pour AWS IoT TwinMaker

AWS IoT TwinMaker nécessite que vous utilisiez un rôle de service pour lui permettre d'accéder aux ressources d'autres services en votre nom. Ce rôle doit entretenir une relation de confiance avec AWS IoT TwinMaker. Lorsque vous créez un espace de travail, vous devez lui attribuer ce rôle. Cette rubrique contient des exemples de politiques qui vous montrent comment configurer les autorisations pour les scénarios courants.

### Attribuez la confiance

La politique suivante établit une relation de confiance entre votre rôle et AWS IoT TwinMaker. Attribuez cette relation de confiance au rôle que vous utilisez pour votre espace de travail.

JSON

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

## Autorisations Amazon S3

La politique suivante permet à votre rôle de lire, de supprimer et d'écrire dans un compartiment Amazon S3. Les espaces de travail stockent les ressources dans Amazon S3. Les autorisations Amazon S3 sont donc requises pour tous les espaces de travail.

### JSON

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

**Note**

Lorsque vous créez un espace de travail, un fichier est AWS IoT TwinMaker créé dans votre compartiment Amazon S3 qui indique qu'il est utilisé par un espace de travail. Cette politique AWS IoT TwinMaker autorise la suppression de ce fichier lorsque vous supprimez l'espace de travail.

AWS IoT TwinMaker place d'autres objets liés à votre espace de travail. Il est de votre responsabilité de supprimer ces objets lorsque vous supprimez un espace de travail.

## Attribuer des autorisations à un compartiment Amazon S3 spécifique

Lorsque vous créez un espace de travail dans la AWS IoT TwinMaker console, vous pouvez choisir de AWS IoT TwinMaker créer un compartiment Amazon S3 pour vous. Vous pouvez trouver des informations sur ce bucket à l'aide de la AWS CLI commande suivante.

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

L'exemple suivant montre le format de sortie de cette commande.

```
{
  "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"
}
```

Pour mettre à jour votre politique afin qu'elle attribue des autorisations pour un compartiment Amazon S3 spécifique, utilisez la valeur de *bucket name*.

La politique suivante permet à votre rôle de lire, de supprimer et d'écrire dans un compartiment Amazon S3 spécifique.

## JSON

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

## Autorisations pour les connecteurs intégrés

Si votre espace de travail interagit avec d'autres AWS services à l'aide de connecteurs intégrés, vous devez inclure les autorisations pour ces services dans cette politique. Si vous utilisez le type de composant `com.amazon.iotsitewise.connector`, vous devez inclure des autorisations pour AWS IoT SiteWise. Pour plus d'informations sur les types de composants, consultez [???](#).

**Note**

Si vous interagissez avec d'autres AWS services à l'aide d'un type de composant personnalisé, vous devez accorder au rôle l'autorisation d'exécuter la fonction Lambda qui implémente la fonction dans votre type de composant. Pour de plus amples informations, veuillez consulter [???](#).

L'exemple suivant montre comment inclure AWS IoT SiteWise dans votre politique.

## JSON

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

```

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

```

Si vous utilisez le type de composant `com.amazon.iotsitewise.connector` et que vous devez lire des données de propriété AWS IoT SiteWise, vous devez inclure l'autorisation suivante dans votre politique.

```

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

```

Si vous utilisez le type de composant `com.amazon.iotsitewise.connector` et que vous devez y écrire des données de propriété AWS IoT SiteWise, vous devez inclure l'autorisation suivante dans votre politique.

```

...
{
  "Action": [

```

```

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

```

Si vous utilisez le type de composant `com.amazon.iotsitewise.connector.edgevideo`, vous devez inclure les autorisations pour et Kinesis Video Streams. AWS IoT SiteWise L'exemple de politique suivant montre comment inclure AWS IoT SiteWise les autorisations Kinesis Video Streams dans votre politique.

```

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

## Autorisations pour un connecteur vers une source de données externe

Si vous créez un type de composant qui utilise une fonction qui se connecte à une source de données externe, vous devez autoriser votre rôle de service à utiliser la fonction Lambda qui implémente la fonction. Pour plus d'informations sur la création de types de composants et de fonctions, consultez [???](#).

L'exemple suivant autorise votre rôle de service à utiliser une fonction Lambda.

JSON

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": [  
        "s3:GetBucket*",  
        "s3:GetObject",  
        "s3:ListBucket",  
        "s3:PutObject"  
      ],  
      "Resource": [  
        "arn:aws:s3:::amzn-s3-demo-bucket",  
        "arn:aws:s3:::amzn-s3-demo-bucket/*"  
      ]  
    },  
    {  
      "Action": [  
        "lambda:invokeFunction"  
      ],  
      "Resource": [  
        "arn:aws:lambda:us-east-1:111122223333:function:example-function"  
      ],  
      "Effect": "Allow"  
    }  
  ]  
}
```

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

Pour plus d'informations sur la création de rôles et leur attribution de politiques et de relations de confiance à l'aide de la console IAM, de l' AWS CLI API IAM, voir [Création d'un rôle pour déléguer des autorisations à un Service AWS](#).

## Modifiez le rôle IAM de votre espace de travail pour utiliser le connecteur de données Athena

Pour utiliser le [connecteur de données tabulaires AWS IoT TwinMaker Athena](#), vous devez mettre à jour le rôle IAM de votre AWS IoT TwinMaker espace de travail. Ajoutez les autorisations suivantes au rôle IAM de votre espace de travail :

### Note

Cette modification IAM ne fonctionne que pour les données tabulaires Athena stockées avec AWS Glue Amazon S3. Pour utiliser Athena avec d'autres sources de données, vous devez configurer un rôle IAM pour Athena, voir Gestion des [identités et des accès](#) dans 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
```

Lisez la section [Gestion des identités et des accès dans Athena](#) pour plus d'informations sur la configuration d'Athena IAM.

## Création d'un espace de travail

Pour créer et configurer votre premier espace de travail, procédez comme suit.

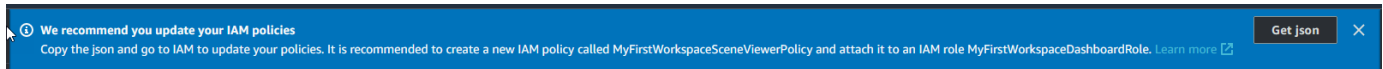
### Note

Cette rubrique explique comment créer un espace de travail simple avec une seule ressource. Pour un espace de travail complet avec plusieurs ressources, essayez l'exemple de configuration dans le référentiel Github d'[AWS IoT TwinMaker exemples](#).

1. Sur la page d'accueil de la [AWS IoT TwinMaker console](#), choisissez Workspaces dans le volet de navigation de gauche.
2. Sur la page Espaces de travail, choisissez Créer un espace de travail.
3. Sur la page Créer un espace de travail, entrez le nom de votre espace de travail.
4. (Facultatif) Ajoutez une description de votre espace de travail.
5. Sous ressource S3, choisissez Create an S3 bucket. Cette option crée un compartiment Amazon S3 AWS IoT TwinMaker dans lequel sont stockées les informations et les ressources relatives à l'espace de travail. Chaque espace de travail nécessite son propre compartiment.
6. Sous Rôle d'exécution, sélectionnez Générer automatiquement un nouveau rôle ou le rôle IAM personnalisé que vous avez créé pour cet espace de travail.

Si vous choisissez Générer automatiquement un nouveau rôle, AWS IoT TwinMaker associe une politique au rôle qui autorise le nouveau rôle de service à accéder à d'autres AWS services, y compris l'autorisation de lire et d'écrire dans le compartiment Amazon S3 que vous avez spécifié à l'étape précédente. Pour plus d'informations sur l'attribution d'autorisations à ce rôle, consultez???

7. Choisissez Créer un espace de travail. La bannière suivante apparaît en haut de la page Workspaces.



8. Choisissez Get json. Nous vous recommandons d'ajouter la politique IAM que vous voyez au rôle IAM AWS IoT TwinMaker créé pour les utilisateurs et les comptes qui consultent le tableau de bord Grafana. Le nom de ce rôle suit le modèle suivant : *workspace-name* DashboardRole pour obtenir des instructions sur la façon de créer une politique et de l'associer à un rôle, voir [Modifier une politique d'autorisations de rôle \(console\)](#).

L'exemple suivant contient la politique à ajouter au rôle de tableau de bord.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject"
      ],
      "Resource": [
        "arn:aws:s3:::iottwinmaker-workspace-workspace-name-lower-case-123456789012",
        "arn:aws:s3:::iottwinmaker-workspace-workspace-name-lower-case-123456789012/*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "iottwinmaker:Get*",
        "iottwinmaker:List*"
      ],
      "Resource": [
        "arn:aws:iottwinmaker:us-east-1:111122223333:workspace/workspace-name",
        "arn:aws:iottwinmaker:us-east-1:111122223333:workspace/workspace-name/*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": "iottwinmaker:ListWorkspaces",
      "Resource": "*"
    }
  ]
}
```

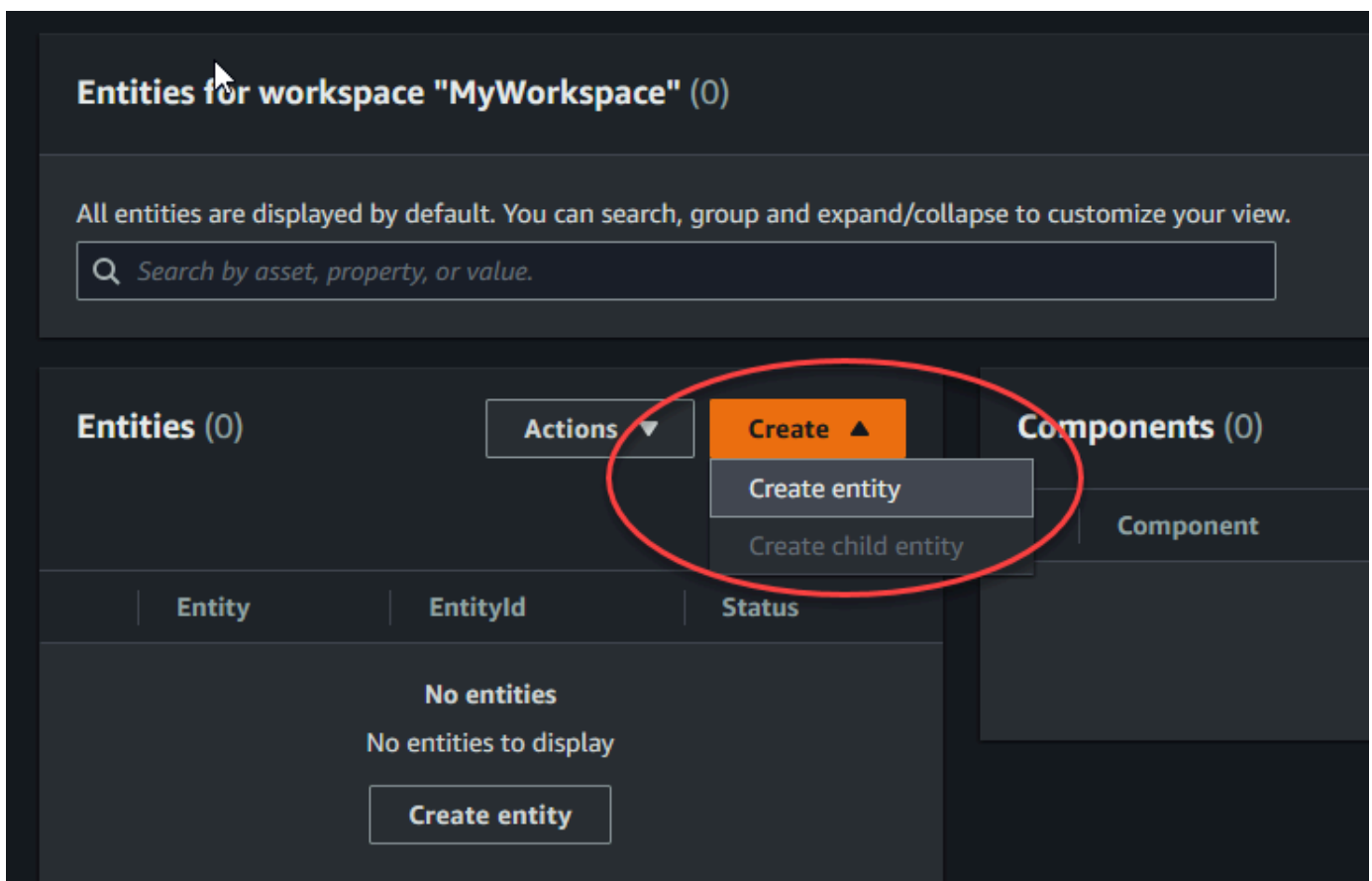
```
}  
  ]  
}
```

Vous êtes maintenant prêt à créer un modèle de données pour votre espace de travail avec votre première entité. Pour obtenir des instructions sur la façon de procéder, veuillez consulter [Créez votre première entité](#).

## Créez votre première entité

Pour créer votre première entité, procédez comme suit.

1. Sur la page Espaces de travail, choisissez votre espace de travail, puis dans le volet de gauche, sélectionnez Entités.
2. Sur la page Entités, choisissez Create, puis Create entity.



3. Dans la fenêtre Créer une entité, entrez le nom de votre entité. Cet exemple utilise une **CookieMixer** entité.

- (Facultatif) Entrez une description pour votre entité.
- Choisissez Créer une entité,

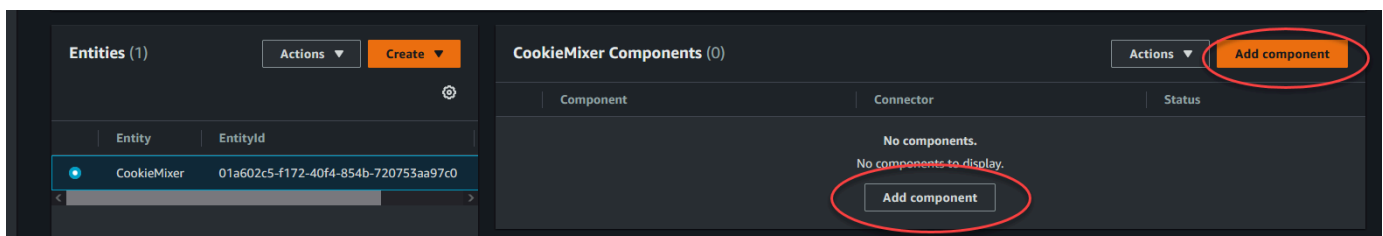
Les entités contiennent des données relatives à chaque élément de votre espace de travail. Vous insérez des données dans des entités en ajoutant des composants. AWS IoT TwinMaker fournit les types de composants intégrés suivants.

- Paramètres : ajoute un ensemble de propriétés clé-valeur.
- Document : ajoute un nom et une URL pour un document contenant des informations sur l'entité.
- Alarmes : se connecte à une source de données chronologiques d'alarmes.
- SiteWise connecteur : extrait les propriétés des séries chronologiques définies dans un AWS IoT SiteWise actif.
- Connecteur Edge pour Kinesis Video AWS IoT Greengrass Streams : extrait les données vidéo du connecteur Edge pour AWS IoT Greengrass KVS. Pour de plus amples informations, veuillez consulter [AWS IoT TwinMaker intégration vidéo](#).

Vous pouvez voir ces types de composants et leurs définitions en choisissant Types de composants dans le volet de gauche. Vous pouvez également créer un nouveau type de composant sur la page Types de composants. Pour plus d'informations sur la création de types de composants, consultez [Utilisation et création de types de composants](#).

Dans cet exemple, nous créons un composant de document simple qui ajoute des informations descriptives sur votre entité.

- Sur la page Entités, choisissez l'entité, puis choisissez Ajouter un composant.



- Dans la fenêtre Ajouter un composant, entrez le nom de votre composant. Comme cet exemple utilise une entité de mixage de cookies, nous saisissons **MixerDescription** dans le champ Nom.

**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

Value field

**Add another property**

Cancel **Add component**

3. Choisissez Ajouter un document, puis entrez des valeurs pour le nom du document et l'URL externe. Avec le composant documents, vous pouvez stocker une liste d'informations externes URLs contenant des informations importantes sur l'entité.

#### 4. Choisissez Ajouter un composant.

Vous êtes maintenant prêt à créer votre première scène. Pour obtenir des instructions sur la façon de procéder, veuillez consulter [Création et édition de AWS IoT TwinMaker scènes](#).

## Création d'un AWS compte

Si vous n'en avez pas Compte AWS, procédez comme suit pour en créer un.

Pour vous inscrire à un Compte AWS

1. Ouvrez l'<https://portal.aws.amazon.com/billing/inscription>.
2. Suivez les instructions en ligne.

Dans le cadre de la procédure d'inscription, vous recevrez un appel téléphonique ou un SMS et vous saisissez un code de vérification en utilisant le clavier numérique du téléphone.

Lorsque vous vous inscrivez à un Compte AWS, un Utilisateur racine d'un compte AWS est créé. Par défaut, seul l'utilisateur racine a accès à l'ensemble des Services AWS et des ressources de ce compte. La meilleure pratique de sécurité consiste à attribuer un accès administratif à un utilisateur, et à utiliser uniquement l'utilisateur racine pour effectuer les [tâches nécessitant un accès utilisateur racine](#).

## Inscrivez-vous pour un Compte AWS

Si vous n'en avez pas Compte AWS, procédez comme suit pour en créer un.

Pour vous inscrire à un Compte AWS

1. Ouvrez l'<https://portal.aws.amazon.com/billing/inscription>.
2. Suivez les instructions en ligne.

Dans le cadre de la procédure d'inscription, vous recevrez un appel téléphonique ou un SMS et vous saisissez un code de vérification en utilisant le clavier numérique du téléphone.

Lorsque vous vous inscrivez à un Compte AWS, un Utilisateur racine d'un compte AWS est créé. Par défaut, seul l'utilisateur racine a accès à l'ensemble des Services AWS et des ressources de ce compte. La meilleure pratique de sécurité consiste à attribuer un accès administratif à un

utilisateur, et à utiliser uniquement l'utilisateur racine pour effectuer les [tâches nécessitant un accès utilisateur racine](#).

AWS vous envoie un e-mail de confirmation une fois le processus d'inscription terminé. À tout moment, vous pouvez consulter l'activité actuelle de votre compte et gérer votre compte en accédant à <https://aws.amazon.com/> et en choisissant Mon compte.

## Création d'un utilisateur doté d'un accès administratif

Une fois que vous vous êtes inscrit à un utilisateur administratif Compte AWS, que vous Utilisez l'utilisateur racine d'un compte AWS l'avez sécurisé AWS IAM Identity Center, que vous l'avez activé et que vous en avez créé un, afin de ne pas utiliser l'utilisateur root pour les tâches quotidiennes.

Sécurisez votre Utilisateur racine d'un compte AWS

1. Connectez-vous en [AWS Management Console](#) tant que propriétaire du compte en choisissant Utilisateur root et en saisissant votre adresse Compte AWS e-mail. Sur la page suivante, saisissez votre mot de passe.

Pour obtenir de l'aide pour vous connecter en utilisant l'utilisateur racine, consultez [Connexion en tant qu'utilisateur racine](#) dans le Guide de l'utilisateur Connexion à AWS .

2. Activez l'authentification multifactorielle (MFA) pour votre utilisateur racine.

Pour obtenir des instructions, consultez la section [Activer un périphérique MFA virtuel pour votre utilisateur Compte AWS root \(console\)](#) dans le guide de l'utilisateur IAM.

Création d'un utilisateur doté d'un accès administratif

1. Activez IAM Identity Center.

Pour obtenir des instructions, consultez [Activation d' AWS IAM Identity Center](#) dans le Guide de l'utilisateur AWS IAM Identity Center .

2. Dans IAM Identity Center, octroyez un accès administratif à un utilisateur.

Pour un didacticiel sur l'utilisation du Répertoire IAM Identity Center comme source d'identité, voir [Configurer l'accès utilisateur avec la valeur par défaut Répertoire IAM Identity Center](#) dans le Guide de AWS IAM Identity Center l'utilisateur.

## Connexion en tant qu'utilisateur doté d'un accès administratif

- Pour vous connecter avec votre utilisateur IAM Identity Center, utilisez l'URL de connexion qui a été envoyée à votre adresse e-mail lorsque vous avez créé l'utilisateur IAM Identity Center.

Pour obtenir de l'aide pour vous connecter en utilisant un utilisateur d'IAM Identity Center, consultez la section [Connexion au portail AWS d'accès](#) dans le guide de l'Connexion à AWS utilisateur.

## Attribution d'un accès à d'autres utilisateurs

1. Dans IAM Identity Center, créez un ensemble d'autorisations qui respecte la bonne pratique consistant à appliquer les autorisations de moindre privilège.

Pour obtenir des instructions, consultez [Création d'un ensemble d'autorisations](#) dans le Guide de l'utilisateur AWS IAM Identity Center .

2. Attribuez des utilisateurs à un groupe, puis attribuez un accès par authentification unique au groupe.

Pour obtenir des instructions, consultez [Ajout de groupes](#) dans le Guide de l'utilisateur AWS IAM Identity Center .

# Utilisation et création de types de composants

Cette rubrique décrit les valeurs et les structures que vous utilisez pour créer un type de AWS IoT TwinMaker composant. Il explique comment créer un objet de requête que vous pouvez transmettre à l'[CreateComponentType](#) API ou en utilisant l'éditeur de type de composant de la AWS IoT TwinMaker console.

Les composants fournissent un contexte pour les propriétés et les données pour leurs entités associées.

## Types de composants intégrés

Dans la AWS IoT TwinMaker console, lorsque vous choisissez un espace de travail, puis que vous choisissez Types de composants dans le volet de gauche, les types de composants suivants s'affichent.

- `com.amazon.iotsitewise.resourcesync` : type de composant qui synchronise automatiquement vos actifs et vos modèles d' AWS IoT SiteWise actifs et les convertit en entités, composants et types de composants. AWS IoT TwinMaker Pour plus d'informations sur l'utilisation de la synchronisation AWS IoT SiteWise des actifs, consultez la section [Synchronisation des actifs avec AWS IoT SiteWise](#).
- `com.amazon.iottwinmaker.alarm.basic` : composant d'alarme de base qui extrait les données d'alarme d'une source externe vers une entité. Ce composant ne contient aucune fonction qui se connecte à une source de données spécifique. Cela signifie que le composant d'alarme est abstrait et peut être hérité par un autre type de composant qui spécifie une source de données et une fonction qui lit à partir de cette source.
- `com.amazon.iottwinmaker.documents` : mappage simple des titres aux URL des documents contenant des informations sur une entité.
- `com.amazon.iotsitewise.connector.edgevideo` : composant qui extrait la vidéo d'un appareil IoT à l'aide du composant Edge Connector for Kinesis Video Streams vers une entité. AWS IoT Greengrass Le composant [Edge Connector for Kinesis Video AWS IoT Greengrass Streams](#) n'est pas AWS IoT TwinMaker un composant, mais plutôt un composant AWS IoT Greengrass prédéfini déployé localement sur votre appareil IoT.
- `com.amazon.iotsitewise.connector` : composant qui extrait des données dans une entité. AWS IoT SiteWise

- `com.amazon.iottwinmaker.parameters` : composant qui ajoute des paires clé-valeur statiques à une entité.
- `com.amazon.kvs.video` : composant qui extrait la vidéo de Kinesis Video Streams vers une entité.

| Component types (6)                                     |             |        |  | Create component type   |
|---|-------------|--------|--|---|
| <input type="text" value="Find component types"/>       |             |        |  | <span>&lt;</span> <span>1</span> <span>&gt;</span> <span>⊙</span> |
| 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)   |   |

## Caractéristiques principales des types de AWS IoT TwinMaker composants

La liste suivante décrit les principales fonctionnalités des types de composants.

- Définitions des propriétés : L'[PropertyDefinitionRequest](#) objet définit une propriété que vous pouvez renseigner dans le compositeur de scène ou avec des données extraites de sources de données externes. Les propriétés statiques que vous définissez sont stockées dans AWS IoT TwinMaker. Les propriétés des séries chronologiques et les autres propriétés extraites de sources de données sont stockées en externe.

Vous spécifiez les définitions de propriétés dans une chaîne de caractères sur la `PropertyDefinitionRequest` carte. Chaque chaîne doit être unique à la carte.

- Fonctions : L'[FunctionRequest](#) objet spécifie une fonction Lambda qui lit et écrit potentiellement dans une source de données externe.

Un type de composant qui contient une propriété dont la valeur est stockée en externe mais qui n'a pas de fonction correspondante pour récupérer les valeurs est un type de composant abstrait. Vous pouvez étendre les types de composants concrets à partir d'un type de composant abstrait. Vous ne pouvez pas ajouter de types de composants abstraits à une entité. Ils n'apparaissent pas dans le compositeur de scène.

Vous spécifiez les fonctions à l'intérieur d'une chaîne à `FunctionRequest` mapper. La chaîne doit spécifier l'un des types de fonctions prédéfinis suivants.

- `dataReader`: fonction qui extrait des données d'une source externe.
- `dataReaderByEntity`: fonction qui extrait des données d'une source externe.

Lorsque vous utilisez ce type de lecteur de données, l'opération d'[GetPropertyValueHistoryAPI](#) ne prend en charge que les requêtes spécifiques à l'entité pour les propriétés de ce type de composant. (Vous ne pouvez demander l'historique des valeurs de la propriété que pour `componentName +entityId`.)

- `dataReaderByComponentType`: fonction qui extrait des données d'une source externe.

Lorsque vous utilisez ce type de lecteur de données, l'opération d'[GetPropertyValueHistoryAPI](#) prend uniquement en charge les requêtes entre entités pour les propriétés de ce type de composant. (Vous ne pouvez demander l'historique des valeurs de la propriété que pour `componentTypeId`.)

- `dataWriter`: fonction qui écrit des données dans une source externe.
- `schemaInitializer`: fonction qui initialise automatiquement les valeurs des propriétés chaque fois que vous créez une entité contenant le type de composant.

L'un des trois types de fonctions de lecture de données est requis dans un type de composant non abstrait.

[Pour un exemple de fonction Lambda qui implémente des composants de télémétrie par flux temporel, y compris des alarmes, consultez le lecteur de données dans Samples.AWS IoT TwinMaker](#)

#### Note

Comme le connecteur d'alarme hérite du type de composant d'alarme abstrait, la fonction Lambda doit renvoyer `alarm_key` la valeur. Si vous ne renvoyez pas cette valeur, Grafana ne la reconnaîtra pas comme une alarme. Cela est obligatoire pour tous les composants qui renvoient des alarmes.

- Héritage : les types de composants favorisent la réutilisabilité du code grâce à l'héritage. Un type de composant peut hériter d'un maximum de 10 types de composants parents.

Utilisez le `extendsFrom` paramètre pour spécifier les types de composants dont votre type de composant hérite des propriétés et des fonctions.

- `isSingleton` : Certains composants contiennent des propriétés, telles que les coordonnées de localisation, qui ne peuvent être incluses qu'une seule fois dans une entité. Définissez la valeur du `isSingleton` paramètre sur `true` pour indiquer que votre type de composant ne peut être inclus qu'une seule fois dans une entité.

## Création de définitions de propriétés

Le tableau suivant décrit les paramètres d'un `PropertyDefinitionRequest`.

| Paramètre                       | Description   |
|---------------------------------|---|
| <code>isExternalId</code>       | <p>Booléen qui indique si la propriété est un identifiant unique (tel qu'un identifiant d' AWS IoT SiteWise actif) d'une valeur de propriété stockée en externe.</p> <p>La valeur par défaut de cette propriété est <code>false</code>.</p> |
| <code>isStoredExternally</code> | <p>Booléen qui indique si la valeur de la propriété est stockée en externe.</p> <p>La valeur par défaut de cette propriété est <code>false</code>.</p>  |
| <code>isTimeSeries</code>       | <p>Booléen qui indique si la propriété stocke des données de séries chronologiques.</p> <p>La valeur par défaut de cette propriété est <code>false</code>.</p>  |
| <code>isRequiredInEntity</code> | <p>Booléen qui indique si la propriété doit avoir une valeur dans une entité utilisant le type de composant.</p>  |

| Paramètre                  | Description  |
|----------------------------|--|
| <code>dataType</code>      | <a href="#">DataType</a> Objet qui spécifie le type de données (chaîne, carte, liste et unité de mesure, par exemple) de la propriété.         |
| <code>defaultValue</code>  | <a href="#">DataValue</a> Objet qui spécifie la valeur par défaut de la propriété.   |
| <code>configuration</code> | string-to-string Carte qui indique les informations supplémentaires dont vous avez besoin pour vous connecter à une source de données externe. |

## Création de fonctions

Le tableau suivant décrit les paramètres d'un `FunctionRequest`.

| Paramètre                       | Description  |
|---------------------------------|--|
| <code>implementedBy</code>      | <a href="#">DataConnector</a> Objet qui spécifie la fonction Lambda qui se connecte à la source de données externe.  |
| <code>requiredProperties</code> | Liste des propriétés dont la fonction a besoin pour lire et écrire dans une source de données externe.   |
| <code>scope</code>              | La portée de la fonction. <code>Workspace</code> À utiliser pour les fonctions dont la portée couvre l'ensemble de l'espace de travail. <code>Entity</code> À utiliser pour les fonctions dont la portée est limitée à l'entité qui contient le composant. |

Pour des exemples montrant comment créer et étendre des types de composants, consultez [???](#).

## Exemples de types de composants

Cette rubrique contient des exemples qui montrent comment implémenter les concepts clés des types de composants.

### Alarme (résumé)

L'exemple suivant représente le type de composant d'alarme abstrait qui apparaît dans la AWS IoT TwinMaker console. Il contient une fonctions liste composée d'un dataReader qui n'a aucune implementedBy valeur.

```
{
  "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
}
}
}
```

### Remarques :

Les valeurs pour `componentTypeId` et `workspaceId` sont obligatoires. La valeur de `componentTypeId` doit être unique à votre espace de travail. La valeur de `alarm_key` est un identifiant unique qu'une fonction peut utiliser pour récupérer les données d'alarme d'une source externe. La valeur de la clé est requise et stockée dans AWS IoT TwinMaker. Les valeurs des séries `alarm_status` chronologiques sont stockées dans la source externe.

D'autres exemples sont disponibles dans [AWS IoT TwinMaker Samples](#).

## Télémetrie Timestream

L'exemple suivant est un type de composant simple qui extrait des données de télémétrie relatives à un type spécifique de composant (tel qu'une alarme ou un mixeur de cookies) à partir d'une source externe. Elle spécifie une fonction Lambda dont les types de composants héritent.

```
{
  "componentTypeId": "com.example.timestream-telemetry",
  "workspaceId": "MyWorkspace",
  "functions": {
    "dataReader": {
      "implementedBy": {
        "lambda": {
          "arn": "LambdaArn"
        }
      }
    }
  }
}
```

```

    },
    "propertyDefinitions": {
      "telemetryType": {
        "dataType": { "type": "STRING" },
        "isExternalId": false,
        "isStoredExternally": false,
        "isTimeSeries": false,
        "isRequiredInEntity": true
      },
      "telemetryId": {
        "dataType": { "type": "STRING" },
        "isExternalId": false,
        "isStoredExternally": false,
        "isTimeSeries": false,
        "isRequiredInEntity": true
      }
    }
  }
}

```

## Alarme (hérite d'une alarme abstraite)

L'exemple suivant hérite à la fois des types de composants de télémétrie abstraite et d'alarme temporelle. Il spécifie sa propre fonction Lambda qui récupère les données d'alarme.

```

{
  "componentTypeId": "com.example.cookiefactory.alarm",
  "workspaceId": "MyWorkspace",
  "extendsFrom": [
    "com.example.timestream-telemetry",
    "com.amazon.iottwinmaker.alarm.basic"
  ],
  "propertyDefinitions": {
    "telemetryType": {
      "defaultValue": {
        "stringValue": "Alarm"
      }
    }
  },
  "functions": {
    "dataReader": {
      "implementedBy": {

```

```
        "lambda": {
            "arn": "LambdaArn"
        }
    }
}
```

### Note

Comme le connecteur d'alarme hérite du type de composant d'alarme abstrait, la fonction Lambda doit renvoyer `alarm_key` la valeur. Si vous ne renvoyez pas cette valeur, Grafana ne la reconnaîtra pas comme une alarme. Cela est obligatoire pour tous les composants qui renvoient des alarmes.

## Exemples d'équipements

Les exemples présentés dans cette section montrent comment modéliser des équipements potentiels. Vous pouvez utiliser ces exemples pour avoir des idées sur la façon de modéliser des équipements dans le cadre de vos propres processus.

### Mélangeur à biscuits

L'exemple suivant hérite du type de composant de télémétrie timestream. Il spécifie des propriétés de séries chronologiques supplémentaires pour le taux de rotation et la température d'un mixeur à biscuits.

```
{
  "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
    }
}
}

```

## Réservoir d'eau

L'exemple suivant hérite du type de composant de télémétrie timestream. Il spécifie des propriétés de séries chronologiques supplémentaires pour le volume et le débit d'un réservoir d'eau.

```

{
  "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
    }
  }
}
```

## Emplacement de l'espace

L'exemple suivant contient des propriétés dont les valeurs sont stockées dans AWS IoT TwinMaker. Les valeurs étant spécifiées par les utilisateurs et stockées en interne, aucune fonction n'est requise pour les récupérer. L'exemple utilise également le type de RELATIONSHIP données pour spécifier une relation avec un autre type de composant.

Ce composant fournit un mécanisme léger pour ajouter du contexte à un jumeau numérique. Vous pouvez l'utiliser pour ajouter des métadonnées indiquant où se trouve un objet. Vous pouvez également utiliser ces informations dans le cadre de la logique utilisée pour déterminer quelles caméras peuvent voir un équipement ou un espace, ou pour savoir comment envoyer quelqu'un sur place.

```
{
  "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 opérations en vrac

Utilisez `metadataTransferJob` pour transférer et gérer vos AWS IoT TwinMaker ressources à grande échelle. `metadataTransferJob` permet d'effectuer des opérations en masse et de transférer des ressources entre AWS IoT TwinMaker AWS IoT SiteWise et Amazon S3.

Vous pouvez utiliser des opérations groupées dans les scénarios suivants :

- Migration massive des actifs et des données entre les comptes, par exemple la migration d'un compte de développement vers un compte de production.
- Gestion des actifs à grande échelle, telle que le téléchargement et la modification d' AWS IoT actifs à grande échelle.
- Importation massive de vos actifs dans AWS IoT TwinMaker et AWS IoT SiteWise.
- Importation en masse d' AWS IoT TwinMaker entités à partir de fichiers d'ontologie existants tels que des BIM fichiers `revit` ou.

## Rubriques

- [Concepts clés et terminologie](#)
- [Réalisation d'opérations d'importation et d'exportation en vrac](#)
- [AWS IoT TwinMaker schéma de tâche de transfert de métadonnées](#)

## Concepts clés et terminologie

AWS IoT TwinMaker les opérations groupées utilisent les concepts et la terminologie suivants :

- **Importer** : action qui consiste à déplacer des ressources dans un AWS IoT TwinMaker espace de travail. Par exemple, depuis un fichier local, un fichier dans un compartiment Amazon S3 ou depuis un AWS IoT TwinMaker espace AWS IoT SiteWise de travail.
- **Exporter** : action qui consiste à déplacer des ressources d'un AWS IoT TwinMaker espace de travail vers une machine locale ou un compartiment Amazon S3.
- **Source** : point de départ à partir duquel vous souhaitez déplacer les ressources.

Par exemple, un compartiment Amazon S3 est une source d'importation et un AWS IoT TwinMaker espace de travail est une source d'exportation.

- **Destination** : emplacement souhaité vers lequel vous souhaitez déplacer vos ressources.

Par exemple, un compartiment Amazon S3 est une destination d'exportation et un AWS IoT TwinMaker espace de travail est une destination d'importation.

- AWS IoT SiteWise Schéma : schéma utilisé pour importer et exporter des ressources vers et depuis AWS IoT SiteWise.
- AWS IoT TwinMaker Schéma : schéma utilisé pour importer et exporter des ressources vers et depuis AWS IoT TwinMaker.
- AWS IoT TwinMaker ressources de haut niveau : ressources utilisées dans les ressources existantes APIs. Plus précisément, une entité ou un ComponentType.
- AWS IoT TwinMaker ressources de sous-niveau : types de ressources imbriqués utilisés dans les définitions de métadonnées. Plus précisément, un composant.
- Métadonnées : informations clés requises pour réussir l'importation ou l'exportation AWS IoT SiteWise et les AWS IoT TwinMaker ressources.
- metadataTransferJob: objet créé lors de l'exécution `CreateMetadataTransferJob`.

## AWS IoT TwinMaker metadataTransferJob fonctionnalité

Cette rubrique explique le comportement à AWS IoT TwinMaker suivre lorsque vous exécutez une opération en masse : comment metadataTransferJob est traitée une. Il explique également comment définir un schéma avec les métadonnées requises pour transférer vos ressources. AWS IoT TwinMaker les opérations groupées prennent en charge les fonctionnalités suivantes :

- Création ou remplacement de ressources de haut niveau : AWS IoT TwinMaker créera de nouvelles ressources ou remplacera toutes les ressources existantes identifiées de manière unique par un identifiant de ressource.

Par exemple, si une entité existe dans le système, la définition de l'entité sera remplacée par la nouvelle définie dans le modèle sous la Entity clé.

- Création ou remplacement d'une sous-ressource :

À partir du EntityComponent niveau, vous pouvez uniquement créer ou remplacer un composant. L'entité doit déjà exister, sinon l'action produira un ValidationException.

Au niveau de la propriété ou de la relation, vous pouvez uniquement créer ou remplacer une propriété ou une relation, et le conteneur EntityComponent doit déjà exister.

- Suppression d'une sous-ressource :

AWS IoT TwinMaker prend également en charge la suppression de sous-ressources. Une sous-ressource peut être un composant, une propriété ou une relation.

Si vous souhaitez supprimer un composant, vous devez le faire au niveau de l'entité.

Si vous souhaitez supprimer une propriété ou une relation, vous devez le faire à partir de l'entité ou du EntityComponent niveau.

Pour supprimer une sous-ressource, vous devez mettre à jour la ressource de niveau supérieur et omettre la définition de la sous-ressource.

- Aucune suppression de ressources de premier niveau : ne AWS IoT TwinMaker supprimera jamais les ressources de premier niveau. Une ressource de haut niveau fait référence à une entité ou ComponentType.
- Aucune définition de sous-ressource pour la même ressource de niveau supérieur dans un modèle :

Vous ne pouvez pas fournir la définition complète de l'entité et la définition des sous-ressources (comme les propriétés) de la même entité dans le même modèle.

Si un EntityID est utilisé dans Entity, vous ne pouvez pas utiliser le même ID dans Entity EntityComponent, property ou relation.

Si une combinaison EntityId ou ComponentName est utilisée dans EntityComponent, vous ne pouvez pas utiliser la même combinaison dans EntityComponent, propriété ou relation.

Si une combinaison EntityId, ComponentName, PropertyName est utilisée dans une propriété ou une relation, vous ne pouvez pas utiliser la même combinaison dans la propriété ou la relation.

- ExternalId est facultatif pour AWS IoT TwinMaker : Il ExternalId peut être utilisé pour vous aider à identifier vos ressources.

## Réalisation d'opérations d'importation et d'exportation en vrac

Cette rubrique explique comment effectuer des opérations d'importation et d'exportation en masse et comment gérer les erreurs dans vos tâches de transfert. Il fournit des exemples de tâches de transfert utilisant des commandes CLI.

La référence AWS IoT TwinMaker d'API contient des informations sur les actions d'API [CreateMetadataTransferJob](#) et sur les autres actions de l'API.

## Rubriques

- [metadataTransferJob prérequis](#)
- [Autorisations IAM](#)
- [Exécuter une opération groupée](#)
- [Gestion des erreurs](#)
- [Modèles de métadonnées d'importation](#)
- [AWS IoT TwinMaker metadataTransferJob exemples](#)

## metadataTransferJob prérequis

Veillez remplir les conditions préalables suivantes avant d'exécuter un metadataTransferJob :

- Créez un AWS IoT TwinMaker espace de travail. L'espace de travail peut être la destination d'importation ou la source d'exportation d'un metadataTransferJob. Pour plus d'informations sur la création d'un espace de travail, voir [Création d'un espace de travail](#).
- Créez un compartiment Amazon S3 pour stocker les ressources. Pour plus d'informations sur l'utilisation d'Amazon S3, consultez [Qu'est-ce qu'Amazon S3 ?](#)

## Autorisations IAM

Lorsque vous effectuez des opérations en masse, vous devez créer une politique IAM avec des autorisations permettant l'échange de AWS ressources entre Amazon S3, AWS IoT TwinMaker AWS IoT SiteWise, et votre machine locale. Pour plus d'informations sur la création de politiques IAM, consultez la section [Création de politiques IAM](#).

Les déclarations de politique relatives à AWS IoT TwinMaker Amazon S3 AWS IoT SiteWise et à Amazon S3 sont répertoriées ici :

- AWS IoT TwinMaker politique :

JSON

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

```

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

```

- AWS IoT SiteWise politique :

JSON

```

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

```

```

        "s3:AbortMultipartUpload",
        "s3:ListBucketMultipartUploads",
        "s3:ListMultipartUploadParts"
    ],
    "Resource": "*"
  },
  {
    "Effect": "Allow",
    "Action": [
      "iotsitewise:CreateAsset",
      "iotsitewise:CreateAssetModel",
      "iotsitewise:UpdateAsset",
      "iotsitewise:UpdateAssetModel",
      "iotsitewise:UpdateAssetProperty",
      "iotsitewise:ListAssets",
      "iotsitewise:ListAssetModels",
      "iotsitewise:ListAssetProperties",
      "iotsitewise:ListAssetModelProperties",
      "iotsitewise:ListAssociatedAssets",
      "iotsitewise:DescribeAsset",
      "iotsitewise:DescribeAssetModel",
      "iotsitewise:DescribeAssetProperty",
      "iotsitewise:AssociateAssets",
      "iotsitewise:DisassociateAssets",
      "iotsitewise:AssociateTimeSeriesToAssetProperty",
      "iotsitewise:DisassociateTimeSeriesFromAssetProperty",
      "iotsitewise:BatchPutAssetPropertyValue",
      "iotsitewise:BatchGetAssetPropertyValue",
      "iotsitewise:TagResource",
      "iotsitewise:UntagResource",
      "iotsitewise:ListTagsForResource"
    ],
    "Resource": "*"
  }
]
}

```

- Politique Amazon S3 :

```

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

```

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

Vous pouvez également définir votre politique Amazon S3 de manière à n'accéder qu'à un seul compartiment Amazon S3. Consultez la politique suivante.

### Politique d'Amazon S3 relative à un compartiment unique

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

## Définissez le contrôle d'accès pour un metadataTransferJob

Pour contrôler le type de tâches auxquelles un utilisateur peut accéder, ajoutez la politique IAM suivante au rôle utilisé pour appeler AWS IoT TwinMaker.

### Note

Cette politique autorise uniquement l'accès aux tâches AWS IoT TwinMaker d'importation et d'exportation qui transfèrent des ressources vers et depuis Amazon S3.

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

## Exécuter une opération groupée

Cette section explique comment effectuer des opérations d'importation et d'exportation en masse.

Importez des données depuis Amazon S3 vers AWS IoT TwinMaker

1. Spécifiez les ressources que vous souhaitez transférer à l'aide du AWS IoT TwinMaker metadataTransferJob schéma. Créez et stockez votre fichier de schéma dans votre compartiment Amazon S3.

Pour des exemples de schémas, voir [Modèles de métadonnées d'importation](#).

2. Créez un corps de requête et enregistrez-le sous forme de fichier JSON. Le corps de la demande indique la source et la destination de la tâche de transfert. Assurez-vous de spécifier votre compartiment Amazon S3 comme source et votre AWS IoT TwinMaker espace de travail comme destination.

Voici un exemple de corps de demande :

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

```

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

```

Enregistrez le nom de fichier que vous avez indiqué dans le corps de votre demande, vous en aurez besoin à l'étape suivante. Dans cet exemple, le corps de la demande est nommé `createMetadataTransferJobImport.json`.

3. Exécutez la commande CLI suivante pour appeler `CreateMetadataTransferJob` (remplacez le nom du fichier `input-json` par le nom que vous avez donné au corps de votre demande) :

```

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

```

Cela crée un `metadataTransferJob` et commence le processus de transfert des ressources que vous avez sélectionnées.

## Exporter des données depuis AWS IoT TwinMaker Amazon S3

1. Créez un corps de requête JSON avec les filtres appropriés pour choisir les ressources que vous souhaitez exporter. Pour cet exemple, nous utilisons :

```

{
  "metadataTransferJobId": "your-transfer-job-Id",
  "sources": [{
    "type": "iottwinmaker",
    "iotTwinMakerConfiguration": {
      "workspace": "arn:aws:iottwinmaker:us-
east-1:111122223333:workspace/your-workspace-name",
      "filters": [{
        "filterByEntity": {

```

```

        "entityId": "parent"
      }},
      {
        "filterByEntity": {
          "entityId": "child"
        }},
      {
        "filterByComponentType": {
          "componentTypeId": "component.type.minimal"
        }
      }
    ]
  },
  "destination": {
    "type": "s3",
    "s3Configuration": {
      "location": "arn:aws:s3:::amzn-s3-demo-bucket"
    }
  }
}

```

Le `filters` tableau vous permet de spécifier les ressources qui seront exportées. Dans cet exemple, nous filtrons `parent` entity, et `componentType`.

Assurez-vous de spécifier votre AWS IoT TwinMaker espace de travail comme source et votre compartiment Amazon S3 comme destination de la tâche de transfert de métadonnées.

Enregistrez le corps de votre demande et enregistrez le nom du fichier, vous en aurez besoin à l'étape suivante. Dans cet exemple, nous avons nommé le corps de notre demande `createMetadataTransferJobExport.json`.

2. Exécutez la commande CLI suivante pour appeler `CreateMetadataTransferJob` (remplacez le nom du fichier `input-json` par le nom que vous avez donné au corps de votre demande) :

```

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

```

Cela crée un `metadataTransferJob` et commence le processus de transfert des ressources que vous avez sélectionnées.

Pour vérifier ou mettre à jour le statut d'une tâche de transfert, utilisez les commandes suivantes :

- Pour annuler une tâche, utilisez l'action [CancelMetadataTransferJobAPI](#). Lorsque vous appelez `CancelMetadataTransferJob`, l'API annule uniquement une exécution `metadataTransferJob`, et les ressources déjà exportées ou importées ne sont pas affectées par cet appel d'API.
- Pour récupérer des informations sur une tâche spécifique, utilisez l'action [GetMetadataTransferJobAPI](#).

Vous pouvez également faire appel `GetMetadataTransferJob` à une tâche de transfert existante à l'aide de la commande CLI suivante :

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

Si vous faites appel `GetMetadataTransferJob` à une tâche AWS IoT TwinMaker d'importation ou d'exportation inexistante, vous recevez une `ResourceNotFoundException` erreur en réponse.

- Pour répertorier les tâches en cours, utilisez l'action [ListMetadataTransferJobsAPI](#).

Voici un exemple de CLI qui appelle `ListMetadataTransferJobs` en AWS IoT TwinMaker tant que `DestinationType` et en `s3` tant que `SourceType` :

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

#### Note

Vous pouvez modifier les valeurs des paramètres `SourceType` et `DestinationType` pour qu'elles correspondent à la source et à la destination de votre tâche d'importation ou d'exportation.

Pour plus d'exemples de commandes CLI qui invoquent ces actions d'API, consultez [AWS IoT TwinMaker metadataTransferJob exemples](#).

Si vous rencontrez des erreurs lors de la tâche de transfert, consultez [Gestion des erreurs](#).

## Gestion des erreurs

Après avoir créé et exécuté une tâche de transfert, vous pouvez appeler `GetMetadataTransferJob` pour diagnostiquer les erreurs survenues :

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

Une fois que l'état de la tâche est passé à COMPLETED zéro, vous pouvez vérifier les résultats de la tâche. GetMetadataTransferJob renvoie un objet appelé [MetadataTransferJobProgress](#) qui contient les champs suivants :

- FailedCount : indique le nombre de ressources qui ont échoué pendant le processus de transfert.
- SkippedCount : indique le nombre de ressources ignorées pendant le processus de transfert.
- SucceededCount : indique le nombre de ressources qui ont réussi pendant le processus de transfert.
- TotalCount : indique le nombre total de ressources impliquées dans le processus de transfert.

En outre, un élément ReportURL contenant une URL pré-signée est renvoyé. Si votre tâche de transfert comporte des erreurs que vous souhaitez approfondir, vous pouvez télécharger un rapport d'erreur complet à l'aide de cette URL.

## Modèles de métadonnées d'importation

Vous pouvez importer de nombreux composants, types de composants ou entités en une seule opération d'importation en bloc. Les exemples présentés dans cette section montrent comment procéder.

template: Importing entities

Utilisez le format de modèle suivant pour une tâche qui importe des entités :

```
{  
  "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

Utilisez le format de modèle suivant pour une tâche qui importe des ComponentTypes :

```

{
  "componentTypes": [
    {
      "componentTypeId": "string",
      "componentTypeName": "string",
      "description": "string",

```

```
"extendsFrom": [
  "string"
],
"functions": {
  "string": {
    "implementedBy": {
      "isNative": "boolean",
      "lambda": {
        "functionName": "Telemetry-tsDataReader",
        "arn": "Telemetry-tsDataReaderARN"
      }
    },
    "requiredProperties": [
      "string"
    ],
    "scope": "string"
  }
},
"isSingleton": "boolean",
"propertyDefinitions": {
  "string": {
    "configuration": {
      "string": "string"
    },
    "dataType": "DataType",
    "defaultValue": "DataValue",
    "displayName": "string",
    "isExternalId": "boolean",
    "isRequiredInEntity": "boolean",
    "isStoredExternally": "boolean",
    "isTimeSeries": "boolean"
  }
},
"propertyGroups": {
  "string": {
    "groupType": "string",
    "propertyNames": [
      "string"
    ]
  }
},
"tags": {
  "string": "string"
}
```

```

    }
  ]
}

```

## template: Importing components

Utilisez le format de modèle suivant pour une tâche qui importe des composants :

```

{
  "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 exemples

Utilisez les commandes suivantes pour gérer vos transferts de métadonnées :

- [CreateMetadataTransferJob](#) Action de l'API.

Exemple de commande CLI :

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

- Pour annuler une tâche, utilisez l'action [CancelMetadataTransferJob](#) API.

Exemple de commande CLI :

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

Lorsque vous appelez `CancelMetadataTransferJob`, cela annule uniquement une tâche de transfert de métadonnées spécifique, et les ressources déjà exportées ou importées ne sont pas affectées.

- Pour récupérer des informations sur une tâche spécifique, utilisez l'action [GetMetadataTransferJob](#) API.

Exemple de commande CLI :

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

```

- Pour répertorier les tâches en cours, utilisez l'action [ListMetadataTransferJobs](#) API.

Vous pouvez filtrer les résultats renvoyés à `ListMetadataTransferJobs` à l'aide d'un fichier JSON. Consultez la procédure suivante à l'aide de la CLI :

1. Créez un fichier JSON d'entrée CLI pour spécifier les filtres que vous souhaitez utiliser :

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

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

Enregistrez-le et enregistrez le nom du fichier, vous en aurez besoin lors de la saisie de la commande CLI.

2. Utilisez le fichier JSON comme argument de la commande CLI suivante :

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

## AWS IoT TwinMaker schéma de tâche de transfert de métadonnées

metadataTransferJob schéma d'importation : utilisez ce schéma de AWS IoT TwinMaker métadonnées pour valider vos données lorsque vous les chargez dans un compartiment Amazon S3 :

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

```

    "minLength": 0,
    "maxLength": 512,
    "default": ""
  },
  "ComponentTypeName": {
    "description": "A friendly name for the component type.",
    "type": "string",
    "pattern": ".*[^\u0000-\\u001F\\u007F]*.*",
    "minLength": 1,
    "maxLength": 256
  },
  "ComponentTypeId": {
    "description": "The ID of the component type.",
    "type": "string",
    "pattern": "[a-zA-Z_\\.\\-0-9:]+",
    "minLength": 1,
    "maxLength": 256
  },
  "ComponentName": {
    "description": "The name of the component.",
    "type": "string",
    "pattern": "[a-zA-Z_\\.\\-0-9:]+",
    "minLength": 1,
    "maxLength": 256
  },
  "EntityId": {
    "description": "The ID of the entity.",
    "type": "string",
    "minLength": 1,
    "maxLength": 128,
    "pattern": "[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}|^[a-zA-Z0-9][a-zA-Z_\\.\\-0-9:]*[a-zA-Z0-9]+"
  },
  "EntityName": {
    "description": "The name of the entity.",
    "type": "string",
    "minLength": 1,
    "maxLength": 256,
    "pattern": "[a-zA-Z_0-9-\\.][a-zA-Z_0-9-\\. ]*[a-zA-Z0-9]+"
  },
  "ParentEntityId": {
    "description": "The ID of the parent entity.",
    "type": "string",
    "minLength": 1,

```

```

    "maxLength": 128,
    "pattern": "\\$ROOT|^[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}|^[a-zA-Z0-9][a-zA-Z_\\-0-9.:]*[a-zA-Z0-9]+",
    "default": "$ROOT"
  },
  "DisplayName": {
    "description": "A friendly name for the property.",
    "type": "string",
    "pattern": ".*[^\u0000-\\u001F\\u007F]*.*",
    "minLength": 0,
    "maxLength": 256
  },
  "Tags": {
    "description": "Metadata that you can use to manage the entity / componentType",
    "patternProperties": {
      "^[a-zA-Z0-9_./+=\\-@]*$": {
        "type": "string",
        "minLength": 1,
        "maxLength": 256
      }
    }
  },
  "existingJavaType": "java.util.Map<String,String>",
  "minProperties": 0,
  "maxProperties": 50
},
"Relationship": {
  "description": "The type of the relationship.",
  "type": "object",
  "properties": {
    "relationshipType": {
      "description": "The type of the relationship.",
      "type": "string",
      "pattern": ".*",
      "minLength": 1,
      "maxLength": 256
    },
    "targetComponentTypeId": {
      "description": "The ID of the target component type associated with this relationship.",
      "$ref": "#/definitions/ComponentTypeId"
    }
  },
  "additionalProperties": false
},

```

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

```

```

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

```

```
    "type": "array",
    "minItems": 0,
    "maxItems": 50,
    "uniqueItems": false,
    "insertionOrder": false,
    "items": {
      "$ref": "#/definitions/DataValue"
    },
    "default": null
  },
  "nestedType": {
    "description": "The nested type in the data type.",
    "$ref": "#/definitions/DataType"
  },
  "relationship": {
    "description": "A relationship that associates a component with another
component.",
    "$ref": "#/definitions/Relationship"
  },
  "type": {
    "description": "The underlying type of the data type.",
    "type": "string",
    "enum": [
      "RELATIONSHIP",
      "STRING",
      "LONG",
      "BOOLEAN",
      "INTEGER",
      "DOUBLE",
      "LIST",
      "MAP"
    ]
  },
  "unitOfMeasure": {
    "description": "The unit of measure used in this data type.",
    "type": "string",
    "pattern": ".*",
    "minLength": 1,
    "maxLength": 256
  }
},
"required": [
  "type"
],
```

```

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

```

```

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

```

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

```

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

```

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

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

```
    "entityId",
    "entityName"
  ],
  "additionalProperties": false
},
"ComponentType": {
  "type": "object",
  "properties": {
    "description": {
      "description": "The description of the component type.",
      "$ref": "#/definitions/DescriptionWithDefault"
    },
    "componentTypeId": {
      "$ref": "#/definitions/ComponentTypeId"
    },
    "componentTypeExternalId": {
      "description": "The external ID of the component type.",
      "$ref": "#/definitions/ExternalId"
    },
    "componentTypeName": {
      "$ref": "#/definitions/ComponentTypeName"
    },
    "extendsFrom": {
      "description": "Specifies the parent component type to extend.",
      "type": "array",
      "minItems": 1,
      "maxItems": 256,
      "uniqueItems": true,
      "insertionOrder": false,
      "items": {
        "$ref": "#/definitions/ComponentTypeId"
      },
      "default": null
    },
    "functions": {
      "description": "a Map of functions in the component type. Each function's key must be unique to this map.",
      "type": "object",
      "patternProperties": {
        "[a-zA-Z_\\-0-9]+": {
          "$ref": "#/definitions/Function"
        }
      },
      "additionalProperties": {
```

```
    "$ref": "#/definitions/Function"
  }
},
"isSingleton": {
  "description": "A Boolean value that specifies whether an entity can have
more than one component of this type.",
  "type": "boolean",
  "default": false
},
"propertyDefinitions": {
  "description": "An map of the property definitions in the component type.
Each property definition's key must be unique to this map.",
  "$ref": "#/definitions/PropertyDefinitions"
},
"propertyGroups": {
  "description": "An object that maps strings to the property groups to set in
the component type. Each string in the mapping must be unique to this object.",
  "$ref": "#/definitions/PropertyGroups"
},
"tags": {
  "$ref": "#/definitions/Tags"
}
},
"required": [
  "componentTypeId"
],
"additionalProperties": false
},
"EntityComponent": {
  "type": "object",
  "properties": {
    "entityId": {
      "$ref": "#/definitions/EntityId"
    },
    "componentName": {
      "$ref": "#/definitions/ComponentName"
    },
    "componentExternalId": {
      "description": "The external ID of the component.",
      "$ref": "#/definitions/ExternalId"
    },
    "componentTypeId": {
      "$ref": "#/definitions/ComponentTypeId"
    }
  },

```

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

```
    },
    "default": null
  }
}
```

Voici un exemple qui crée un nouveau `ComponentType` appelé `component.type.initial` et crée une entité appelée `initial` :

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

Voici un exemple qui met à jour les entités existantes :

```
{
  "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 connecteurs de données

AWS IoT TwinMaker utilise une architecture basée sur des connecteurs qui vous permet de connecter les données de votre propre banque de données à AWS IoT TwinMaker. Cela signifie que vous n'avez pas besoin de migrer les données avant de les utiliser AWS IoT TwinMaker. Actuellement, AWS IoT TwinMaker prend en charge les connecteurs propriétaires pour AWS IoT SiteWise. Si vous stockez des données de modélisation et de propriété dans AWS IoT SiteWise, vous n'avez pas besoin d'implémenter vos propres connecteurs. Si vous stockez vos données de modélisation ou de propriété dans d'autres magasins de données, tels que Timestream, DynamoDB ou Snowflake, vous devez implémenter des AWS Lambda connecteurs avec l'interface du connecteur de données AWS IoT TwinMaker afin de pouvoir appeler votre connecteur si nécessaire.

## Rubriques

- [AWS IoT TwinMaker connecteurs de données](#)
- [AWS IoT TwinMaker Connecteur de données tabulaires Athena](#)
- [Développement de connecteurs AWS IoT TwinMaker de données de séries chronologiques](#)

# AWS IoT TwinMaker connecteurs de données

Les connecteurs doivent accéder à votre banque de données sous-jacente pour résoudre les requêtes envoyées et renvoyer des résultats ou une erreur.

Pour en savoir plus sur les connecteurs disponibles, leurs interfaces de demande et leurs interfaces de réponse, consultez les rubriques suivantes.

Pour plus d'informations sur les propriétés utilisées dans les interfaces de connecteur, consultez l'action [GetPropertyValueHistory](#) API.

### Note

Certains connecteurs comportent deux champs d'horodatage dans les interfaces de demande et de réponse pour les propriétés d'heure de début et de fin. Les deux `startDateTime` et `endDateTime` utilisent un nombre long pour représenter la seconde époque, ce qui n'est plus pris en charge. Pour garantir la rétrocompatibilité, nous envoyons toujours une

valeur d'horodatage à ce champ, mais nous recommandons d'utiliser les endTime champs startTime et conformes au format d'horodatage de notre API.

## Rubriques

- [Connecteur d'initialisation de schéma](#)
- [DataReaderByEntity](#)
- [DataReaderByComponentType](#)
- [DataReader](#)
- [AttributePropertyValueReaderByEntity](#)
- [DataWriter](#)
- [Exemples](#)

## Connecteur d'initialisation de schéma

Vous pouvez utiliser l'initialiseur de schéma dans le type de composant ou le cycle de vie de l'entité pour récupérer le type de composant ou les propriétés du composant à partir de la source de données sous-jacente. L'initialiseur de schéma importe automatiquement le type de composant ou les propriétés du composant sans appeler explicitement une action d'API pour le configurer.

properties

### SchemaInitializer interface de demande

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

**Note**

La carte des propriétés de cette interface de demande est un `PropertyRequest`. Pour de plus amples informations, veuillez consulter [PropertyRequest](#).

## SchemaInitializer interface de réponse

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

**Note**

La carte des propriétés de cette interface de demande est un `PropertyResponse`. Pour de plus amples informations, veuillez consulter [PropertyResponse](#).

## DataReaderByEntity

`DataReaderByEntity` est un connecteur de plan de données utilisé pour obtenir les valeurs chronologiques des propriétés d'un seul composant.

Pour plus d'informations sur les types de propriétés, la syntaxe et le format de ce connecteur, consultez l'action [GetPropertyValueHistoryAPI](#).

## DataReaderByEntity interface de demande

```
{
  "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"
}
```

## DataReaderByEntity interface de réponse

```
{
  "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

Pour obtenir les valeurs chronologiques des propriétés communes issues du même type de composant, utilisez le connecteur DataReaderByEntity de plan de données. Par exemple, si vous définissez des propriétés de séries chronologiques dans le type de composant et que plusieurs composants utilisent ce type de composant, vous pouvez interroger ces propriétés pour tous les composants d'une plage de temps donnée. C'est un cas d'utilisation courant lorsque vous souhaitez interroger l'état d'alarme de plusieurs composants pour obtenir une vue globale de vos entités.

Pour plus d'informations sur les types de propriétés, la syntaxe et le format de ce connecteur, consultez l'action [GetPropertyValueHistory](#) API.

## DataReaderByComponentType interface de demande

```
{
  "startDateTime": long, // In epoch sec, deprecated
  "startTime": "string", // ISO-8601 timestamp format
  "endDateTime": long, // In epoch sec, deprecated
  "endTime": "string", // ISO-8601 timestamp format
  "properties": { // A map of properties as in the get-entity API response
    // property name as key,
    // value is of type PropertyResponse
    "string": "PropertyResponse"
  },
  "workspaceId": "string",
  "selectedProperties": List:"string",
  "propertyFilters": List:PropertyFilter,
  "componentTypeId": "string",
  "interpolation": InterpolationParameters,
  "nextToken": "string",
  "maxResults": int,
  "orderByTime": "string"
}
```

## DataReaderByComponentType interface de réponse

```
{
  "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` est un connecteur de plan de données capable de gérer à la fois le cas de `DataReaderByEntity` et `DataReaderByComponentType`.

Pour plus d'informations sur les types de propriétés, la syntaxe et le format de ce connecteur, consultez l'action [GetPropertyValueHistoryAPI](#).

### DataReader interface de demande

La `EntityId` et l'`componentName` sont facultatifs.

```
{
  "startDateTime": long, // In epoch sec, deprecated
  "startTime": "string", // ISO-8601 timestamp format
  "endDateTime": long, // In epoch sec, deprecated
  "endTime": "string", // ISO-8601 timestamp format
  "properties": { // A map of properties as in the get-entity API response
    // property name as key,
    // value is of type PropertyRequest
    "string": "PropertyRequest"
  },

  "workspaceId": "string",
  "selectedProperties": List:"string",
  "propertyFilters": List:PropertyFilter,
  "entityId": "string",
  "componentName": "string",
  "componentTypeId": "string",
  "interpolation": InterpolationParameters,
  "nextToken": "string",
  "maxResults": int,
  "orderByTime": "string"
}
```

### DataReader interface de réponse

```
{
  "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 est un connecteur de plan de données que vous pouvez utiliser pour récupérer la valeur des propriétés statiques d'une seule entité.

Pour plus d'informations sur les types de propriétés, la syntaxe et le format de ce connecteur, consultez l'action [GetPropertyValueAPI](#).

### AttributePropertyValueReaderByEntity interface de demande

```

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

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

```

### AttributePropertyValueReaderByEntity interface de réponse

```

{
  "propertyValues": {

```

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

## DataWriter

DataWriter est un connecteur de plan de données que vous pouvez utiliser pour réécrire des points de données de séries chronologiques dans le magasin de données sous-jacent pour les propriétés d'un seul composant.

Pour plus d'informations sur les types de propriétés, la syntaxe et le format de ce connecteur, consultez l'action [BatchPutPropertyValuesAPI](#).

### DataWriter interface de demande

```
{
  "workspaceId": "string",
  "properties": {
    // entity id as key
    "String": {
      // property name as key,
      // value is of type PropertyResponse
      "string": PropertyResponse
    }
  },
  "entries": [
    {
      "entryId": "string",
      "entityPropertyReference": EntityPropertyReference, // The same
as EntityPropertyReference
      "propertyValues": [
        {
          "timestamp": long, // Epoch sec, deprecated
          "time": "string", // ISO-8601 timestamp format
          "value": DataValue // The same as DataValue
        }
      ]
    }
  ]
}
```

```
}
```

## DataWriter interface de réponse

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

## Exemples

Les exemples JSON suivants sont des exemples de syntaxe de réponse et de demande pour plusieurs connecteurs.

- SchemaInitializer:

Les exemples suivants montrent l'initialiseur de schéma dans le cycle de vie d'un type de composant.

Requête :

```
{
  "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"
    }
  }
}
}
}

```

### Réponse :

```

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

```

```

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

```

- Initialiseur de schéma dans le cycle de vie des entités :

Requête :

```

{
  "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"
    }
  }
}
}

```

Réponse :

```

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

```

- `DataReaderByEntity` et `DataReader` :

Requête :

```

{
  "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
    }
  }
}
}

```

Réponse :

```

{
  "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:

Requête :

```

{
  "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  
    }  
  }  
}  
}
```

Réponse :

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

- **DataWriter:**

Requête :

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

```

Réponse :

```

{
    "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 Connecteur de données tabulaires Athena

Avec le connecteur de données tabulaires Athena, vous pouvez accéder à vos magasins de données Athena et les utiliser dans AWS IoT TwinMaker. Vous pouvez utiliser vos données Athena pour créer des jumeaux numériques sans avoir à effectuer un effort intensif de migration des données. Vous pouvez utiliser le connecteur prédéfini ou créer un connecteur Athena personnalisé pour accéder aux données de vos sources de données Athena.

## AWS IoT TwinMaker Conditions requises pour le connecteur de données Athena

Avant d'utiliser le connecteur de données tabulaires Athena, remplissez les conditions préalables suivantes :

- Créez des tables Athena gérées et leurs ressources Amazon S3 associées. [Pour plus d'informations sur l'utilisation d'Athena, consultez la documentation d'Athena.](#)

- Créez un AWS IoT TwinMaker espace de travail. Vous pouvez créer un espace de travail dans la [AWS IoT TwinMaker console](#).
- Mettez à jour le rôle IAM de votre espace de travail avec les autorisations Athena. Pour de plus amples informations, veuillez consulter [Modifiez le rôle IAM de votre espace de travail pour utiliser le connecteur de données Athena](#).
- Familiarisez-vous avec AWS IoT TwinMaker le système de composants d'entités et apprenez à créer des entités. Pour de plus amples informations, veuillez consulter [Créez votre première entité](#).
- Familiarisez-vous avec AWS IoT TwinMaker les connecteurs de données. Pour de plus amples informations, veuillez consulter [AWS IoT TwinMaker connecteurs de données](#).

## Utilisation du connecteur de données Athena

Pour utiliser le connecteur de données Athena, vous devez créer un composant en utilisant le connecteur Athena comme type de composant. Vous attachez ensuite le composant à une entité de votre scène pour l'utiliser dans AWS IoT TwinMaker.

### Création d'un type de composant avec le connecteur de données Athena

Utilisez cette procédure pour créer un type de AWS IoT TwinMaker composant avec le connecteur de données tabulaires Athena :

1. Accédez à la [console AWS IoT TwinMaker](#).
2. Ouvrez un espace de travail existant ou [créez-en un nouveau](#).
3. Dans le menu de navigation de gauche, choisissez Types de composants, puis sélectionnez Créer un type de composant pour ouvrir la page de création du type de composant.
4. Sur la page Créer un type de composant, renseignez le champ ID avec un ID correspondant à votre cas d'utilisation.

**Component type information**

ID

com.test.athena.connector.example

Description

Example athena connector child component type

Must be less than 2048 characters

Base Type

Choose a pre-defined Component Type or create your own

com.amazon.athena.connector

5. Choisissez le type de base. Dans la liste déroulante, sélectionnez le connecteur de données tabulaire Athena intitulé `com.amazon.athena.connector`.
6. Configurez la source de données du type de composant en choisissant les ressources Athena dans les champs suivants :
  - Choisissez une source de données Athena.
  - Choisissez une base de données Athena.
  - Choisissez un nom de table.
  - Choisissez un groupe de travail Athena.
7. Une fois que vous avez sélectionné les ressources Athena que vous souhaitez utiliser comme source de données, choisissez les colonnes du tableau que vous souhaitez inclure.
8. Sélectionnez le nom d'une colonne d'ID externe. Sélectionnez une colonne à l'étape précédente pour servir de colonne d'ID externe. L'identifiant externe est l'identifiant utilisé pour représenter un actif Athena et le mapper à une AWS IoT TwinMaker entité.

## 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. (Facultatif) Ajoutez des AWS balises à ces ressources afin de pouvoir les regrouper et les organiser.
10. Choisissez Créer un type de composant pour terminer la création du type de composant.

Créez un composant avec le type de connecteur de données Athena et attachez-le à une entité

Utilisez cette procédure pour créer un AWS IoT TwinMaker composant avec le connecteur de données tabulaires Athena et l'associer à une entité :

 Note

Vous devez disposer d'un type de composant existant qui utilise le connecteur de données tabulaires Athena comme source de données pour effectuer cette procédure. Reportez-vous à la procédure précédente Créez un type de composant avec le connecteur de données Athena avant de commencer cette procédure pas à pas.

1. Accédez à la [console AWS IoT TwinMaker](#).
2. Ouvrez un espace de travail existant ou [créez-en un nouveau](#).
3. Dans le menu de navigation de gauche, choisissez Entités, puis sélectionnez l'entité à laquelle vous souhaitez ajouter le composant ou créer une nouvelle entité.
4. [Créez une nouvelle entité](#).
5. Sélectionnez ensuite Ajouter un composant. , renseignez le champ Nom du composant avec un nom correspondant à votre cas d'utilisation.
6. Dans le menu déroulant Type de composant, sélectionnez l'ID du type de composant que vous avez créé lors de la procédure précédente.
7. Entrez les informations du composant, un nom de composant et sélectionnez l'enfant ComponentType créé précédemment. Il s'agit de ce ComponentType que vous avez créé avec le connecteur de données Athena.
8. Dans la section Propriétés, entrez l'athenaComponentExternalidentifiant du composant.

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

[Add another property](#)

9. Choisissez Ajouter un composant pour terminer la création du composant.

Vous venez de créer avec succès un composant avec le connecteur de données Athena comme type de composant et de l'associer à une entité.

## Utilisation de la référence JSON du connecteur de données tabulaires Athena

L'exemple suivant est la référence JSON complète pour le connecteur de données tabulaires Athena. Utilisez-le comme ressource pour créer des connecteurs de données et des types de composants personnalisés.

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

## Utilisation du connecteur de données Athena

Vous pouvez faire apparaître vos entités qui utilisent des tables Athena dans Grafana. Pour plus d'informations, consultez la section Intégration du tableau de [bord AWS IoT TwinMaker Grafana](#).

Consultez la [documentation Athena](#) pour plus d'informations sur la création et l'utilisation des tables Athena pour stocker des données.

## Résolution des problèmes liés au connecteur de données Athena

Cette rubrique traite des problèmes courants que vous pouvez rencontrer lors de la configuration du connecteur de données Athena.

## Emplacement du groupe de travail Athena :

Lors de la création du type de composant du connecteur Athena, un groupe de travail Athena doit avoir configuré l'emplacement de sortie. Consultez la section [Fonctionnement des groupes de travail](#).

## Autorisations de rôle IAM manquantes :

Il se peut que le AWS IoT TwinMaker rôle d'espace de travail ne dispose pas de l'autorisation d'accès à l'API Athena lors de la création d'un ComponentType, de l'ajout d'un composant Ca à une entité ou de l'exécution de l'API. GetPropertyValue Pour mettre à jour les autorisations IAM, voir [Créer et gérer un rôle de service pour AWS IoT TwinMaker](#).

## Visualisez les données tabulaires d'Athena dans Grafana

Un plugin Grafana est également disponible pour visualiser vos données tabulaires sur Grafana, un panneau de tableau de bord doté de fonctionnalités supplémentaires telles que le tri et le filtrage en fonction des propriétés sélectionnées sans effectuer d'appels d'API AWS IoT TwinMaker ou d'interactions avec Athena. Cette rubrique explique comment configurer Grafana pour visualiser les données tabulaires d'Athena.

### Conditions préalables

Avant de configurer un panneau Grafana pour visualiser les données tabulaires d'Athena, vérifiez les conditions préalables suivantes :

- Vous avez configuré un environnement Grafana. Pour plus d'informations, voir Intégration de [AWS IoT TwinMaker Grafana](#).
- Vous pouvez configurer une source de données Grafana. Pour plus d'informations, voir [Grafana AWS IoT TwinMaker](#).
- Vous êtes habitué à créer un nouveau tableau de bord et à ajouter un nouveau panneau.

## Visualisez les données tabulaires d'Athena dans Grafana

Cette procédure explique comment configurer un panneau Grafana pour visualiser les données tabulaires d'Athena.

1. Ouvrez votre tableau de AWS IoT TwinMaker bord Grafana.

2. Sélectionnez le panneau Table dans les paramètres du panneau.
3. Sélectionnez votre source de données dans la configuration de la requête.
4. Sélectionnez la requête Obtenir la valeur de la propriété.
5. Sélectionnez une entité.
6. Sélectionnez un composant doté d'un ComponentType qui étend le type de composant de base Athena.
7. Sélectionnez le groupe de propriétés de votre table Athena.
8. Sélectionnez autant de propriétés que vous le souhaitez dans le groupe de propriétés.
9. Configurez les conditions tabulaires à l'aide d'une liste de filtres et d'ordres de propriétés. Avec les options suivantes :
  - Filtre : définissez une expression pour une valeur de propriété afin de filtrer vos données.
  - OrderBy: spécifiez si les données doivent être renvoyées par ordre croissant ou décroissant pour une propriété.

The screenshot displays a table of data and a configuration panel for a query. The table, titled "Panel Title", has columns: crit {componentName=, description {component, equipment\_type {compo, status {componentNam, total {componentName=, and won {componentName=. The data rows are as follows:

| 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                    | BYTN-02-XV-06517 do      | VALVE                 | COMPLETED            | 71326                 | 128458              |

Below the table is a configuration panel for a query. The "Query type" is "Get Property value". The "Entity" is "TabularEntity1", the "Component Name" is "TabularComponent", and the "Property Group" is "tabularPropertyGroup (TABULAR)". The "Selected Properties" are: won (INTEGER), status (STRING), total (INTEGER), crit (INTEGER), description (STRING), and equipment\_type (STRING). The "Filter" is set to "crit (INTEGER) = 5". The "OrderBy" is set to "total (INTEGER) DESC".

# Développement de connecteurs AWS IoT TwinMaker de données de séries chronologiques

Cette section explique comment développer un connecteur de données de séries chronologiques dans le step-by-step cadre d'un processus. En outre, nous présentons un exemple de connecteur de données chronologiques basé sur l'ensemble de l'échantillon de Cookie Factory, qui inclut des modèles 3D, des entités, des composants, des alarmes et des connecteurs. La source d'échantillons de cookie factory est disponible dans le [GitHub référentiel AWS IoT TwinMaker d'échantillons](#).

## Rubriques

- [AWS IoT TwinMaker prérequis pour les connecteurs de données chronologiques](#)
- [Contexte du connecteur de données chronologiques](#)
- [Développement d'un connecteur de données chronologiques](#)
- [Améliorer votre connecteur de données](#)
- [Test de votre connecteur](#)
- [Sécurité](#)
- [Création de AWS IoT TwinMaker ressources](#)
- [Quelle est la prochaine étape](#)
- [AWS IoT TwinMaker exemple de connecteur de séries chronologiques de Cookie Factory](#)

## AWS IoT TwinMaker prérequis pour les connecteurs de données chronologiques

Avant de développer votre connecteur de données chronologiques, nous vous recommandons d'effectuer les tâches suivantes :

- Créez un [AWS IoT TwinMaker espace de travail](#).
- Créez des [types de AWS IoT TwinMaker composants](#).
- Créez [AWS IoT TwinMaker des entités](#).
- (Facultatif) Lisez [Utilisation et création de types de composants](#).
- (Facultatif) Lisez [l'interface AWS IoT TwinMaker des connecteurs de données](#) pour avoir une compréhension générale des connecteurs de AWS IoT TwinMaker données.

**Note**

Pour un exemple de connecteur entièrement implémenté, consultez notre exemple d'implémentation d'une fabrique de cookies.

## Contexte du connecteur de données chronologiques

Imaginez que vous travaillez dans une usine qui possède un ensemble de mélangeurs à biscuits et un réservoir d'eau. Vous souhaiteriez créer des jumeaux AWS IoT TwinMaker numériques de ces entités physiques afin de pouvoir surveiller leur état opérationnel en vérifiant divers indicateurs de séries chronologiques.

Vous avez installé des capteurs sur site et vous diffusez déjà les données de mesure dans une base de données Timestream. Vous voulez être en mesure de visualiser et d'organiser les données de mesure AWS IoT TwinMaker avec un minimum de frais. Vous pouvez accomplir cette tâche à l'aide d'un connecteur de données chronologiques. L'image suivante montre un exemple de table de télémétrie, qui est renseignée à l'aide d'un connecteur de séries chronologiques.

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

< 1 2 3 4 5 6 7 ... 100 > 🔍

| TelemetryAssetId                              | TelemetryAssetType | measure_name | time                          | measure_value:varchar | measure_value:double |
|---|--------------------|--------------|-------------------------------|-----------------------|----------------------|
| Mixer_22_680b5b8e-1afe-4a77-87ab-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_7f0b75b-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_7f0b75b-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_2d_8e76844-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      |

[Les ensembles de données et la table Timestream utilisés dans cette capture d'écran sont disponibles dans le AWS IoT TwinMaker référentiel d'exemples. GitHub](#) Consultez également [l'exemple de connecteur cookie factory](#) pour l'implémentation, qui produit le résultat indiqué dans la capture d'écran précédente.

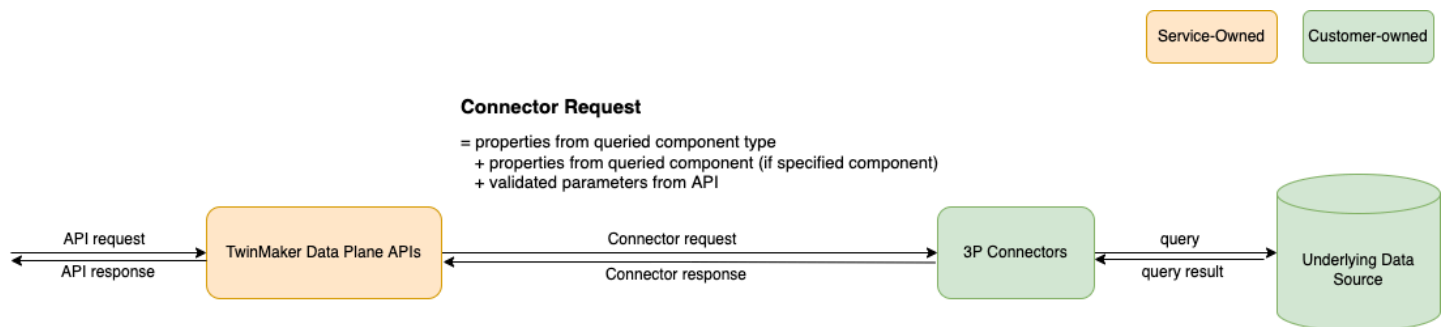
## Flux de données du connecteur de données chronologique

Pour les requêtes de plan de données AWS IoT TwinMaker, extrait les propriétés correspondantes des composants et des types de composants à partir des définitions des composants et des types de composants. AWS IoT TwinMaker transmet les propriétés aux AWS Lambda fonctions ainsi que tous les paramètres de requête d'API contenus dans la requête.

AWS IoT TwinMaker utilise les fonctions Lambda pour accéder aux requêtes provenant de sources de données, les résoudre et renvoyer les résultats de ces requêtes. Les fonctions Lambda utilisent les propriétés du composant et du type de composant du plan de données pour résoudre la demande initiale.

Les résultats de la requête Lambda sont mappés à une réponse d'API et vous sont renvoyés.

AWS IoT TwinMaker définit l'interface du connecteur de données et l'utilise pour interagir avec les fonctions Lambda. À l'aide de connecteurs de données, vous pouvez interroger votre source de données depuis AWS IoT TwinMaker l'API sans aucun effort de migration de données. L'image suivante décrit le flux de données de base décrit dans les paragraphes précédents.



## Développement d'un connecteur de données chronologiques

La procédure suivante décrit un modèle de développement qui se développe progressivement pour devenir un connecteur de données de séries chronologiques fonctionnel. La procédure de base est la suivante :

### 1. Création d'un type de composant de base valide

Dans un type de composant, vous définissez des propriétés communes partagées entre vos composants. Pour en savoir plus sur la définition des types de composants, consultez la section [Utilisation et création de types de composants](#).

AWS IoT TwinMaker utilise un [modèle de modélisation entité-composant](#) afin que chaque composant soit attaché à une entité. Nous vous recommandons de modéliser chaque élément physique en tant qu'entité et de modéliser différentes sources de données avec leurs propres types de composants.

L'exemple suivant montre un type de composant de modèle Timestream doté d'une propriété :

```
{"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
  }
}
}
```

Les éléments clés du type de composant sont les suivants :

- La `telemetryId` propriété identifie la clé unique de l'élément physique dans la source de données correspondante. Le connecteur de données utilise cette propriété comme condition de filtre pour n'interroger que les valeurs associées à l'élément donné. En outre, si vous incluez la valeur de la `telemetryId` propriété dans la réponse de l'API du plan de données, le côté client prend l'ID et peut effectuer une recherche inversée si nécessaire.

- Le `lambdaArn` champ identifie la fonction Lambda avec laquelle le type de composant s'engage.
- Le `isRequiredInEntity` drapeau impose la création de l'identifiant. Cet indicateur est obligatoire pour que, lors de la création du composant, l'ID de l'élément soit également instancié.
- Le `TelemetryId` est ajouté au type de composant en tant qu'identifiant externe afin que l'élément puisse être identifié dans le tableau Timestream.

## 2. Création d'un composant avec le type de composant

Pour utiliser le type de composant que vous avez créé, vous devez créer un composant et l'associer à l'entité à partir de laquelle vous souhaitez récupérer les données. Les étapes suivantes décrivent le processus de création de ce composant :

- a. Accédez à la [console AWS IoT TwinMaker](#).
- b. Sélectionnez et ouvrez le même espace de travail dans lequel vous avez créé les types de composants.
- c. Accédez à la page de l'entité.
- d. Créez une nouvelle entité ou sélectionnez une entité existante dans le tableau.
- e. Une fois que vous avez sélectionné l'entité que vous souhaitez utiliser, choisissez Ajouter un composant pour ouvrir la page Ajouter un composant.
- f. Donnez un nom au composant et pour le type, choisissez le type de composant que vous avez créé avec le modèle en 1. Créez un type de composant de base valide.

## 3. Faites en sorte que votre type de composant appelle un connecteur Lambda

Le connecteur Lambda doit accéder à la source de données, générer l'instruction de requête en fonction de l'entrée et la transmettre à la source de données. L'exemple suivant montre un modèle de demande JSON qui effectue cette opération.

```
{
  "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
    }
  }
}
```

```
}
```

Les principaux éléments de la demande :

- `selectedProperties` Il s'agit d'une liste que vous renseignez avec les propriétés pour lesquelles vous souhaitez obtenir des mesures Timestream.
- Les champs `startTime`, `endTime`, `startDate`, `endDate`, `startTime`, `endTime`, et spécifient une plage de temps pour la demande. Cela détermine la plage d'échantillonnage pour les mesures renvoyées.
- `entityId` Il s'agit du nom de l'entité à partir de laquelle vous demandez des données.
- `componentName` Il s'agit du nom du composant à partir duquel vous interrogez les données.
- Utilisez le `orderByTime` champ pour organiser l'ordre dans lequel les résultats sont affichés.

Dans l'exemple de demande précédent, nous nous attendrions à obtenir une série d'échantillons pour les propriétés sélectionnées pendant la fenêtre temporelle donnée pour l'élément donné, avec l'ordre temporel sélectionné. La déclaration de réponse peut être résumée comme suit :

```
{
  "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. Mettez à jour votre type de composant pour qu'il ait deux propriétés

Le modèle JSON suivant montre un type de composant valide avec deux propriétés :

```

{
  "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. Mettez à jour le connecteur Lambda pour gérer la deuxième propriété

L'API du plan de AWS IoT TwinMaker données prend en charge l'interrogation de plusieurs propriétés en une seule demande et AWS IoT TwinMaker suit une seule demande adressée à un connecteur en fournissant une liste `deselectedProperties`.

La requête JSON suivante montre un modèle modifié qui prend désormais en charge une demande pour deux propriétés.

```

{
  "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
    }
}
```

```
    }  
  }  
}
```

De même, la réponse correspondante est également mise à jour, comme indiqué dans l'exemple suivant :

```
{  
  "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

En termes de pagination dans ce cas, le format de page indiqué dans la demande s'applique à toutes les propriétés. Cela signifie qu'avec cinq propriétés dans la requête et une taille de page de 100, s'il y a suffisamment de points de données dans la source, vous devriez vous attendre à voir 100 points de données par propriété, soit 500 points de données au total.

Pour un exemple d'implémentation, voir [Exemple de connecteur Snowflake activé](#). GitHub

## Améliorer votre connecteur de données

## Gestion des exceptions

Le connecteur Lambda peut lancer des exceptions en toute sécurité. Dans l'appel d'API du plan de données, le AWS IoT TwinMaker service attend que la fonction Lambda renvoie une réponse. Si l'implémentation du connecteur génère une exception, le type d'exception est AWS IoT TwinMaker traduit par : le client API est ainsi informé qu'un problème s'est produit dans le connecteur.

`ConnectorFailure`

## Gestion de la pagination

Dans l'exemple, Timestream fournit une [fonction utilitaire](#) qui peut aider à prendre en charge la pagination de manière native. Cependant, pour certaines autres interfaces de requête, telles que SQL, la mise en œuvre d'un algorithme de pagination efficace peut nécessiter des efforts supplémentaires. Il existe un exemple de connecteur [Snowflake](#) qui gère la pagination dans une interface SQL.

Lorsque le nouveau jeton est renvoyé AWS IoT TwinMaker via l'interface de réponse du connecteur, le jeton est crypté avant d'être renvoyé au client API. Lorsque le jeton est inclus dans une autre demande, il le AWS IoT TwinMaker déchiffre avant de le transmettre au connecteur Lambda. Nous vous recommandons d'éviter d'ajouter des informations sensibles au jeton.

## Test de votre connecteur

Bien que vous puissiez toujours mettre à jour l'implémentation après avoir lié le connecteur au type de composant, nous vous recommandons vivement de vérifier le connecteur Lambda avant de l'intégrer. AWS IoT TwinMaker

Il existe plusieurs manières de tester votre connecteur Lambda : vous pouvez tester le connecteur Lambda dans la console Lambda ou localement dans le. AWS CDK

[Pour plus d'informations sur le test de vos fonctions Lambda, consultez Tester les fonctions Lambda et Tester les applications localement. AWS CDK](#)

## Sécurité

Pour obtenir de la documentation sur les meilleures pratiques en matière de sécurité avec Timestream, consultez la section [Sécurité dans](#) Timestream.

Pour un exemple de prévention des injections SQL, consultez le [script Python](#) suivant dans AWS IoT TwinMaker Samples GitHub Repository.

## Création de AWS IoT TwinMaker ressources

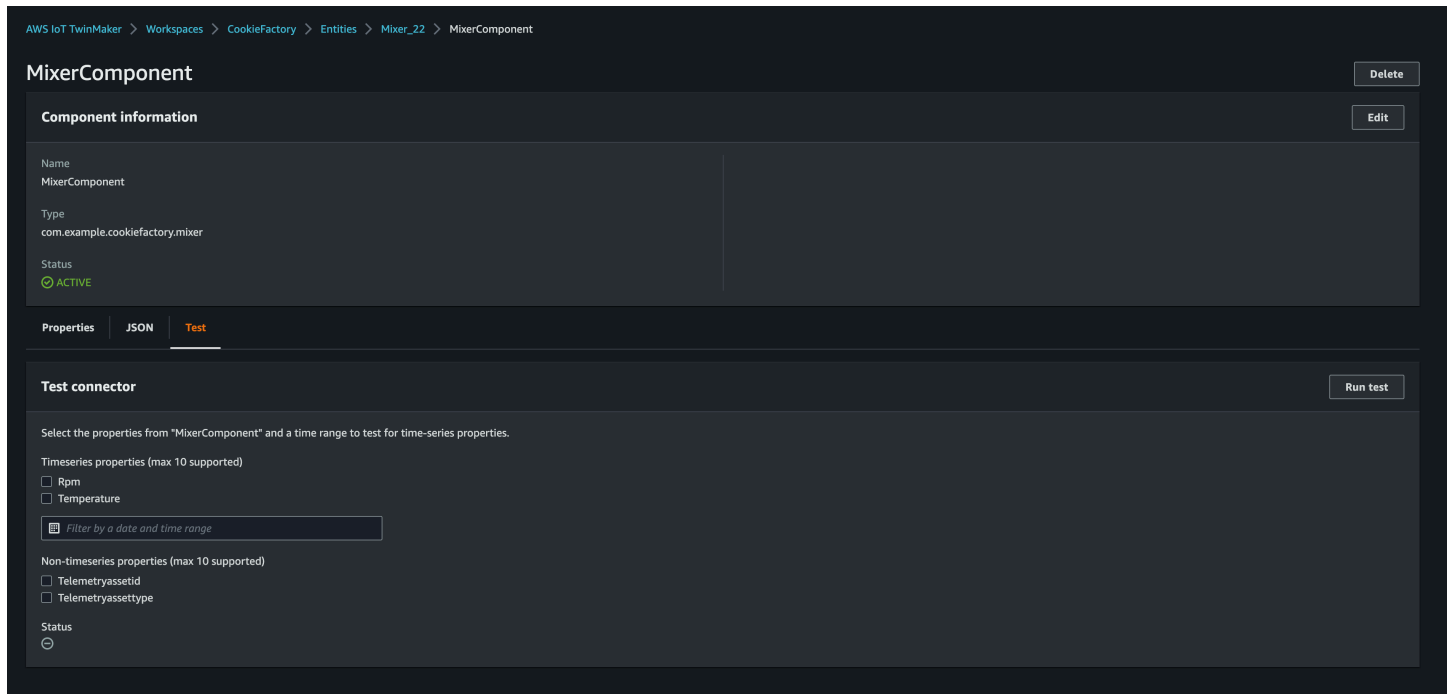
Une fois que vous avez implémenté la fonction Lambda, vous pouvez créer des AWS IoT TwinMaker ressources telles que des types de composants, des entités et des composants via la [AWS IoT TwinMaker console](#) ou l'[API](#).

### Note

Si vous suivez les instructions de configuration de l' [GitHub exemple](#), toutes les AWS IoT TwinMaker ressources sont disponibles automatiquement. Vous pouvez vérifier les définitions des types de composants dans l'[AWS IoT TwinMaker GitHub exemple](#). Une fois que le type de composant est utilisé par un composant, les définitions de propriétés et les fonctions du type de composant ne peuvent pas être mises à jour.

## Tests d'intégration

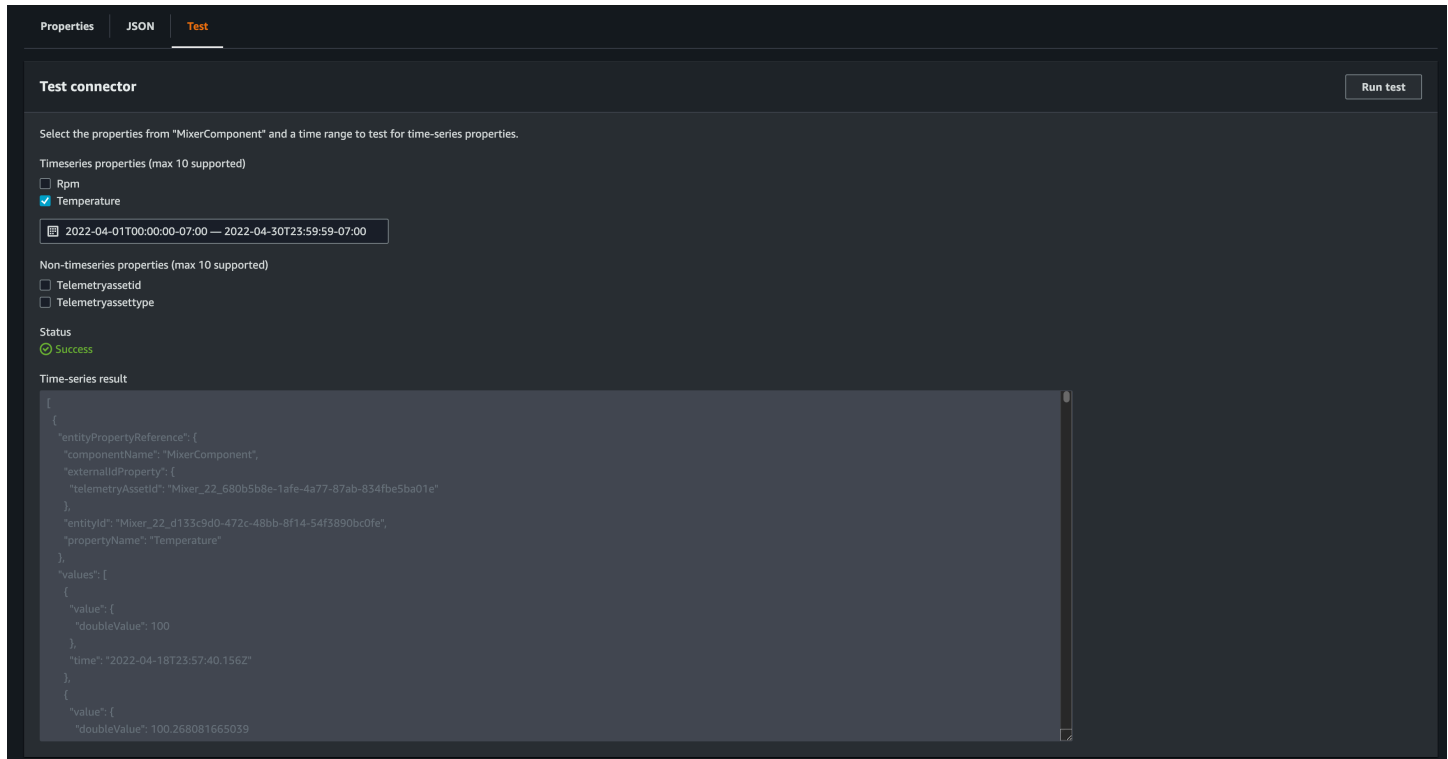
Nous vous recommandons d'effectuer un test intégré AWS IoT TwinMaker pour vérifier que la requête du plan de données fonctionne end-to-end. Vous pouvez le faire via [GetPropertyValueHistory](#) l'API ou facilement dans [AWS IoT TwinMaker la console](#).



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

Dans la AWS IoT TwinMaker console, accédez aux détails du composant, puis sous Test, vous verrez que toutes les propriétés du composant y sont répertoriées. La zone Test de la console

vous permet de tester les propriétés des séries chronologiques ainsi que les non-time-series propriétés. Pour les propriétés des séries chronologiques, vous pouvez également utiliser l' [GetPropertyValueHistory](#) API et pour les non-time-series propriétés, utilisez l' [GetPropertyValue](#) API. Si votre connecteur Lambda prend en charge les requêtes de propriétés multiples, vous pouvez choisir plusieurs propriétés.



```
[{"entityPropertyReference": {"componentName": "MixerComponent", "externalIdProperty": {"telemetryAssetId": "Mixer_22_680b5b8e-1afe-4a77-87ab-834f8e5ba01e"}, "entityId": "Mixer_22_d133c9d0-472c-48bb-8f14-54f3890bc0fe", "propertyName": "Temperature"}, "values": [{"value": {"doubleValue": 100}, "time": "2022-04-18T23:57:40.156Z"}, {"value": {"doubleValue": 100.268081665039}}
```

## Quelle est la prochaine étape

Vous pouvez désormais configurer un tableau de [bord AWS IoT TwinMaker Grafana](#) pour visualiser les métriques. Vous pouvez également explorer d'autres exemples de connecteurs de données dans le [GitHub référentiel d'AWS IoT TwinMaker échantillons](#) pour voir s'ils correspondent à votre cas d'utilisation.

## AWS IoT TwinMaker exemple de connecteur de séries chronologiques de Cookie Factory

Le [code complet de la fonction Lambda de la fabrique de cookies](#) est disponible sur GitHub. Bien que vous puissiez toujours mettre à jour l'implémentation après avoir lié le connecteur au type de composant, nous vous recommandons vivement de vérifier le connecteur Lambda avant de l'intégrer. AWS IoT TwinMaker Vous pouvez tester votre fonction Lambda dans la console Lambda

ou localement dans le. [AWS CDK](#)[Pour plus d'informations sur le test de vos fonctions Lambda, consultez Tester les fonctions Lambda et Tester les applications localement. AWS CDK](#)

## Exemples de types de composants de Cookie Factory

Dans un type de composant, nous définissons des propriétés communes partagées entre les composants. Dans l'exemple de la fabrique à cookies, les composants physiques du même type partagent les mêmes mesures. Nous pouvons donc définir le schéma des mesures dans le type de composant. À titre d'exemple, le type de mixeur est défini dans l'exemple suivant.

```
{
  "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
    }
  }
}
```

Par exemple, un composant physique peut avoir des mesures dans une base de données Timestream, des enregistrements de maintenance dans une base de données SQL ou des données d'alarme dans des systèmes d'alarme. La création de plusieurs composants et leur association à une entité relie différentes sources de données à l'entité et renseigne le graphe entité-composant. Dans ce contexte, chaque composant a besoin d'une `telemetryId` propriété pour identifier la clé unique du composant dans la source de données correspondante. La spécification de la `telemetryId` propriété présente deux avantages : la propriété peut être utilisée dans le connecteur de données comme condition de filtre pour interroger uniquement les valeurs du composant donné et, si vous incluez la valeur de la `telemetryId` propriété dans la réponse de l'API du plan de données, le côté client prend l'ID et peut effectuer une recherche inversée si nécessaire.

Si vous ajoutez le `TelemetryId` au type de composant en tant qu'identifiant externe, il identifie le composant dans le `TimeStream` tableau.

```
{
  "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
    }
  }
}
```

De même, nous avons le type de composant pour le `WaterTank`, comme indiqué dans l'exemple JSON suivant.

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

```
"flowrate2": {
  "dataType": { "type": "DOUBLE" },
  "isTimeSeries": true,
  "isRequiredInEntity": false,
  "isExternalId": false,
  "isStoredExternally": true
},
"tankVolume1": {
  "dataType": { "type": "DOUBLE" },
  "isTimeSeries": true,
  "isRequiredInEntity": false,
  "isExternalId": false,
  "isStoredExternally": true
},
"tankVolume2": {
  "dataType": { "type": "DOUBLE" },
  "isTimeSeries": true,
  "isRequiredInEntity": false,
  "isExternalId": false,
  "isStoredExternally": true
},
"telemetryId": {
  "dataType": { "type": "STRING" },
  "isTimeSeries": false,
  "isRequiredInEntity": true,
  "isExternalId": true,
  "isStoredExternally": false
}
}
}
```

TelemetryType s'agit d'une propriété facultative dans le type de composant si elle vise à interroger les valeurs des propriétés dans le périmètre de l'entité. Pour un exemple, consultez les types de composants définis dans le [GitHub référentiel AWS IoT TwinMaker d'échantillons](#). Certains types d'alarmes sont également intégrés dans la même table. Ils sont donc définis et vous pouvez extraire des propriétés communes, telles que le TelemetryId et TelemetryType vers un type de composant parent pour que d'autres types d'enfants puissent les partager. TelemetryType

## Exemple Lambda

Le connecteur Lambda doit accéder à la source de données, générer l'instruction de requête en fonction de l'entrée et la transmettre à la source de données. Un exemple de demande envoyée au Lambda est illustré dans l'exemple JSON suivant.

```
{
  'workspaceId': 'CookieFactory',
  'selectedProperties': ['Temperature'],
  'startDateTime': 1648796400,
  'startTime': '2022-04-01T07:00:00.000Z',
  'endDateTime': 1650610799,
  'endTime': '2022-04-22T06:59:59.000Z',
  'properties': {
    'telemetryId': {
      'definition': {
        'dataType': { 'type': 'STRING' },
        'isTimeSeries': False,
        'isRequiredInEntity': True,
        'isExternalId': True,
        'isStoredExternally': False,
        'isImported': False,
        'isFinal': False,
        'isInherited': True,
      },
      'value': {
        'stringValue': 'Mixer_22_680b5b8e-1afe-4a77-87ab-834fbe5ba01e'
      }
    }
    'Temperature': {
      'definition': {
        'dataType': { 'type': 'DOUBLE' },
        'isTimeSeries': True,
        'isRequiredInEntity': False,
        'isExternalId': False,
        'isStoredExternally': True,
        'isImported': False,
        'isFinal': False,
        'isInherited': False
      }
    }
    'RPM': {
      'definition': {
```

```

        'dataType': { 'type': 'DOUBLE' },
        'isTimeSeries': True,
        'isRequiredInEntity': False,
        'isExternalId': False,
        'isStoredExternally': True,
        'isImported': False,
        'isFinal': False,
        'isInherited': False
    }
},
'entityId': 'Mixer_22_d133c9d0-472c-48bb-8f14-54f3890bc0fe',
'componentName': 'MixerComponent',
'maxResults': 100,
'orderByTime': 'ASCENDING'
}

```

L'objectif de la fonction Lambda est d'interroger les données de mesure historiques pour une entité donnée. AWS IoT TwinMaker fournit une carte des propriétés des composants, et vous devez spécifier une valeur instanciée pour l'ID du composant. Par exemple, pour traiter la requête au niveau du type de composant (ce qui est courant dans les cas d'utilisation des alarmes) et renvoyer l'état d'alarme de tous les composants de l'espace de travail, la carte des propriétés contient des définitions des propriétés des types de composants.

Dans le cas le plus simple, comme dans la demande précédente, nous voulons une série d'échantillons de température pendant la fenêtre temporelle donnée pour le composant donné, par ordre chronologique croissant. L'instruction de requête peut être résumée comme suit :

```

...
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}
...

```

# Création et édition de AWS IoT TwinMaker scènes

Les scènes sont des visualisations tridimensionnelles de votre jumeau numérique. Ils constituent le principal moyen pour vous de modifier votre jumeau numérique. Découvrez comment ajouter des alarmes, des données de séries chronologiques, des superpositions de couleurs, des balises et des règles visuelles à votre scène afin d'aligner vos visualisations jumelles numériques sur votre cas d'utilisation réel.

Cette section couvre les rubriques suivantes :

- [Avant de créer votre première scène](#)
- [Télécharger des ressources dans la bibliothèque AWS IoT TwinMaker de ressources](#)
- [Créez vos scènes](#)
- [Ajouter des caméras fixes aux entités](#)
- [Montage de scène amélioré](#)
- [Modifiez vos scènes](#)
- [Format du modèle 3D Tiles](#)
- [Scènes dynamiques](#)

## Avant de créer votre première scène

Les scènes s'appuient sur des ressources pour représenter votre jumeau numérique. Ces ressources sont constituées de modèles 3D, de données ou de fichiers de texture. La taille et la complexité de vos ressources, des éléments de la scène tels que l'éclairage et de votre matériel informatique ont un impact sur les performances des AWS IoT TwinMaker scènes. Utilisez les informations de cette rubrique pour réduire le décalage, les temps de chargement et améliorer la fréquence d'images de vos scènes.

## Optimisez vos ressources avant de les importer dans AWS IoT TwinMaker

Vous pouvez l'utiliser AWS IoT TwinMaker pour interagir avec votre jumeau numérique en temps réel. Pour une expérience optimale avec vos scènes, nous vous recommandons d'optimiser vos ressources pour une utilisation dans un environnement en temps réel.

Vos modèles 3D peuvent avoir un impact significatif sur les performances. La géométrie et les maillages complexes des modèles peuvent réduire les performances. Par exemple, les modèles

de CAO industriels présentent un niveau de détail élevé. Nous recommandons de compresser les maillages de ces modèles et de réduire leur nombre de polygones avant de les utiliser dans AWS IoT TwinMaker des scènes. Si vous créez de nouveaux modèles 3D pour AWS IoT TwinMaker, vous devez établir un niveau de détail et le maintenir dans tous vos modèles. Supprimez les détails des modèles qui n'affectent pas la visualisation ou l'interprétation de votre cas d'utilisation.

Pour compresser les modèles et réduire la taille des fichiers, utilisez des outils de compression de maillage open source, tels que la [compression de données 3D DRACO](#).

Des textures non optimisées peuvent également avoir un impact sur les performances. Si vous n'avez pas besoin de transparence dans vos textures, pensez à choisir le format d'image PEG plutôt que le format PNG. Vous pouvez compresser vos fichiers de texture à l'aide d'outils de compression de texture open source, tels que [la compression de texture Basis Universal](#).

## Les meilleures pratiques en matière de performance dans AWS IoT TwinMaker

Pour de meilleures performances avec AWS IoT TwinMaker, tenez compte des limites et des meilleures pratiques suivantes.

- AWS IoT TwinMaker les performances de rendu de scène dépendent du matériel. Les performances varient en fonction de la configuration matérielle de l'ordinateur.
- Nous recommandons un nombre total de polygones inférieur à 1 million pour tous les objets de votre AWS IoT TwinMaker.
- Nous recommandons un total de 200 objets par scène. L'augmentation du nombre d'objets d'une scène au-delà de 200 peut réduire la fréquence d'images de votre scène.
- Nous recommandons que la taille totale de toutes les ressources 3D uniques de votre scène ne dépasse pas 100 mégaoctets. Dans le cas contraire, vous risquez de rencontrer des temps de chargement lents ou une dégradation des performances en fonction de votre navigateur et de votre matériel.
- Les scènes disposent d'un éclairage d'ambiance par défaut. Vous pouvez ajouter des lumières supplémentaires à une scène pour mettre en valeur certains objets ou projeter des ombres sur des objets. Nous recommandons d'utiliser une lampe par scène. Utilisez des lumières là où c'est nécessaire et évitez de reproduire les lumières du monde réel dans une scène.

## En savoir plus

Utilisez ces ressources pour en savoir plus sur les techniques d'optimisation que vous pouvez utiliser pour améliorer les performances de vos scènes.

- [Comment convertir et compresser des modèles OBJ en GLTF pour les utiliser avec AWS IoT TwinMaker](#)
- [Optimisez vos modèles 3D pour le contenu Web](#)
- [Optimisation des scènes pour de meilleures performances WebGL](#)

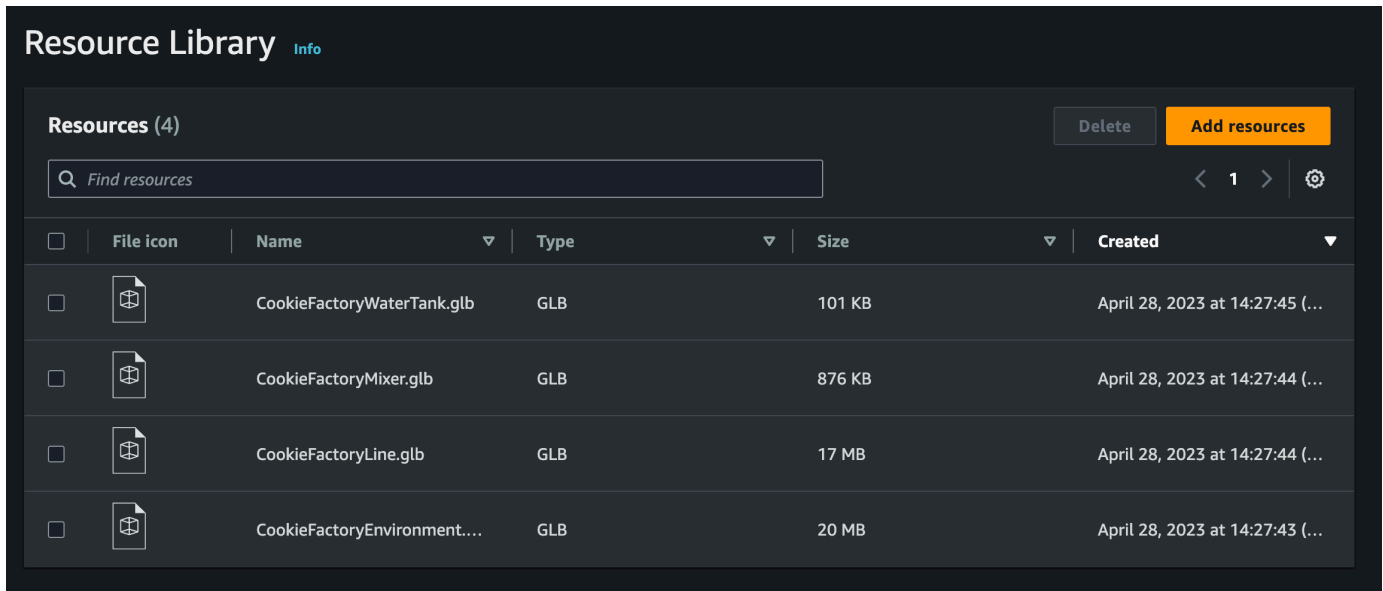
## Télécharger des ressources dans la bibliothèque AWS IoT TwinMaker de ressources

Vous pouvez utiliser la bibliothèque de ressources pour contrôler et gérer toutes les ressources que vous souhaitez placer dans des scènes pour votre application de jumelage numérique. Pour AWS IoT TwinMaker connaître les ressources, téléchargez-les à l'aide de la page de console de la bibliothèque de ressources.

### Téléchargez des fichiers dans la bibliothèque de ressources à l'aide de la console

Procédez comme suit pour ajouter des fichiers à la bibliothèque de ressources à l'aide de la AWS IoT TwinMaker console.

1. Dans le menu de navigation de gauche, sous Espaces de travail, sélectionnez Bibliothèque de ressources.
2. Sélectionnez Ajouter des ressources et choisissez les fichiers que vous souhaitez télécharger.



## Créez vos scènes

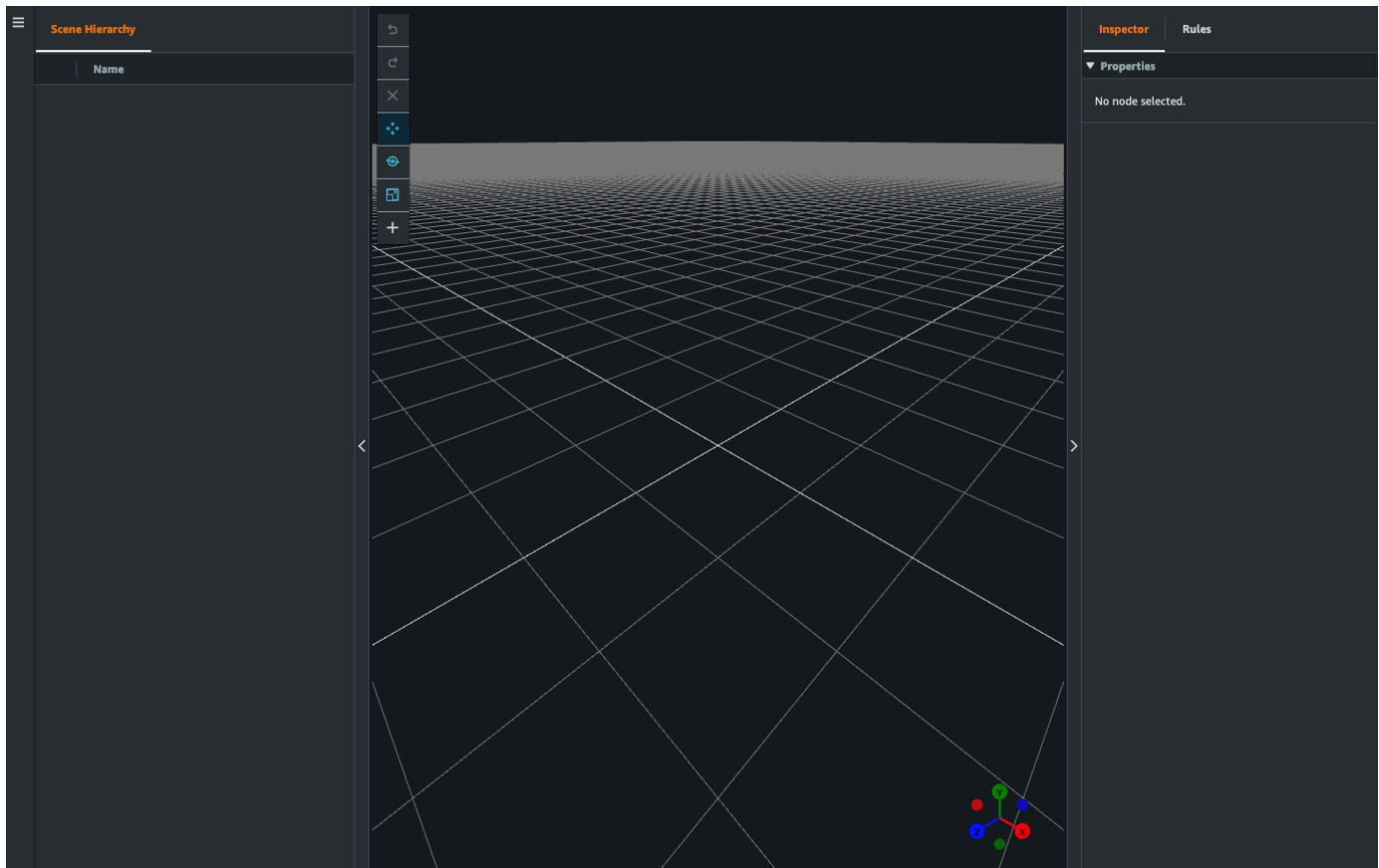
Dans cette section, vous allez configurer une scène afin de pouvoir modifier votre jumeau numérique. Vous pouvez importer un modèle 3D qui a été chargé dans la [bibliothèque de ressources](#), puis ajouter des widgets et lier des données de propriété à des objets pour compléter votre jumeau numérique. Les objets de scène peuvent inclure un bâtiment ou un espace entier, ou des pièces d'équipement individuelles positionnées à leur emplacement physique.

### Note

Avant de créer une scène, vous devez créer un espace de travail.

Utilisez la procédure suivante pour créer votre scène dans AWS IoT TwinMaker.

1. Pour ouvrir le volet des scènes, dans le menu de navigation de gauche de votre espace de travail, sélectionnez Scènes.
2. Choisissez Create scene (Créer la scène). Le volet de création d'une nouvelle scène s'ouvre.
3. Dans le volet de création de scène, entrez le nom et la description de votre nouvelle scène. Si vous avez un plan tarifaire standard ou échelonné, vous pouvez sélectionner votre type de scène. Il est recommandé d'utiliser une [scène dynamique](#).
4. Lorsque vous êtes prêt à créer la scène, choisissez Create scene. La nouvelle scène s'ouvre et est prête à être utilisée.

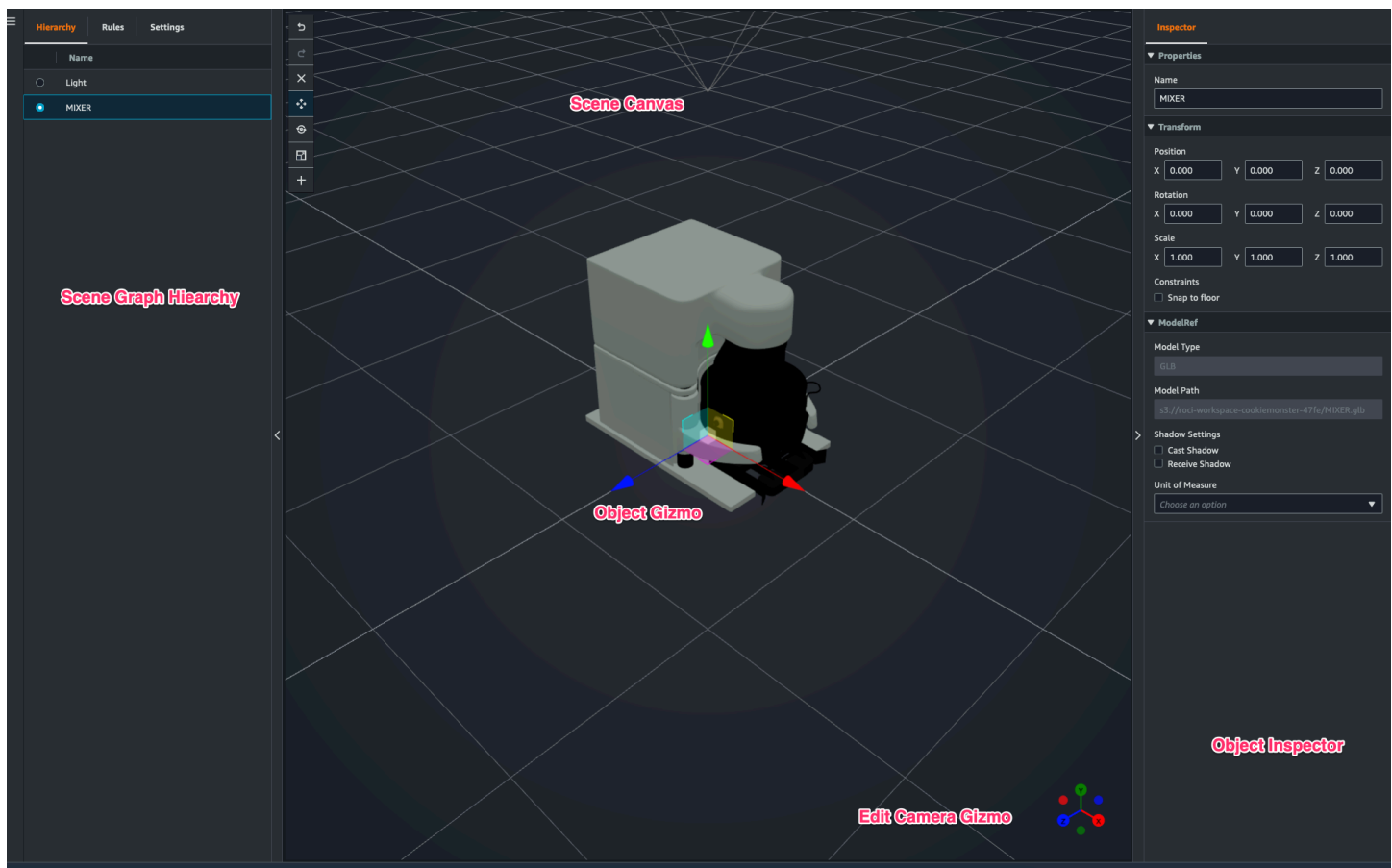


## Utilisez la navigation 3D AWS IoT TwinMaker dans vos scènes

La AWS IoT TwinMaker scène dispose d'un ensemble de commandes de navigation que vous pouvez utiliser pour naviguer efficacement dans l'espace 3D de votre scène. Pour interagir avec l'espace 3D et les objets représentés par votre scène, utilisez les widgets et options de menu suivants.

- **Inspecteur** : utilisez la fenêtre Inspector pour afficher et modifier les propriétés et les paramètres d'une entité ou d'un composant sélectionné dans votre hiérarchie.
- **Canevas de scène** : Le canevas de scène est l'espace 3D dans lequel vous pouvez positionner et orienter les ressources 3D que vous souhaitez utiliser.
- **Hiérarchie des graphes de scène** : vous pouvez utiliser ce panneau pour voir toutes les entités présentes dans votre scène. Il apparaît sur le côté gauche de la fenêtre.
- **Gadget d'objet** : utilisez ce gadget pour déplacer des objets sur le canevas. Il apparaît au centre d'un objet 3D sélectionné dans le canevas de scène.

- Modifier le gadget de caméra : utilisez le gadget d'édition de caméra pour visualiser rapidement l'orientation actuelle de la caméra de visualisation de scène et modifier l'angle de vue. Vous pouvez trouver ce gadget dans le coin inférieur droit de la vue de scène.
- Commandes de zoom : pour naviguer sur le canevas de scène, cliquez avec le bouton droit de la souris et faites glisser le pointeur dans la direction que vous souhaitez déplacer. Pour faire pivoter, cliquez avec le bouton gauche de la souris et faites glisser pour faire pivoter. Pour zoomer, utilisez la molette de défilement de votre souris ou pincez et écartez vos doigts sur le pavé tactile de votre ordinateur portable.



Les boutons de scène du volet hiérarchique comportent les fonctions suivantes répertoriées, dans l'ordre de disposition des boutons :

- Annuler : annule la dernière modification apportée à la scène.
- Rétablir : refaites votre dernière modification dans la scène.
- Plus (+) : utilisez ce bouton pour accéder aux actions suivantes : Ajouter un nœud vide, ajouter un modèle 3D, ajouter une balise, ajouter de la lumière et ajouter un shader de modèle.

- Modifier le mode de navigation : accédez aux options de navigation de la caméra de scène, Orbit et Pan.
- Corbeille (supprimer) : utilisez ce bouton pour supprimer un objet sélectionné dans votre scène.
- Outils de manipulation d'objets : utilisez ce bouton pour déplacer, faire pivoter et redimensionner l'objet sélectionné.

## Ajouter des caméras fixes aux entités

Vous pouvez associer des vues de caméra fixes à vos entités au sein de vos AWS IoT TwinMaker scènes. Ces caméras fournissent une perspective fixe sur un modèle 3D, ce qui vous permet de déplacer rapidement et facilement votre perspective dans une scène vers une entité ciblée.

1. Accédez à votre scène dans la [AWS IoT TwinMaker console](#).
2. Dans le menu de hiérarchie des scènes, sélectionnez l'entité à laquelle vous souhaitez associer la caméra.
3. Appuyez sur le bouton +, puis dans les options du menu déroulant, sélectionnez Ajouter une caméra à partir de la vue actuelle. Pour appliquer une caméra avec la perspective actuelle à l'entité.
4. Dans l'inspecteur, vous pouvez configurer votre caméra et régler les paramètres suivants :
  - Un nom de caméra
  - La position et la rotation de la caméra
  - Distance focale de la caméra
  - Le niveau de zoom
  - Avions découpés de près ou de loin
5. Pour accéder à votre caméra après l'avoir placée. Sélectionnez l'entité à laquelle vous avez ajouté la caméra dans la hiérarchie. Recherchez le nom de la caméra indiqué sous l'entité.
6. Une fois que vous avez sélectionné la caméra placée dans votre entité, la vue de la caméra des scènes s'adapte à la perspective définie de la caméra placée.

## Montage de scène amélioré

AWS IoT TwinMaker les scènes comportent un ensemble d'outils permettant d'améliorer, d'éditer et de manipuler les ressources présentes dans votre scène.

Les rubriques suivantes vous expliquent comment utiliser les fonctionnalités d'édition améliorées dans vos AWS IoT TwinMaker scènes.

- [Placement ciblé des objets de scène](#)
- [Sélection du sous-modèle](#)
- [Modifier les entités dans la hiérarchie des scènes](#)

## Placement ciblé des objets de scène

AWS IoT TwinMaker vous permet de placer et d'ajouter des objets avec précision dans votre scène. Cette fonction d'édition améliorée vous permet de mieux contrôler l'endroit où vous placez les balises, les entités, les lumières et les modèles dans votre scène.

1. Accédez à votre scène dans la [AWS IoT TwinMaker console](#).
2. Appuyez sur le bouton +, puis sélectionnez l'une des options dans le menu déroulant. Il peut s'agir d'un modèle, d'une lampe, d'un tag ou de tout autre élément du menu +.

Lorsque vous déplacez votre curseur dans l'espace 3D de votre scène, vous devriez voir une cible autour de votre curseur.

3. Utilisez la cible pour placer des éléments avec précision dans votre scène.

## Sélection du sous-modèle


AWS IoT TwinMaker vous permet de sélectionner des sous-modèles de modèles 3D dans des scènes et de leur appliquer des propriétés standard, telles que des balises, des lumières ou des règles.

Les formats de fichier de modèle 3D contiennent des métadonnées qui peuvent spécifier des sous-zones du modèle en tant que sous-modèles au sein d'un modèle plus large. Par exemple, un modèle peut être un système de filtration, les différentes parties du système, telles que les réservoirs, les tuyaux ou un moteur, sont marquées comme des sous-modèles du modèle 3D de la filtration.

Formats de fichiers 3D pris en charge dans les scènes : GLB et GLTF.

1. Accédez à votre scène dans la [AWS IoT TwinMaker console](#).
2. Si votre scène ne contient aucun modèle, assurez-vous d'en ajouter un en sélectionnant l'option dans le menu +.

3. Sélectionnez le modèle répertorié dans la hiérarchie de votre scène. Une fois sélectionné, la hiérarchie doit afficher tous les sous-modèles situés sous le modèle.

 Note

Si aucun sous-modèle n'est répertorié, il est probable que le modèle n'ait pas été configuré pour comporter de sous-modèles.

4. Pour activer la visibilité d'un sous-modèle, appuyez sur l'icône en forme d'œil, située à droite du nom du sous-modèle dans la hiérarchie.
5. Pour modifier les données du sous-modèle, telles que son nom ou sa position, l'inspecteur de scène s'ouvre lorsqu'un sous-modèle est sélectionné. Utilisez le menu de l'inspecteur pour mettre à jour ou modifier les données du sous-modèle.
6. Pour ajouter des balises, des lumières, des règles ou d'autres propriétés aux sous-modèles, appuyez sur le + pendant que le sous-modèle est sélectionné dans la hiérarchie.

## Modifier les entités dans la hiérarchie des scènes

AWS IoT TwinMaker les scènes vous permettent de modifier directement les propriétés des entités dans la table hiérarchique. La procédure suivante indique les actions que vous pouvez effectuer sur une entité via le menu de hiérarchie.

1. Accédez à votre scène dans la [AWS IoT TwinMaker console](#).
2. Ouvrez la hiérarchie des scènes et sélectionnez un sous-élément d'une entité que vous souhaitez manipuler.
3. Une fois l'élément sélectionné, appuyez sur le bouton +, puis dans le menu déroulant, sélectionnez l'une des options suivantes :
  - Ajouter un nœud vide
  - Ajouter un modèle 3D
  - Ajoutez de la lumière
  - Ajouter une caméra depuis la vue actuelle
  - Ajouter un tag
  - Ajouter un shader de modèle
  - Ajouter un indicateur de mouvement

- Après avoir sélectionné l'une des options dans le menu déroulant, la sélection sera appliquée à la scène en tant qu'enfant de l'élément sélectionné à l'étape 2.
- Vous pouvez réorganiser les éléments enfants et les éléments reparents en sélectionnant un élément enfant et en le faisant glisser dans la hiérarchie vers un nouveau parent.

## Ajouter des annotations aux entités

Le compositeur de AWS IoT TwinMaker scène vous permet d'annoter n'importe quel élément de la hiérarchie de scène. L'annotation est rédigée en Markdown.

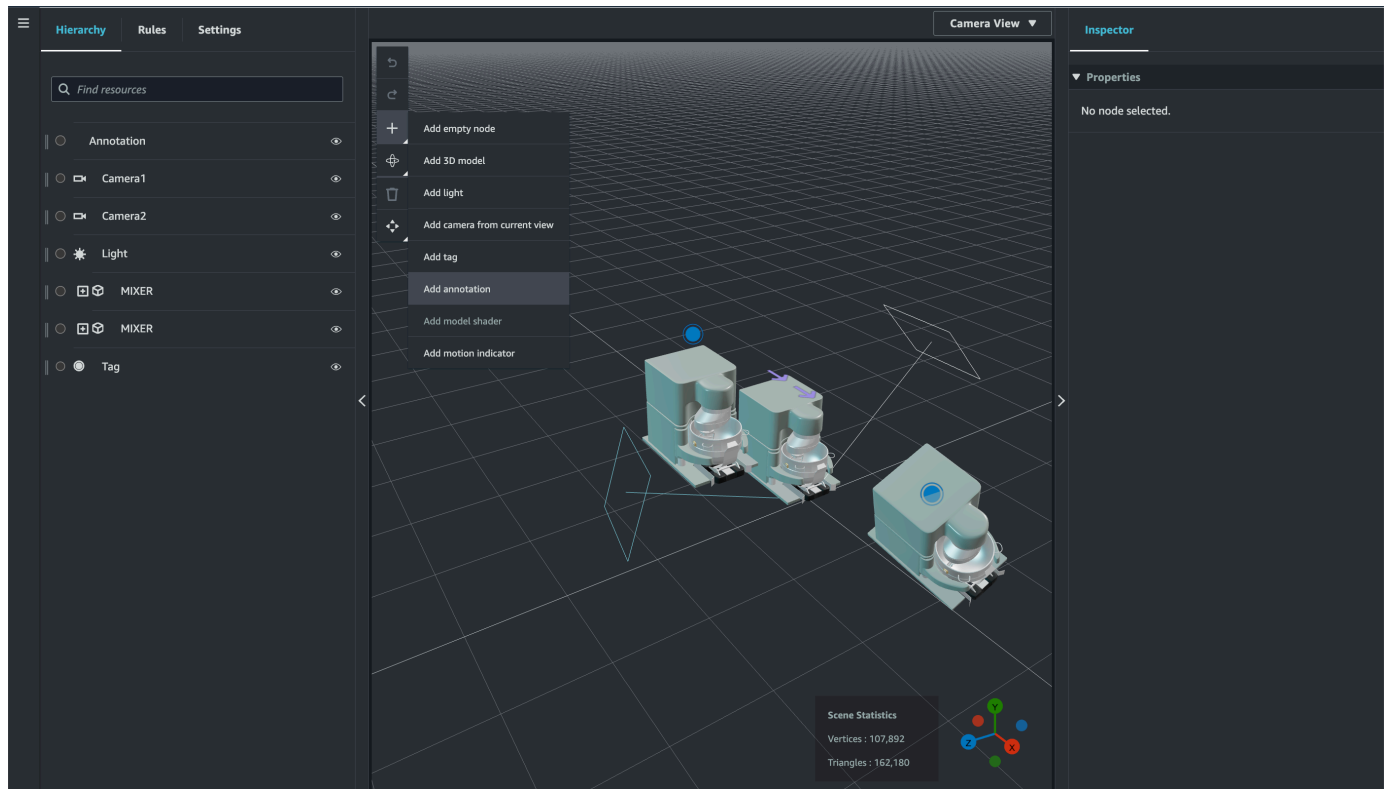
Pour plus d'informations sur l'écriture en Markdown, consultez la documentation officielle sur la syntaxe Markdown, [Basic Syntax](#).

### Note

AWS IoT TwinMaker annotations et superpositions : syntaxe Markdown uniquement et non HTML.

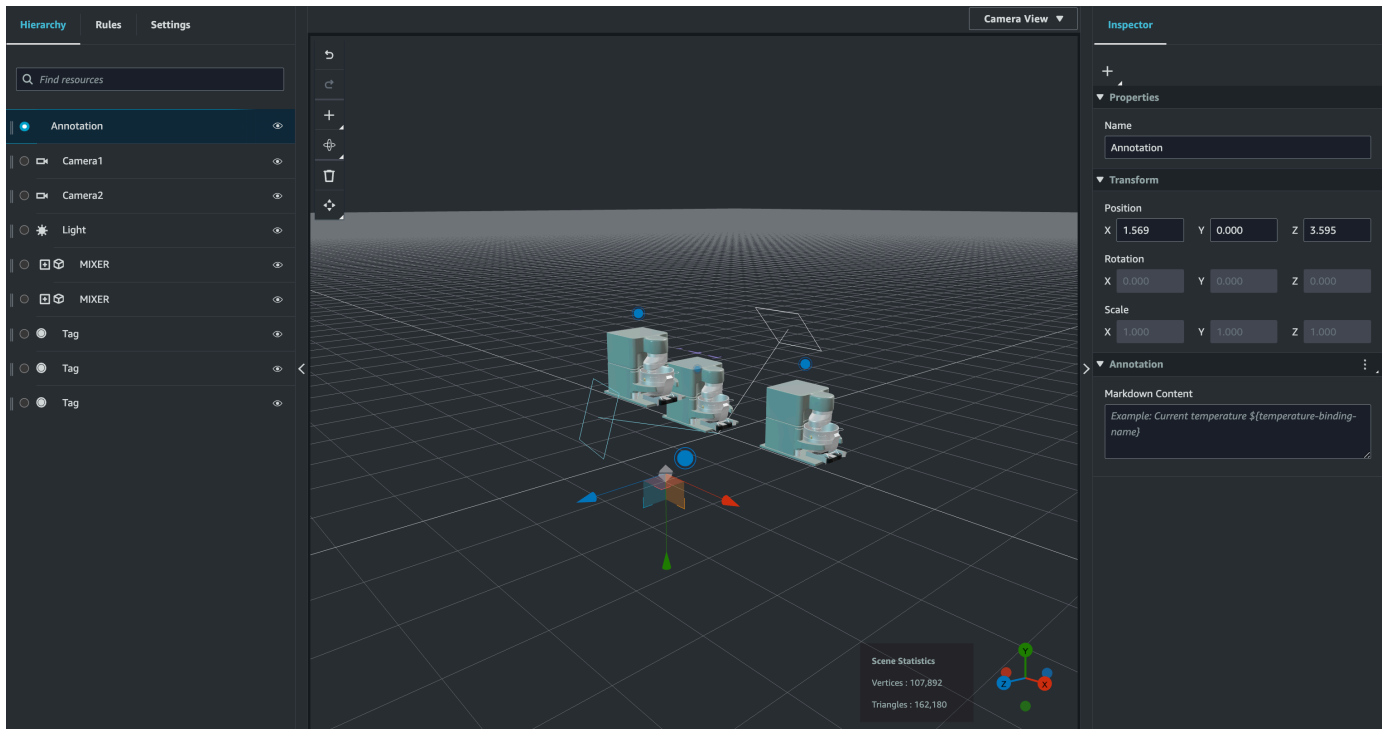
### Ajouter une annotation à une entité

- Accédez à votre scène dans la [AWS IoT TwinMaker console](#).
- Sélectionnez un élément dans la hiérarchie des scènes que vous souhaitez annoter. Si aucun élément de la hiérarchie n'est sélectionné, vous pouvez ajouter une annotation à la racine.
- Appuyez sur le bouton plus + et choisissez l'option Ajouter une annotation.



4. Dans la fenêtre Inspector sur la gauche, faites défiler la page vers le bas jusqu'à la section des annotations. À l'aide de la syntaxe Markdown, écrivez le texte que vous souhaitez que votre annotation affiche.

Pour plus d'informations sur l'écriture en Markdown, consultez la documentation officielle sur la syntaxe Markdown, [Basic Syntax](#).



5. Pour lier les données de votre AWS IoT TwinMaker scène à une annotation, choisissez Ajouter une liaison de données, ajoutez l'identifiant de l'entité, puis sélectionnez le nom du composant et le nom de propriété de l'entité à partir de laquelle vous souhaitez faire apparaître les données. Vous pouvez mettre à jour le nom de la liaison pour l'utiliser comme variable Markdown et faire apparaître les données dans l'annotation.

# 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

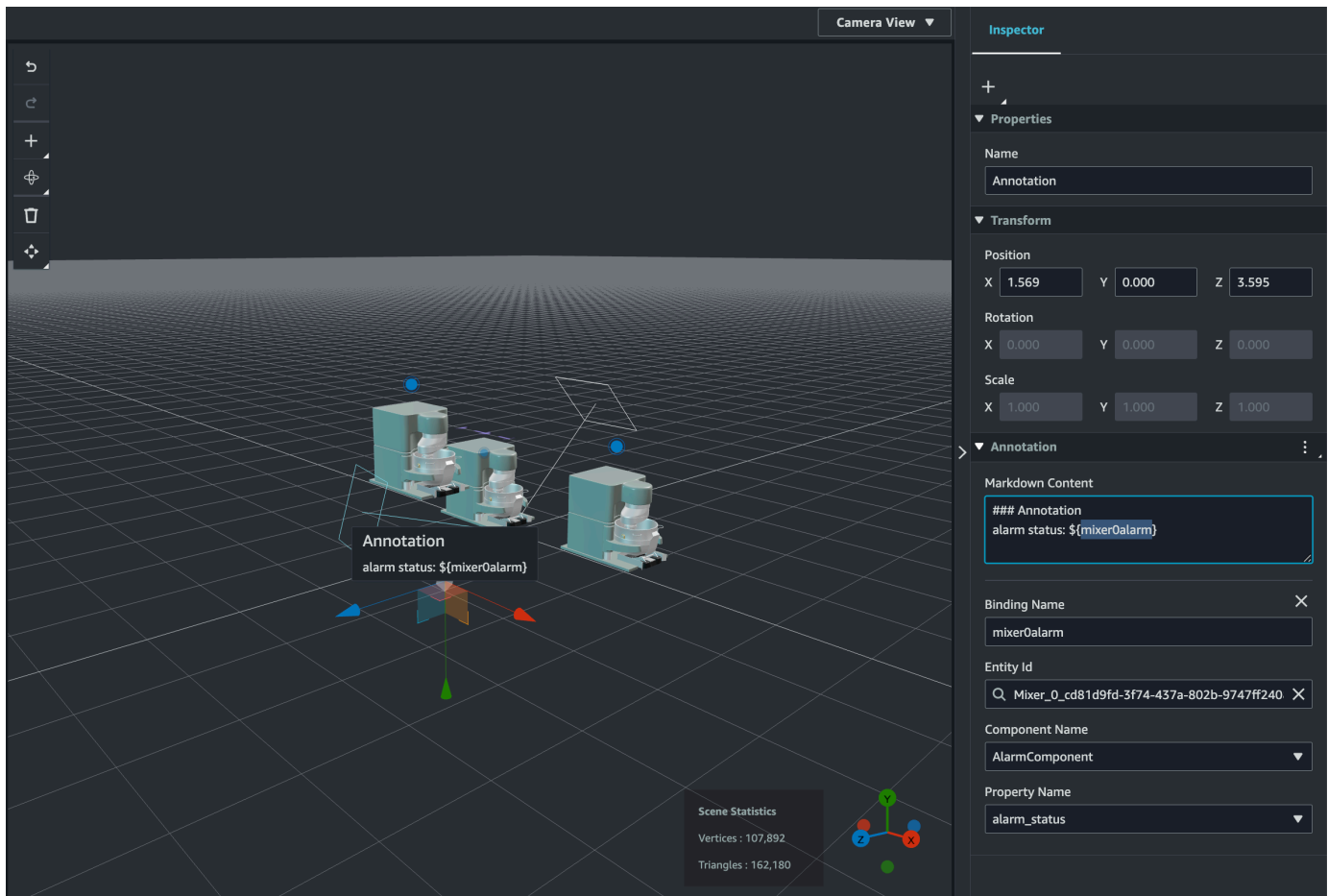
*Example: Current temperature `#{temperature-binding-name}`*

## 6. Le nom de liaison est utilisé pour représenter la variable de l'annotation.

Entrez un nom de liaison pour faire apparaître la dernière valeur historique d'une série chronologique d'une entité dans la syntaxe variable AWS IoT TwinMaker de l'annotation :

`${variable-name}`

Par exemple, cette superposition affiche la valeur `demixer0alarm`, dans l'annotation avec la syntaxe `${mixer0alarm}`.



## Ajouter des superpositions aux balises

Vous pouvez créer des superpositions pour vos AWS IoT TwinMaker scènes. Les superpositions de scènes sont associées à des balises et peuvent être utilisées pour faire apparaître des données critiques associées à vos entités de scène. La superposition est créée et rendue dans Markdown.

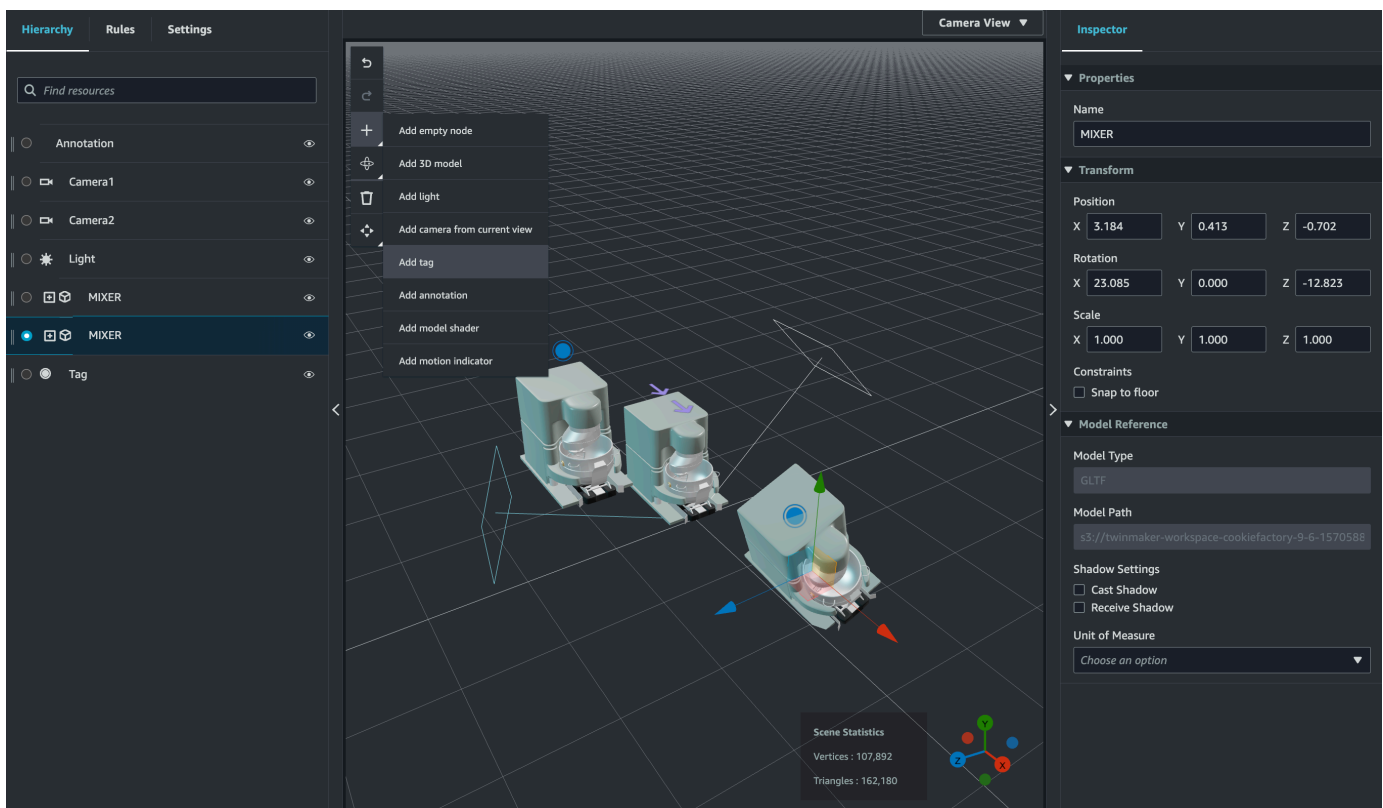
Pour plus d'informations sur l'écriture en Markdown, consultez la documentation officielle sur la syntaxe Markdown, [Basic Syntax](#).

**Note**

Par défaut, une superposition n'est visible dans une scène que lorsque la balise qui lui est associée est sélectionnée. Vous pouvez activer cette option dans les paramètres de la scène afin que toutes les superpositions soient visibles en même temps.

1. Accédez à votre scène dans la [AWS IoT TwinMaker console](#).
2. La AWS IoT TwinMaker superposition est associée à une scène de tag, vous pouvez mettre à jour un tag existant ou en ajouter un nouveau.

Appuyez sur le bouton Plus + et choisissez l'option Ajouter une étiquette.



3. Dans le panneau Inspector sur la droite, sélectionnez le bouton + (symbole plus), puis sélectionnez Ajouter une superposition.

## Inspector



Add overlay

Add entity binding

### Default Icon

Choose an icon



### Entity Id



### Component Name

Select an option



### Property Name

Select an option



### Rule Id

Choose a rule

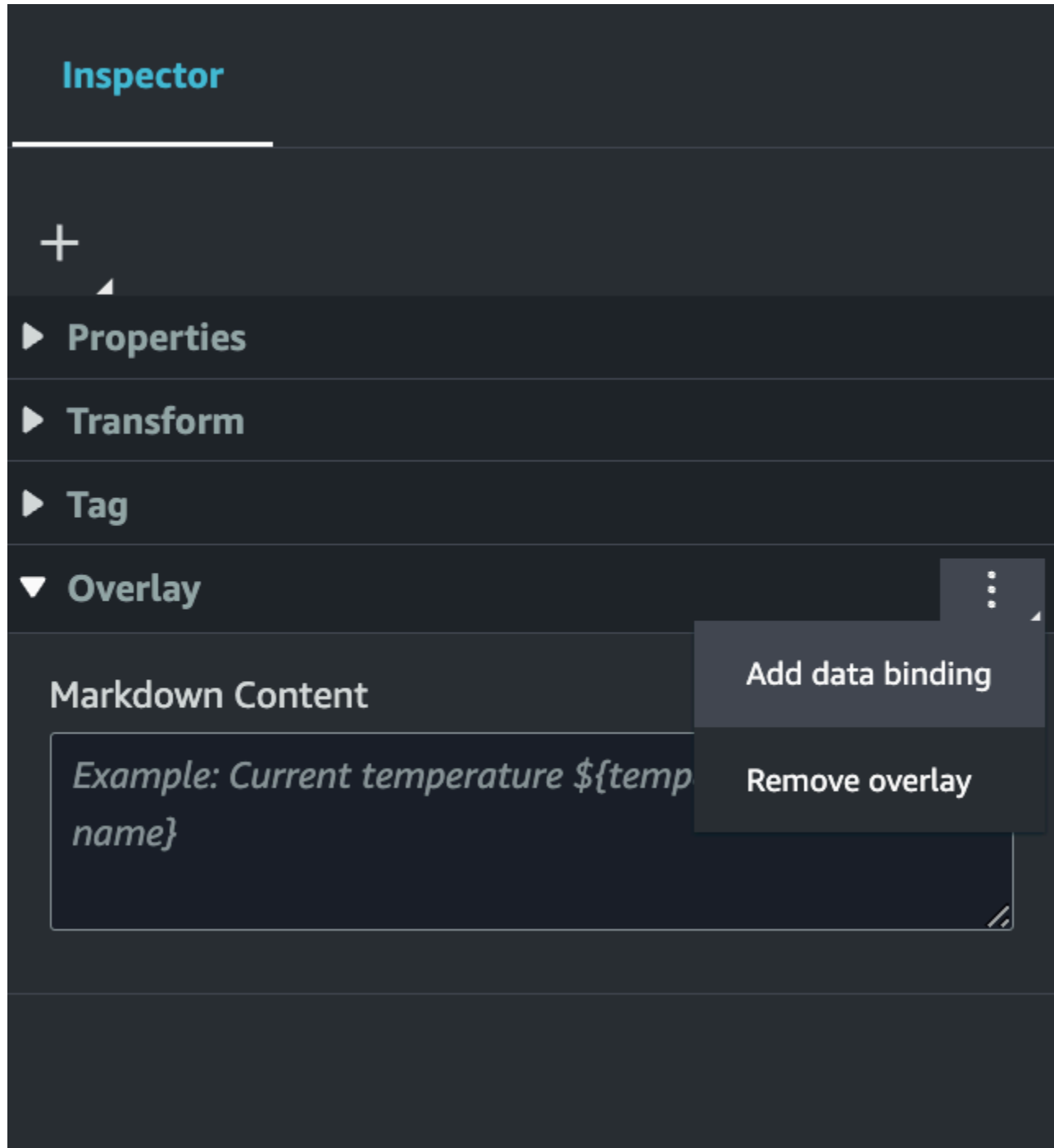


### Link Target

4. Dans la syntaxe Markdown, écrivez le texte que vous souhaitez afficher dans votre superposition.

Pour plus d'informations sur l'écriture en Markdown, consultez la documentation officielle sur la syntaxe Markdown, [Basic Syntax](#).

5. Pour lier les données de votre AWS IoT TwinMaker scène à une superposition, sélectionnez Ajouter une liaison de données.



Ajoutez le nom de la liaison et l'identifiant de l'entité, puis sélectionnez le nom du composant et le nom de propriété de l'entité dont vous souhaitez faire apparaître les données.

6. Vous pouvez faire apparaître la dernière valeur historique des données chronologiques d'une entité dans la syntaxe variable AWS IoT TwinMaker de la superposition :. `${variable-name}`

Par exemple, cette superposition affiche la valeur `demixer0alarm`, dans la superposition avec la syntaxe. `${mixer0alarm}`

## Inspector



▶ Properties

▶ Transform

▶ Tag

▼ Overlay ⋮

### Markdown Content

```
### Overlay  
alarm status: ${mixer0alarm}
```

Binding Name ✕

mixer0alarm

Entity Id


🔍 Mixer\_0\_cd81d9fd-3f74-437a-802b-9747ff240 ✕

Component Name

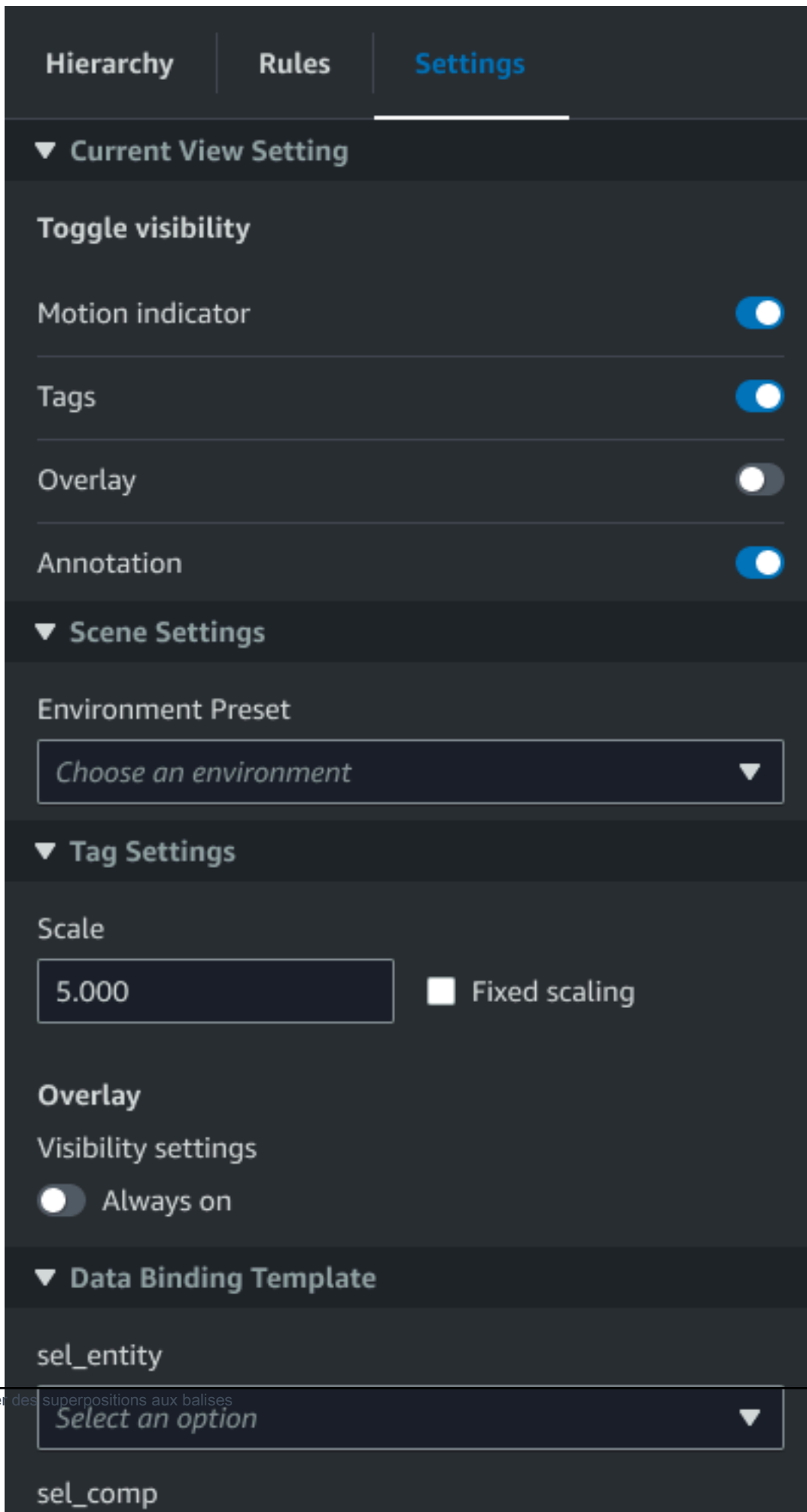
AlarmComponent ▼

Property Name

7. Pour activer la visibilité des superpositions, ouvrez l'onglet Paramètres en haut à gauche et assurez-vous que le bouton Superposition est activé afin que toutes les superpositions soient visibles en même temps.

 Note

Par défaut, une superposition n'est visible dans une scène que lorsque la balise qui lui est associée est sélectionnée.



# Modifiez vos scènes

Après avoir créé une scène, vous pouvez y ajouter des entités, des composants et configurer des widgets augmentés. Utilisez des composants d'entité et des widgets pour modéliser votre jumeau numérique et fournir des fonctionnalités adaptées à votre cas d'utilisation.

## Rubriques

- [Ajoutez des modèles à vos scènes](#)
- [Ajoutez des widgets d'interface utilisateur augmentée avec Model Shader à votre scène](#)
- [Création de tags pour vos scènes](#)

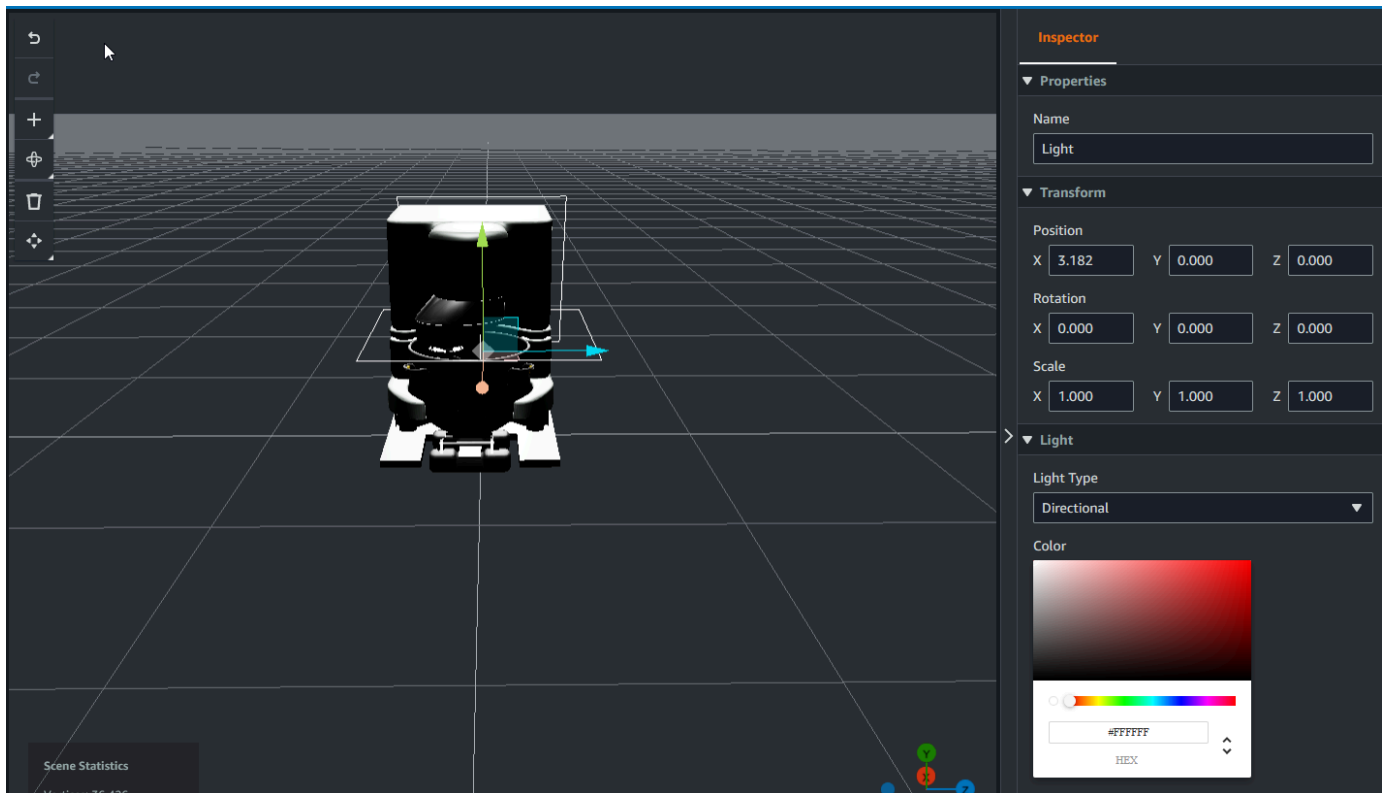
## Ajoutez des modèles à vos scènes

Pour ajouter des modèles à votre scène, procédez comme suit.

### Note

Pour ajouter des modèles dans votre scène, vous devez d'abord les télécharger dans la bibliothèque de AWS IoT TwinMaker ressources. Pour de plus amples informations, veuillez consulter [Télécharger des ressources dans la bibliothèque AWS IoT TwinMaker de ressources](#).

1. Sur la page du compositeur de scène, choisissez le signe plus (+), puis choisissez Ajouter un modèle 3D.
2. Dans la fenêtre Ajouter une ressource à partir de la bibliothèque de ressources, sélectionnez le fichier CookieFactorMixer.glb, puis sélectionnez Ajouter. Le compositeur de scène s'ouvre.
3. Facultatif : choisissez le signe plus (+), puis choisissez Ajouter de la lumière.
4. Choisissez chaque option d'éclairage pour voir comment elle affecte la scène.



### Note

Les scènes disposent d'un éclairage d'ambiance par défaut. Pour éviter toute perte de fréquence d'images, pensez à limiter le nombre de lumières supplémentaires placées dans votre scène.

## Ajoutez des widgets d'interface utilisateur augmentée avec Model Shader à votre scène

Les widgets Model Shader peuvent modifier la couleur d'un objet selon les conditions que vous définissez. Par exemple, vous pouvez créer un widget couleur qui modifie la couleur d'un mixeur de cookies dans votre scène en fonction des données de température du mixeur.

Utilisez la procédure suivante pour ajouter des widgets de model-shader à un objet sélectionné.

1. Sélectionnez un objet dans la hiérarchie auquel vous souhaitez ajouter un widget. Appuyez sur le bouton +, puis choisissez Model Shader.

2. Pour ajouter un nouveau groupe de règles visuelles, suivez d'abord les instructions ci-dessous pour le créer ColorRule, puis choisissez dans le panneau Inspector pour l'objet de l'ID de règle ColorRule.
3. Sélectionnez l'EntityId ComponentName, et PropertyName vous souhaitez lier le shader du modèle à.

## Créez des règles visuelles pour vos scènes

Vous pouvez utiliser des cartes de règles visuelles pour spécifier les conditions basées sur les données qui modifient l'apparence visuelle d'un widget d'interface utilisateur augmentée, tel qu'une balise ou un shader de modèle. Des exemples de règles sont fournis, mais vous pouvez également créer les vôtres. L'exemple suivant montre une règle visuelle.

The screenshot displays the AWS IoT TwinMaker console interface for managing rules. It features a dark theme and a vertical list of three rules. Each rule configuration includes an 'Expression' field, a 'Target' section with an 'Icon' dropdown and a severity dropdown (Error, Warning, or Info), and a 'Remove statement' button. At the bottom of the rule list, there are buttons for 'Add new statement' and 'Remove Rule'. Below the rule list, a specific rule named 'sampleTimeSeriesColorRule' is expanded, showing 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

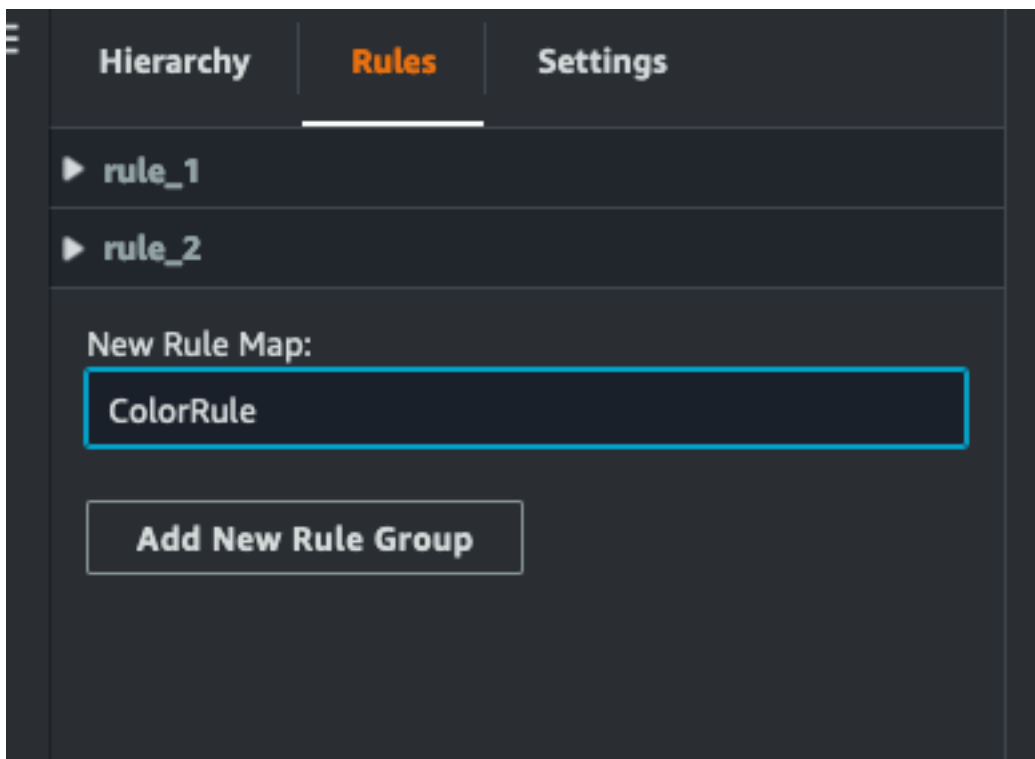
▶ sampleTimeSeriesColorRule

Rule Id

L'image ci-dessus montre une règle concernant le moment où une propriété de données définie précédemment avec l'ID « température » est vérifiée par rapport à une certaine valeur. Par exemple, si la « température » est supérieure ou égale à 40, l'état transformera l'apparence de l'étiquette en cercle rouge. La cible, lorsqu'elle est choisie dans le tableau de bord Grafana, remplit un panneau détaillé configuré pour utiliser la même source de données.

La procédure suivante explique comment ajouter un nouveau groupe de règles visuelles pour la couche d'interface utilisateur augmentée par colorisation du maillage.

1. Dans l'onglet règles de la console, entrez un nom tel que ColorRule dans le champ de texte et choisissez Ajouter un nouveau groupe de règles.



2. Définissez une nouvelle règle pour votre cas d'utilisation. Par exemple, vous pouvez en créer un en fonction de la propriété de données « temperature », où la valeur déclarée est inférieure à 20. Utilisez la syntaxe suivante pour les expressions de règles : inférieur à <, supérieur à >, inférieur ou égal est <=, supérieur ou égal est >= et égal est ==. (Pour plus d'informations, consultez la [syntaxe JEXL d'Apache Commons](#).)
3. Définissez la cible sur une couleur. Pour définir une couleur, utilisez par exemple #fcb03 des valeurs hexadécimales. (Pour plus d'informations sur les valeurs hexadécimales, voir [Hexadécimal](#).)

## Création de tags pour vos scènes

Une balise est une annotation ajoutée à une position de  $x, y, z$  coordonnées spécifique d'une scène. La balise utilise une propriété d'entité pour connecter une partie de scène au graphe de connaissances. Vous pouvez utiliser une balise pour configurer le comportement ou l'apparence visuelle d'un élément de la scène, tel qu'une alarme.

### Note

Pour ajouter des fonctionnalités aux balises, vous devez leur appliquer des règles visuelles.

Pour ajouter des balises à votre scène, procédez comme suit.

1. Sélectionnez un objet dans la hiérarchie, cliquez sur le bouton +, puis choisissez Ajouter une étiquette.
2. Donnez un nom au tag. Ensuite, pour appliquer une règle visuelle, sélectionnez un identifiant de groupe visuel.
3. Dans les listes déroulantes, choisissez ComponentName EntityId, et. PropertyName
4. Pour renseigner le champ Data Path, choisissez Create DataFrameLabel.

## Format du modèle 3D Tiles

### Utilisation de tuiles 3D dans votre scène

Si les temps d'attente sont longs lorsque vous chargez des scènes 3D AWS IoT TwinMaker ou si les performances de rendu sont médiocres lorsque vous naviguez dans un modèle 3D complexe, vous souhaitez peut-être convertir vos modèles en tuiles 3D. Cette section décrit le format des tuiles 3D et les outils tiers disponibles. Poursuivez votre lecture pour savoir si 3D Tiles convient à votre cas d'utilisation et pour obtenir de l'aide pour démarrer.

### Cas d'utilisation d'un modèle complexe

La présence d'un modèle 3D dans votre AWS IoT TwinMaker scène peut entraîner des problèmes de performances tels que des temps de chargement lents et des retards de navigation si le modèle est :

- Grand : la taille de son fichier est supérieure à 100 Mo.

- Dense : il est composé de centaines ou de milliers de mailles distinctes.
- Complexe : la géométrie du maillage comporte des millions de triangles pour former des formes complexes.

## Format 3D Tiles

[Le format 3D Tiles](#) est une solution pour diffuser la géométrie des modèles et améliorer les performances de rendu 3D. Il permet le chargement instantané de modèles 3D dans une AWS IoT TwinMaker scène et optimise les interactions 3D en chargeant des parties d'un modèle en fonction de ce qui est visible dans la vue de la caméra.

Le format 3D Tiles a été créé par [Cesium](#). Cesium dispose d'un service géré pour convertir les modèles 3D en tuiles 3D appelé [Cesium Ion](#). Il s'agit actuellement de la meilleure solution pour créer des tuiles 3D, et nous la recommandons pour vos modèles complexes dans les [formats pris en charge](#). Vous pouvez enregistrer Cesium et choisir le plan d'abonnement approprié en fonction des besoins de votre entreprise sur la page [de tarification de Cesium](#).

Pour préparer un modèle 3D Tiles que vous pouvez ajouter à une AWS IoT TwinMaker scène, suivez les instructions documentées par Cesium Ion :

- [Importer un modèle dans Cesium Ion](#)

## Téléchargez des tuiles Cesium 3D sur AWS

Une fois que votre modèle a été converti en 3D Tiles, téléchargez les fichiers du modèle, puis chargez-les dans le compartiment Amazon S3 de votre AWS IoT TwinMaker espace de travail :

1. [Créez et téléchargez votre archive de modèles 3D Tiles](#).
2. Décompressez l'archive dans un dossier.
3. Téléchargez l'intégralité du dossier 3D Tiles dans le compartiment Amazon S3 associé à votre AWS IoT TwinMaker espace de travail. (Consultez la section [Chargement d'objets](#) dans le guide de l'utilisateur Amazon S3.)
4. Si votre modèle 3D Tiles a été chargé avec succès, vous verrez un chemin de dossier Amazon S3 dans votre [bibliothèque de AWS IoT TwinMaker ressources](#) avec typeTiles3D.

**Note**

La bibliothèque de AWS IoT TwinMaker ressources ne prend pas en charge le téléchargement direct de modèles 3D Tiles.

## Utilisation de tuiles 3D dans AWS IoT TwinMaker

AWS IoT TwinMaker est au courant de tout modèle 3D Tiles chargé dans le compartiment S3 de votre espace de travail. Le modèle doit avoir un `tileset.json` et tous les fichiers dépendants (`.gltf`, `.b3dm`, `.i3dm`, `.cmpt`, `.pnts`) disponibles dans le même répertoire Amazon S3. Le chemin du répertoire Amazon S3 apparaîtra dans la bibliothèque de ressources avec le `typeTiles3D`.

Pour ajouter le modèle 3D Tiles à votre scène, procédez comme suit :

1. Sur la page du compositeur de scène, choisissez le signe plus (+), puis choisissez Ajouter un modèle 3D.
2. Dans la fenêtre Ajouter une ressource à partir de la bibliothèque de ressources, choisissez le chemin d'accès à votre modèle 3D Tiles avec le `typeTiles3D`, puis choisissez Ajouter.
3. Cliquez sur le canevas pour placer le modèle dans votre scène.

### Différences entre les tuiles 3D

3D Tiles ne prend actuellement pas en charge les métadonnées géométriques et sémantiques, ce qui signifie que la hiérarchie de maillage du modèle d'origine n'est pas disponible pour la fonction de sélection du sous-modèle. Vous pouvez toujours ajouter des widgets à votre modèle 3D Tiles, mais vous ne pouvez pas utiliser les fonctionnalités adaptées aux sous-modèles : shader de modèle, transformations 3D séparées ou liaison d'entités pour un maillage de sous-modèle.

Il est recommandé d'utiliser la conversion de tuiles 3D pour les éléments volumineux qui servent de contexte à l'arrière-plan d'une scène. Si vous souhaitez qu'un sous-modèle soit encore décomposé et annoté, il doit être extrait en tant que glTF/glb ressource distincte et ajouté directement à la scène. Cela peut être fait avec des outils 3D gratuits et courants tels que [Blender](#).

Exemple de cas d'utilisation :

- Vous disposez d'un modèle d'usine de 1 Go avec des salles des machines et des sols détaillés, des boîtiers électriques et des tuyaux de plomberie. Les boîtiers et tuyaux électriques doivent briller en rouge lorsque les données de propriété associées franchissent un seuil.

- Vous isolez les mailles de boîte et de tuyauterie dans le modèle et vous les exportez dans un glTF distinct à l'aide de Blender.
- Vous convertissez l'usine sans éléments électriques et de plomberie en un modèle 3D Tiles et vous le téléchargez sur S3.
- Vous ajoutez à la fois le modèle 3D Tiles et le modèle glTF à une AWS IoT TwinMaker scène à l'origine (0,0,0).
- Vous ajoutez des composants de model-shader aux sous-modèles de boîtiers électriques et de tuyaux du glTF pour rendre les maillages rouges en fonction des règles de propriété.

## Scènes dynamiques

AWS IoT TwinMaker les scènes exploitent la puissance du [graphe de connaissances](#) en stockant les nœuds de scène et les paramètres dans un composant d'entité. Utilisez la AWS IoT TwinMaker console pour créer des scènes dynamiques afin de gérer, de créer et de rendre plus facilement des scènes 3D.

Caractéristiques principales :

- Tous les objets, paramètres et liaisons de données des nœuds de scène 3D sont rendus « dynamiquement » sur la base de requêtes de graphes de connaissances.
- Si vous utilisez le visualiseur de scènes en lecture seule dans une application Grafana ou personnalisée, vous pouvez obtenir des mises à jour de vos scènes toutes les 30 secondes.

## Scènes statiques ou dynamiques

Les scènes statiques sont composées d'un fichier JSON de scène stocké dans S3 qui contient les détails de tous les nœuds et paramètres de scène. Toute modification apportée à la scène doit être apportée au document JSON et enregistrée dans S3. Une scène statique est la seule option si vous avez un [plan tarifaire de base](#).

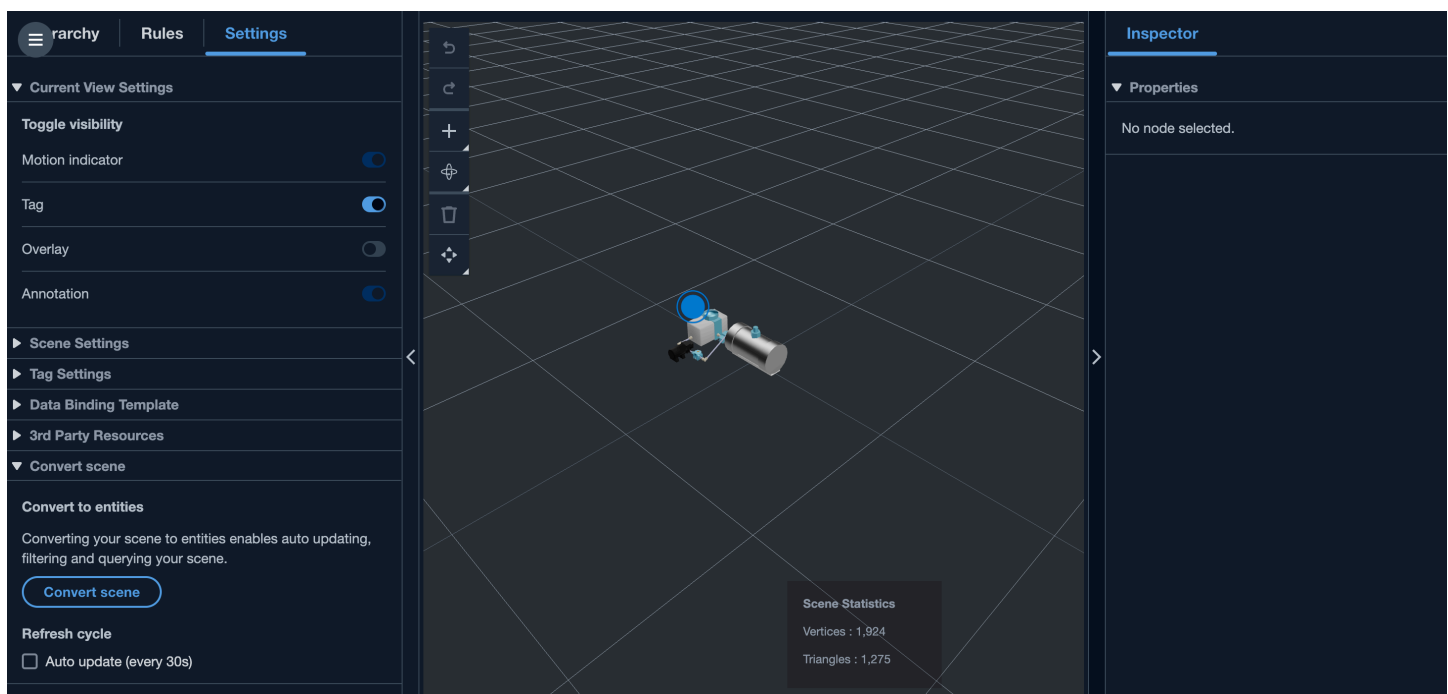
Les scènes dynamiques sont composées d'un fichier JSON contenant les paramètres globaux de la scène, tandis que tous les autres nœuds de scène et paramètres de nœuds sont stockés en tant que composants d'entité dans le graphe de connaissances. Les scènes dynamiques ne sont prises en charge que dans les plans tarifaires standard et échelonnés. Consultez [Changer de mode de AWS IoT TwinMaker tarification](#) pour plus d'informations sur la manière de mettre à niveau votre plan tarifaire).

Vous pouvez convertir une scène statique existante en scène dynamique en procédant comme suit :

- Accédez à votre scène dans la [AWS IoT TwinMaker console](#).
- Sur le panneau de gauche, cliquez sur l'onglet Paramètres.
- Développez la section Convertir la scène en bas du panneau.
- Cliquez sur le bouton Convertir la scène, puis sur Confirmer.

### Warning

La conversion d'une scène statique en scène dynamique est irréversible.



## Types de composants et entités de scène

Afin de créer des composants d'entité spécifiques à une scène, les types de composants 1P suivants sont pris en charge :

- `com.amazon.iottwinmaker.3d.component.camera` [Type de composant qui stocke les paramètres d'un widget de caméra.](#)
- `com.amazon.iottwinmaker.3d.component.dataoverlay` Type de composant qui stocke les paramètres d'une superposition [d'un widget d'annotation ou de balise.](#)

- `com.amazon.iottwinmaker.3d.component.light` Type de composant qui stocke les paramètres d'un widget lumineux.
- `com.amazon.iottwinmaker.3d.component.modelref` Type de composant qui stocke les paramètres et l'emplacement S3 d'un modèle 3D utilisé dans une scène.
- `com.amazon.iottwinmaker.3d.component.modelshader` Type de composant qui stocke les paramètres d'un model shader [sur un modèle 3D](#).
- `com.amazon.iottwinmaker.3d.component.motionindicator` Type de composant qui stocke les paramètres d'un widget indicateur de mouvement.
- `com.amazon.iottwinmaker.3d.component.submodelref` Type de composant qui stocke les paramètres d'un sous-modèle d'un modèle [3D](#).
- `com.amazon.iottwinmaker.3d.component.tag` [Type de composant qui stocke les paramètres d'un widget de balises](#).
- `com.amazon.iottwinmaker.3d.node` Type de composant qui stocke les paramètres de base d'un nœud de scène, tels que sa transformation 3D, son nom et ses propriétés génériques.

## Concepts de scène dynamiques

Les entités de scène dynamiques sont stockées sous une entité globale étiquetée `$SCENES`. Chaque scène est composée d'une entité racine et d'une hiérarchie d'entités enfants correspondant à la hiérarchie des nœuds de scène. Chaque nœud de scène situé sous la racine possède un composant `com.amazon.iottwinmaker.3d.node` et un composant correspondant au type de nœud (modèle 3D, widget, etc.).

### Warning

Ne supprimez aucune entité de scène manuellement, sinon votre scène est peut-être en mauvais état. Si vous souhaitez supprimer partiellement ou totalement une scène, utilisez la page du compositeur de scène pour ajouter et supprimer des nœuds de scène, et utilisez la page des scènes pour sélectionner et supprimer une scène.

# Création d'une application Web personnalisée à l'aide des composants de l' AWS IoT TwinMaker interface utilisateur

AWS IoT TwinMaker fournit des composants d'interface utilisateur open source pour les développeurs AWS IoT d'applications. À l'aide de ces composants d'interface utilisateur, les développeurs peuvent créer des applications Web personnalisées avec des AWS IoT TwinMaker fonctionnalités activées pour leurs jumeaux numériques.

AWS IoT TwinMaker Les composants de l'interface utilisateur font partie du kit d' AWS IoT application, une bibliothèque open source côté client qui permet aux développeurs d'applications IoT de simplifier le développement d'applications IoT complexes

AWS IoT TwinMaker Les composants de l'interface utilisateur incluent :

- AWS IoT TwinMaker source :

Un composant de connecteur de données qui vous permet de récupérer des données et d'interagir avec vos AWS IoT TwinMaker données et vos jumeaux numériques.

Pour plus d'informations, consultez la documentation [AWS IoT TwinMaker source](#).

- Visualiseur de scènes :

Un composant de rendu 3D intégré `@react-three/fiber` qui affiche votre jumeau numérique et vous permet d'interagir avec lui.

Pour plus d'informations, consultez la documentation de [Scene Viewer](#).

- Lecteur vidéo :

Composant de lecteur vidéo qui vous permet de diffuser une vidéo depuis Kinesis Video Streams AWS IoT TwinMaker via.

Pour plus d'informations, consultez la documentation [du lecteur vidéo](#).

Pour en savoir plus sur l'utilisation du kit AWS IoT d'application, rendez-vous sur la page [Github du kit d'AWS IoT application](#).

Pour savoir comment démarrer une nouvelle application Web à l'aide d' AWS IoT Application Kit, rendez-vous sur la page de documentation officielle de [l'IoT App Kit](#).

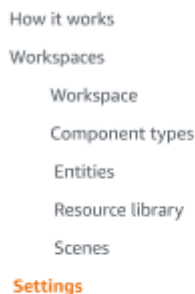
# Changer de mode de AWS IoT TwinMaker tarification

AWS IoT TwinMaker propose actuellement trois modes de tarification : forfait de base, standard ou forfait à plusieurs niveaux. Le mode de tarification standard est défini comme mode de tarification par défaut.





Vous pouvez passer du mode de tarification basé sur l'utilisation au mode de tarification par niveaux à tout moment, mais le changement prendra effet au début de votre prochain cycle de facturation. Une fois que vous êtes passé du mode de tarification basé sur l'utilisation au mode de tarification par niveaux, vous ne pouvez pas revenir au mode de tarification basé sur l'utilisation pour les trois prochains cycles d'utilisation. Si vous passez de la version de base à la version standard, la modification prend effet immédiatement. Pour plus de détails et des informations sur les coûts, voir [AWS IoT TwinMaker Tarification](#)

Cette procédure explique comment changer de mode de tarification dans la [AWS IoT TwinMaker console](#) :

1. Ouvrez la [AWS IoT TwinMaker console](#).
2. Dans le volet de navigation de gauche, sélectionnez Paramètres. La page de tarification s'ouvre.



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

What's new   
Documentation   
FAQ   
Pricing 

3. Choisissez le mode Modifier le prix.
4. Sélectionnez le mode de bundle standard ou hiérarchisé, comme indiqué dans la capture d'écran suivante.

### 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. Choisissez Enregistrer pour confirmer votre nouveau mode de tarification.
6. Vous venez de modifier votre mode de tarification.

#### Note

Vous pouvez passer du mode de tarification basé sur l'utilisation au mode de tarification par niveaux à tout moment, mais le changement prendra effet au début de votre prochain cycle de facturation. Une fois que vous êtes passé du mode de tarification basé sur l'utilisation au mode de tarification par niveaux, vous ne pouvez pas revenir au mode de tarification basé sur l'utilisation pour les trois prochains cycles d'utilisation. Si vous passez de la version de base à la version standard, la modification prend effet immédiatement.

# AWS IoT TwinMaker graphe de connaissances

Le graphe de AWS IoT TwinMaker connaissances organise toutes les informations contenues dans vos AWS IoT TwinMaker espaces de travail et les présente sous forme de graphique visuel. Vous pouvez exécuter des requêtes sur vos entités, composants et types de composants afin de générer des graphiques visuels illustrant les relations entre vos AWS IoT TwinMaker ressources.

Les rubriques suivantes expliquent comment utiliser et intégrer le graphe de connaissances.

## Rubriques

- [AWS IoT TwinMaker concepts de base du graphe de connaissances](#)
- [Comment exécuter des requêtes AWS IoT TwinMaker Knowledge Graph](#)
- [Intégration de scènes de graphes de connaissances](#)
- [Comment utiliser le AWS IoT TwinMaker Knowledge Graph avec Grafana](#)
- [AWS IoT TwinMaker Knowledge Graph \(ressources supplémentaires\)](#)

## AWS IoT TwinMaker concepts de base du graphe de connaissances

Cette rubrique couvre les concepts clés et le vocabulaire de la fonctionnalité Knowledge Graph.

Comment fonctionne le Knowledge Graph :

Le graphe de connaissances crée des relations entre les entités et leurs composants avec le [CreateEntity](#) ou existant [UpdateEntity](#) APIs. Une relation est simplement une propriété d'un type de données spécial [RELATIONSHIP](#) défini sur un composant d'une entité. AWS IoT TwinMaker Knowledge Graph appelle l'[ExecuteQuery](#) API pour effectuer une requête basée sur les données des entités ou sur les relations entre elles. Knowledge Graph utilise le langage de requête flexible partiQL (utilisé par de nombreux AWS services) qui a récemment été ajouté à la prise en charge de la syntaxe des correspondances de graphes pour vous aider à rédiger vos requêtes. Une fois les appels effectués, vous pouvez afficher les résultats sous forme de tableau ou les visualiser sous forme de graphique des nœuds et des arêtes connectés.

Termes clés du Knowledge Graph :

- Graphe d'entités : ensemble de nœuds et d'arêtes dans un espace de travail.

- **Nœud** : chaque entité de votre espace de travail devient un nœud dans le graphe d'entités.
- **Arête** : chaque propriété de relation définie sur un composant d'une entité devient une arête dans le graphe d'entités. En outre, une relation parent-enfant hiérarchique définie à l'aide du `parentEntityId` champ d'une entité devient également une arête dans le graphe d'entités avec un nom de relation « `isChildOf` ». Toutes les arêtes sont des arêtes directionnelles.
- **Relation** : une AWS IoT TwinMaker relation est un type spécial de propriété du composant d'une entité. Vous pouvez utiliser l'[UpdateEntity](#) API AWS IoT TwinMaker [CreateEntity](#) pour définir et modifier une relation. Dans AWS IoT TwinMaker, une relation doit être définie dans un composant d'une entité. Une relation ne peut pas être définie comme une ressource isolée. Une relation doit être directionnelle d'une entité à l'autre.

## Comment exécuter des requêtes AWS IoT TwinMaker Knowledge Graph

Avant d'utiliser le AWS IoT TwinMaker Knowledge Graph, assurez-vous d'avoir rempli les conditions préalables suivantes :

- Créez un AWS IoT TwinMaker espace de travail. Vous pouvez créer un espace de travail dans la [AWS IoT TwinMaker console](#).
- Familiarisez-vous avec AWS IoT TwinMaker le système de composants d'entités et apprenez à créer des entités. Pour de plus amples informations, veuillez consulter [Créer votre première entité](#).
- Familiarisez-vous avec AWS IoT TwinMaker les connecteurs de données. Pour de plus amples informations, veuillez consulter [AWS IoT TwinMaker connecteurs de données](#).

### Note

Pour utiliser le AWS IoT TwinMaker Knowledge Graph, vous devez utiliser le mode de tarification des offres groupées standard ou échelonné. Pour de plus amples informations, veuillez consulter [Changer de mode de AWS IoT TwinMaker tarification](#).

Les procédures suivantes vous montrent comment écrire, exécuter, enregistrer et modifier des requêtes.

## Ouvrez l'éditeur de requêtes

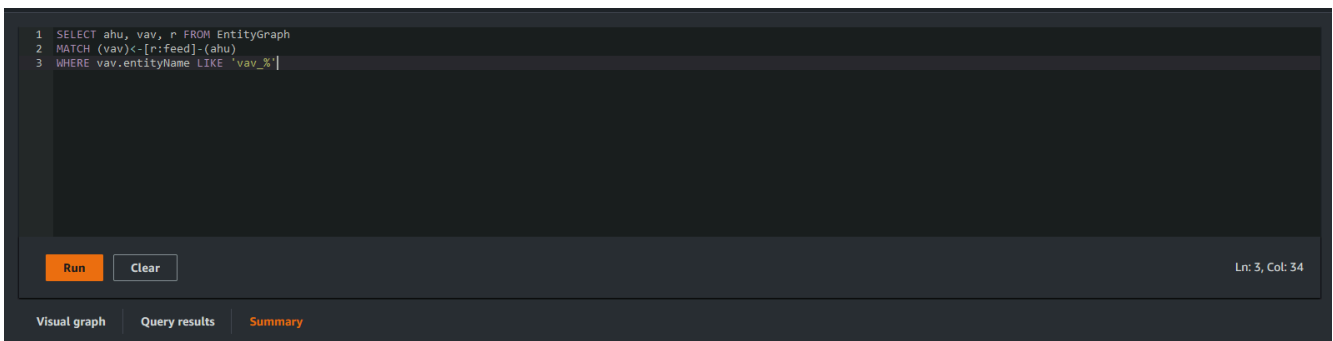
Pour accéder à l'éditeur de requêtes Knowledge Graph

1. Ouvrez la [AWS IoT TwinMaker console](#).
2. Ouvrez l'espace de travail dans lequel vous souhaitez utiliser le Knowledge Graph.
3. Dans le menu de navigation de gauche, choisissez l'éditeur de requêtes.
4. L'éditeur de requêtes s'ouvre. Vous êtes maintenant prêt à exécuter des requêtes sur les ressources de votre espace de travail.

## Exécuter une requête

Pour exécuter une requête et générer un graphique

1. Dans l'éditeur de requêtes, choisissez l'onglet Éditeur pour ouvrir l'éditeur de syntaxe.
2. Dans l'espace éditeur, écrivez la requête que vous souhaitez exécuter sur les ressources de votre espace de travail.



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

Run Clear Ln: 3, Col: 34

Visual graph Query results Summary

Dans l'exemple illustré, la demande recherche les entités dont le nom est indiqué, puis organise ces entités `vav_%` en fonction de la `feed` relation entre elles, à l'aide du code suivant.

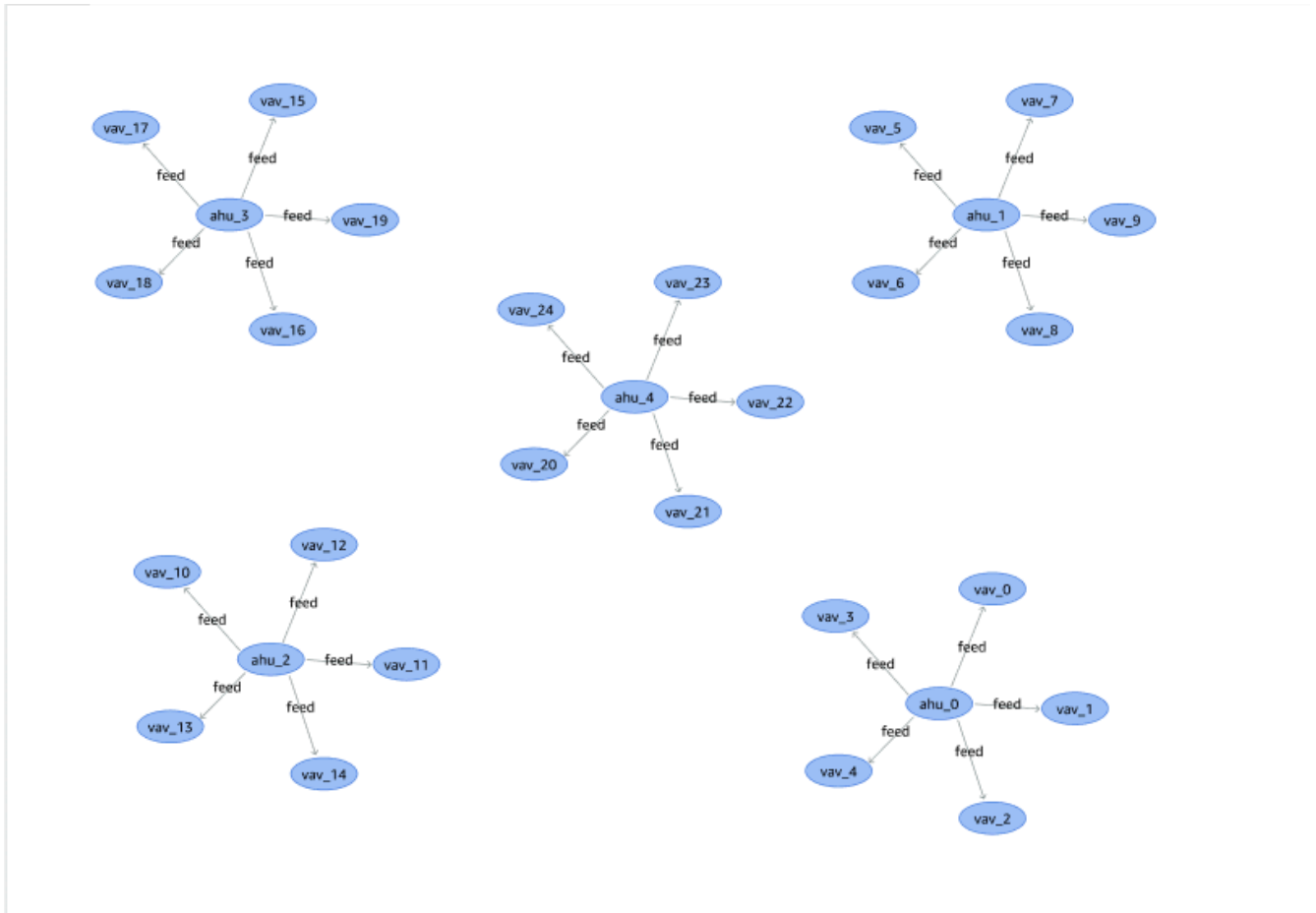
```
SELECT ahu, vav, r FROM EntityGraph
MATCH (vav)<-[:feed]-(ahu)
WHERE vav.entityName LIKE 'vav_%'
```

**Note**

La syntaxe du graphe de connaissances utilise [partiQL](#). Pour plus d'informations sur cette syntaxe, consultez [AWS IoT TwinMaker Knowledge Graph \(ressources supplémentaires\)](#).

3. Choisissez Exécuter la requête pour exécuter la demande que vous avez créée.

Un graphique est généré en fonction de votre demande.



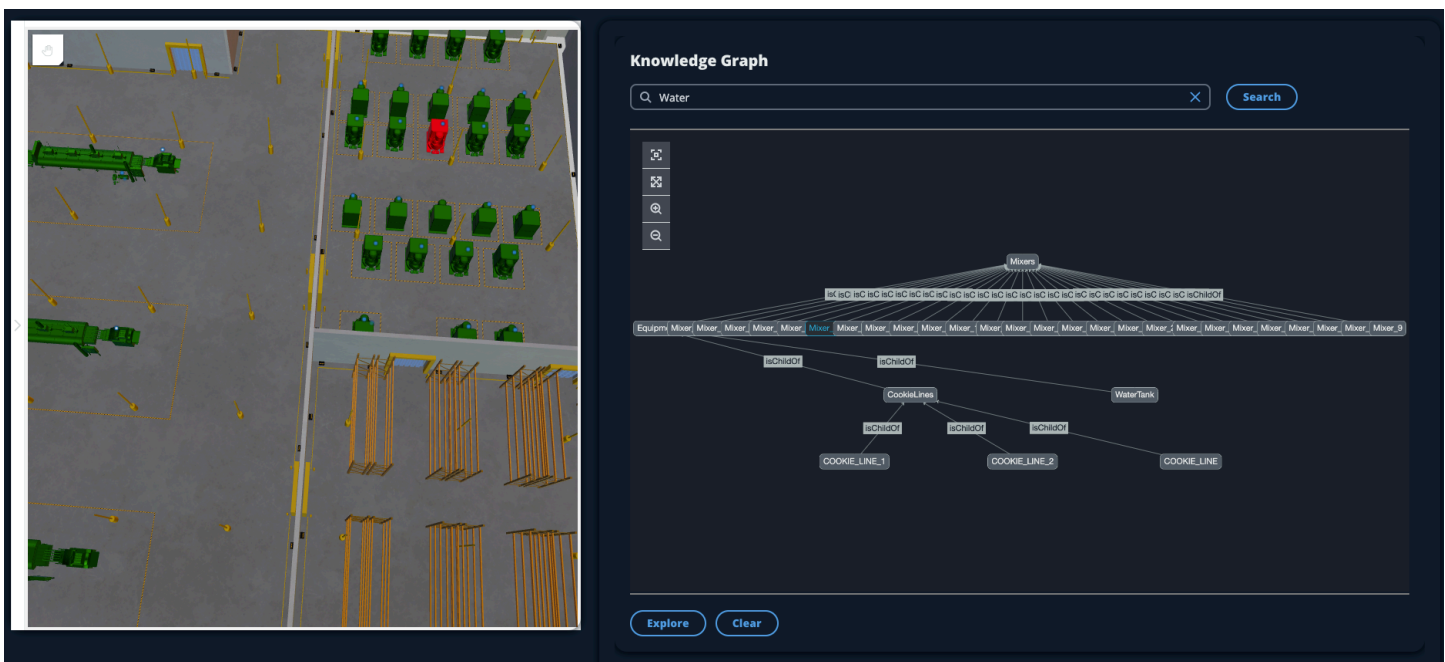
L'exemple de graphique présenté ci-dessus est basé sur l'exemple de requête de l'étape 2.

4. Les résultats de la requête sont également présentés dans une liste. Choisissez les résultats pour afficher les résultats de la requête sous forme de liste.
5. Choisissez éventuellement Exporter sous pour exporter les résultats de la requête au format JSON ou CSV.

Cela couvre l'utilisation de base du Knowledge Graph dans la console. Pour plus d'informations et des exemples illustrant la syntaxe du Knowledge Graph, consultez [AWS IoT TwinMaker Knowledge Graph \(ressources supplémentaires\)](#).

## Intégration de scènes de graphes de connaissances

Vous pouvez utiliser les composants du kit d' AWS IoT applications pour créer une application Web qui intègre le Knowledge Graph dans vos AWS IoT TwinMaker scènes. Cela vous permet de générer des graphiques basés sur les nœuds 3D (les modèles 3D qui représentent votre équipement ou vos systèmes) présents dans votre scène. Pour créer une application qui représente graphiquement des nœuds 3D à partir de votre scène, liez d'abord les nœuds 3D aux entités de votre espace de travail. Ce mappage permet de AWS IoT TwinMaker représenter graphiquement les relations entre les modèles 3D présents dans votre scène et les entités de votre espace de travail. Vous pouvez ensuite créer une application Web, sélectionner des modèles 3D avec votre scène et explorer leurs relations avec d'autres entités sous forme de graphique.



Pour un exemple d'application Web fonctionnelle qui utilise les composants du kit d' AWS IoT applications pour générer des graphiques dans une AWS IoT TwinMaker scène, consultez l'[AWS IoT TwinMaker exemple d'application React](#) sur github.

## AWS IoT TwinMaker prérequis pour les graphes de scène

Avant de créer une application Web qui utilise le AWS IoT TwinMaker Knowledge Graph dans vos scènes, remplissez les conditions préalables suivantes :

- Créez un AWS IoT TwinMaker espace de travail. Vous pouvez créer un espace de travail dans la [AWS IoT TwinMaker console](#).
- Familiarisez-vous avec AWS IoT TwinMaker le système de composants d'entités et apprenez à créer des entités. Pour de plus amples informations, veuillez consulter [Créez votre première entité](#).
- Créez une AWS IoT TwinMaker scène remplie de modèles 3D.
- Familiarisez-vous avec AWS IoT TwinMaker les composants du kit d' AWS IoT applications. Pour plus d'informations sur les AWS IoT TwinMaker composants, consultez [Création d'une application Web personnalisée à l'aide des composants de l' AWS IoT TwinMaker interface utilisateur](#).
- Familiarisez-vous avec les concepts du graphe de connaissances et la terminologie clé. Consultez [AWS IoT TwinMaker concepts de base du graphe de connaissances](#).

### Note

Pour utiliser le AWS IoT TwinMaker Knowledge Graph et les fonctionnalités associées, vous devez utiliser le mode de tarification des offres groupées standard ou échelonné. Pour plus d'informations sur AWS IoT TwinMaker les tarifs, consultez [Changer de mode de AWS IoT TwinMaker tarification](#).

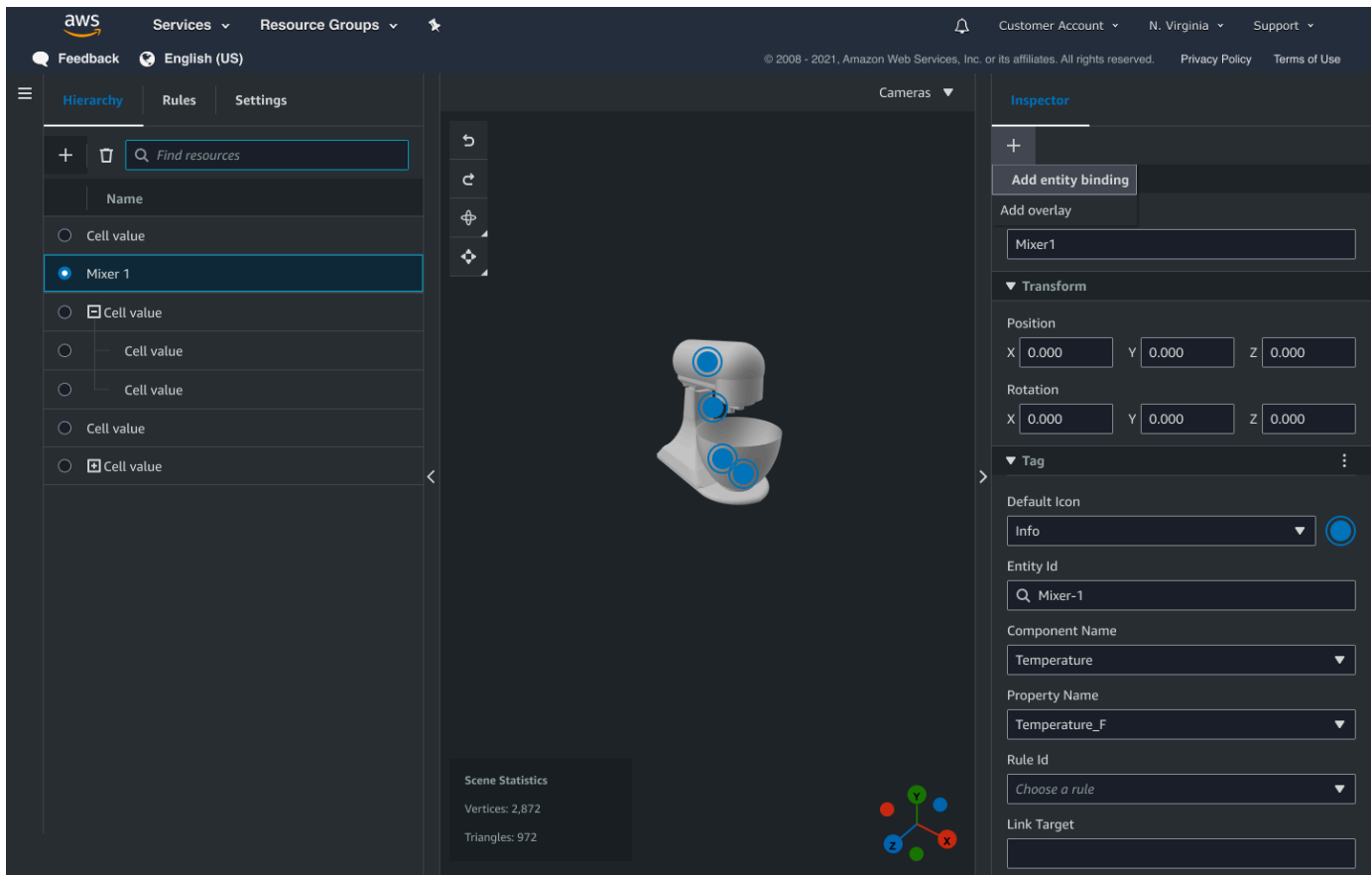
## Liez des nœuds 3D dans votre scène

Avant de créer une application Web qui intègre le graphe de connaissances à votre scène, liez les modèles 3D, appelés nœuds 3D, présents dans votre scène à l'entité d'espace de travail associée. Par exemple, si vous avez un modèle d'équipement de mixage dans une scène et qu'une entité correspondante est appelée `mixer_0`, créez une liaison de données entre le modèle du mélangeur et l'entité représentant le mélangeur, afin que le modèle et l'entité puissent être représentés graphiquement.

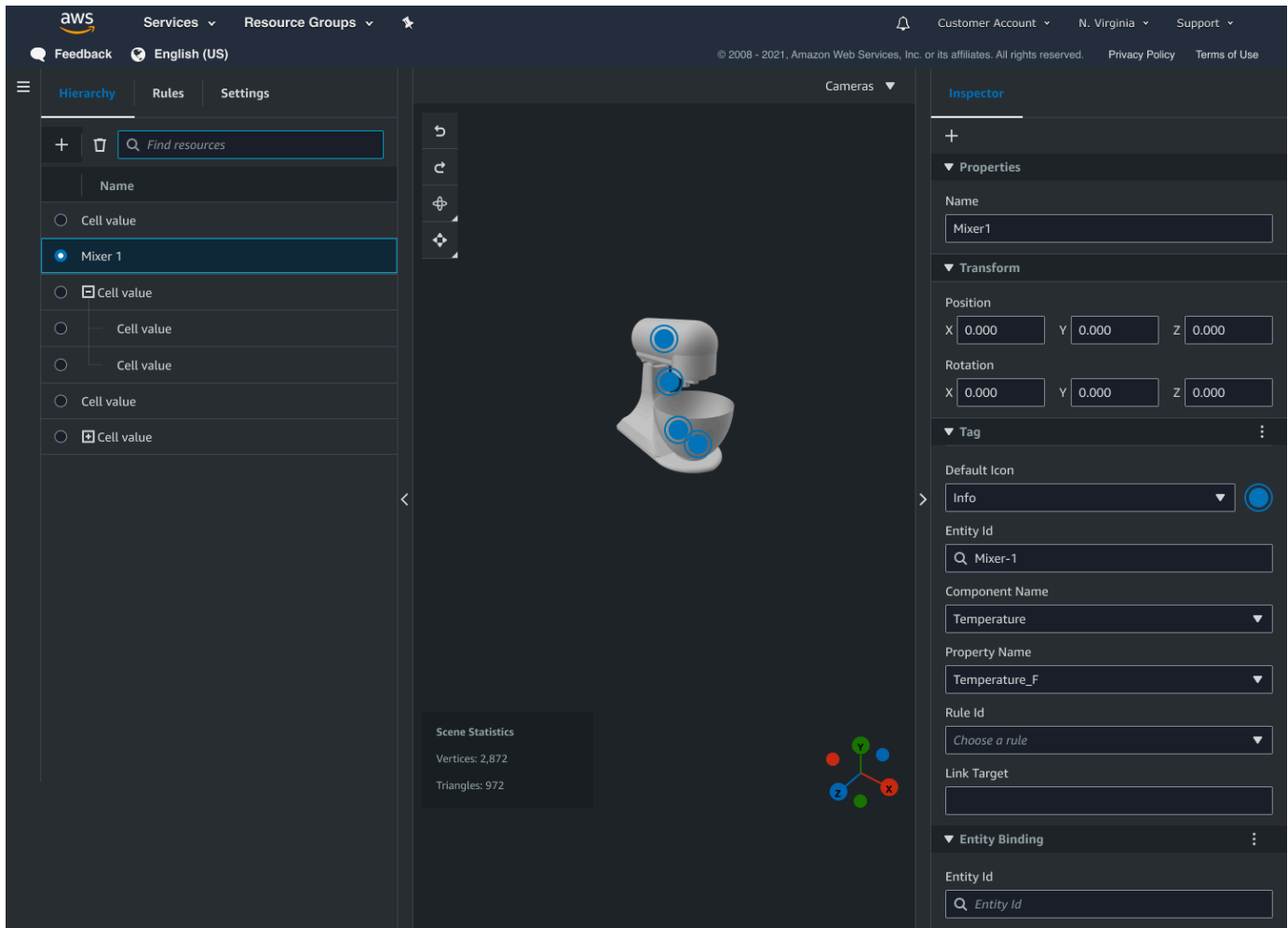
Pour effectuer une action de liaison de données

1. Connectez-vous à la [console AWS IoT TwinMaker](#).

2. Ouvrez votre espace de travail et sélectionnez une scène contenant les nœuds 3D que vous souhaitez lier.
3. Sélectionnez un nœud (modèle 3D) dans le compositeur de scène. Lorsque vous sélectionnez un nœud, il ouvre un panneau d'inspection sur le côté droit de l'écran.
4. Dans le panneau de l'inspecteur, naviguez vers le haut du panneau et sélectionnez le bouton +. Choisissez ensuite l'option Ajouter une liaison d'entité. Cela ouvrira une liste déroulante dans laquelle vous pourrez sélectionner une entité à lier au nœud actuellement sélectionné.



5. Dans le menu déroulant de liaison de données, sélectionnez l'identifiant d'entité que vous souhaitez associer au modèle 3D. Dans les champs Nom du composant et Nom de la propriété, sélectionnez les composants et les propriétés que vous souhaitez lier.



Une fois que vous avez sélectionné les champs Identifiant de l'entité, Nom du composant et Nom de propriété, la liaison est terminée.

- Répétez ce processus pour tous les modèles et entités que vous souhaitez représenter graphiquement.

#### Note

La même opération de liaison de données peut être effectuée sur les balises de votre scène. Il suffit de sélectionner une balise au lieu d'une entité et de suivre le même processus pour lier la balise à un nœud.

## Création d'une application Web

Après avoir lié vos entités, utilisez la bibliothèque du kit d' AWS IoT applications pour créer une application Web avec un widget de graphe de connaissances qui vous permet de visualiser votre scène et d'explorer les relations entre les nœuds de scène et les entités.

Utilisez les ressources suivantes pour créer votre propre application :

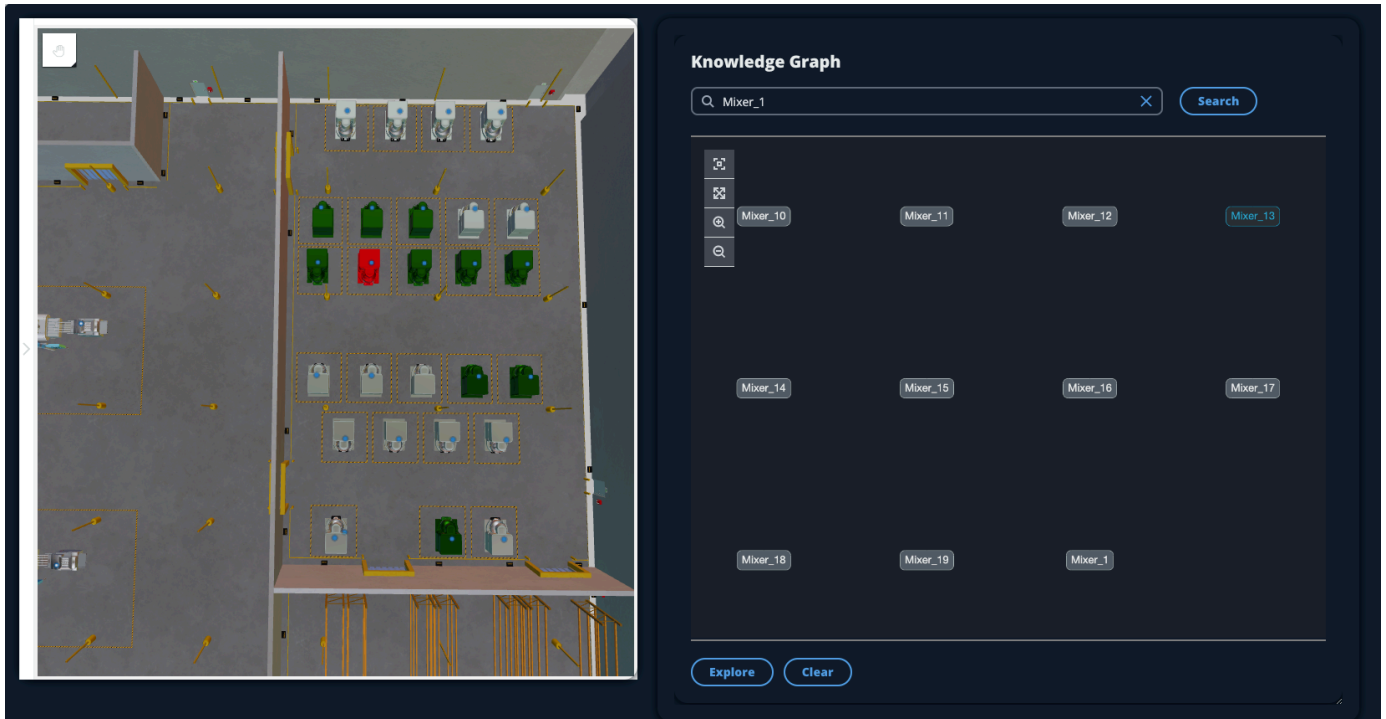
- AWS IoT TwinMaker Exemple de documentation [Readme de l'application React](#) sur github.
- L' AWS IoT TwinMaker exemple de [source](#) de l'application React sur github.
- Le kit d' AWS IoT applications : documentation de [démarrage](#).
- La documentation des [composants du lecteur vidéo du](#) kit d' AWS IoT applications.
- La documentation des [composants du kit d' AWS IoT applications Scene Viewer](#).

La procédure suivante illustre les fonctionnalités du composant Scene Viewer dans une application Web.

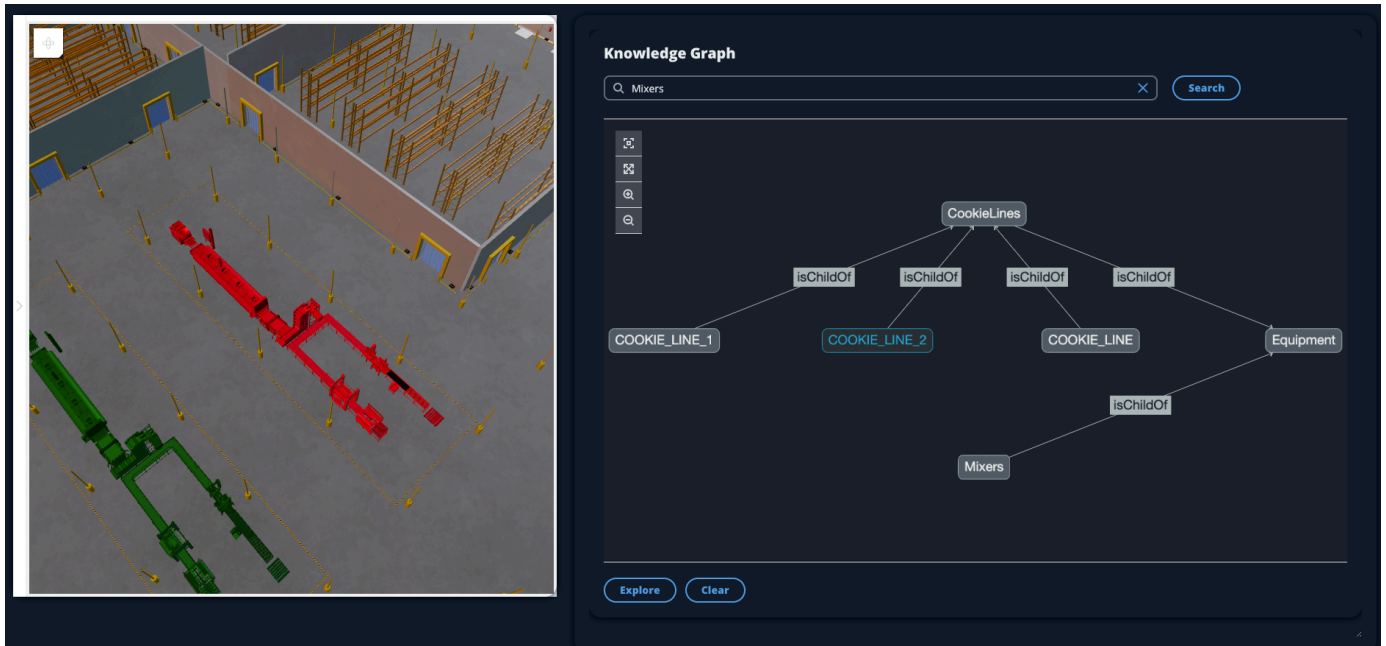
### Note

Cette procédure est basée sur l'implémentation du composant AWS IoT App Kit Scene Viewer dans l' AWS IoT TwinMaker exemple d'application React.

1. Ouvrez le composant Scene Viewer de l' AWS IoT TwinMaker exemple d'application React. Dans le champ de recherche, saisissez un nom d'entité ou un nom d'entité partiel (recherche distinguant majuscules et minuscules), puis sélectionnez le bouton Rechercher. Si un modèle est lié à l'identifiant de l'entité, le modèle de la scène sera mis en surbrillance et un nœud de l'entité sera affiché dans le panneau de visualisation de scène.



2. Pour générer un graphique de toutes les relations, sélectionnez un nœud dans le widget Scene Viewer et cliquez sur le bouton Explorer.



3. Appuyez sur le bouton Effacer pour effacer votre sélection de graphes actuelle et recommencer.

# Comment utiliser le AWS IoT TwinMaker Knowledge Graph avec Grafana

Cette section vous montre comment ajouter un panneau d'éditeur de requêtes à votre tableau de bord AWS IoT TwinMaker Grafana pour exécuter et afficher des requêtes.

## AWS IoT TwinMaker prérequis pour l'éditeur de requêtes

Avant d'utiliser le graphe de AWS IoT TwinMaker connaissances dans Grafana, remplissez les conditions préalables suivantes :

- Créez un AWS IoT TwinMaker espace de travail. Vous pouvez créer un espace de travail dans la [AWS IoT TwinMaker console](#).
- Configurez AWS IoT TwinMaker pour une utilisation avec Grafana. Pour obtenir des instructions, veuillez consulter [AWS IoT TwinMaker Intégration au tableau de bord Grafana](#).

### Note

Pour utiliser le AWS IoT TwinMaker Knowledge Graph, vous devez utiliser le mode de tarification des offres groupées standard ou échelonné. Pour de plus amples informations, veuillez consulter [Changer de mode de AWS IoT TwinMaker tarification](#).

## AWS IoT TwinMaker autorisations de l'éditeur de requêtes

Pour utiliser l'éditeur de AWS IoT TwinMaker requêtes dans Grafana, vous devez avoir un rôle IAM autorisé à effectuer l'action. `iottwinmaker:ExecuteQuery` Ajoutez cette autorisation à votre rôle de tableau de bord de l'espace de travail, comme illustré dans cet exemple :

JSON

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

```

        "s3:GetObject"
    ],
    "Resource": [
        "arn:aws:s3:::amzn-s3-demo-bucket",
        "arn:aws:s3:::amzn-s3-demo-bucket/*"
    ]
},
{
    "Effect": "Allow",
    "Action": [
        "iottwinmaker:GetEntity",
        "iottwinmaker:ListEntities",
        "iottwinmaker:ExecuteQuery"
    ],
    "Resource": [
        "arn:aws:iottwinmaker:us-
east-2:111122223333:workspace/workspaceId",
        "arn:aws:iottwinmaker:us-
east-2:111122223333:workspace/workspaceId/*"
    ]
},
{
    "Effect": "Allow",
    "Action": "iottwinmaker:ListWorkspaces",
    "Resource": "*"
}
]
}

```

### Note

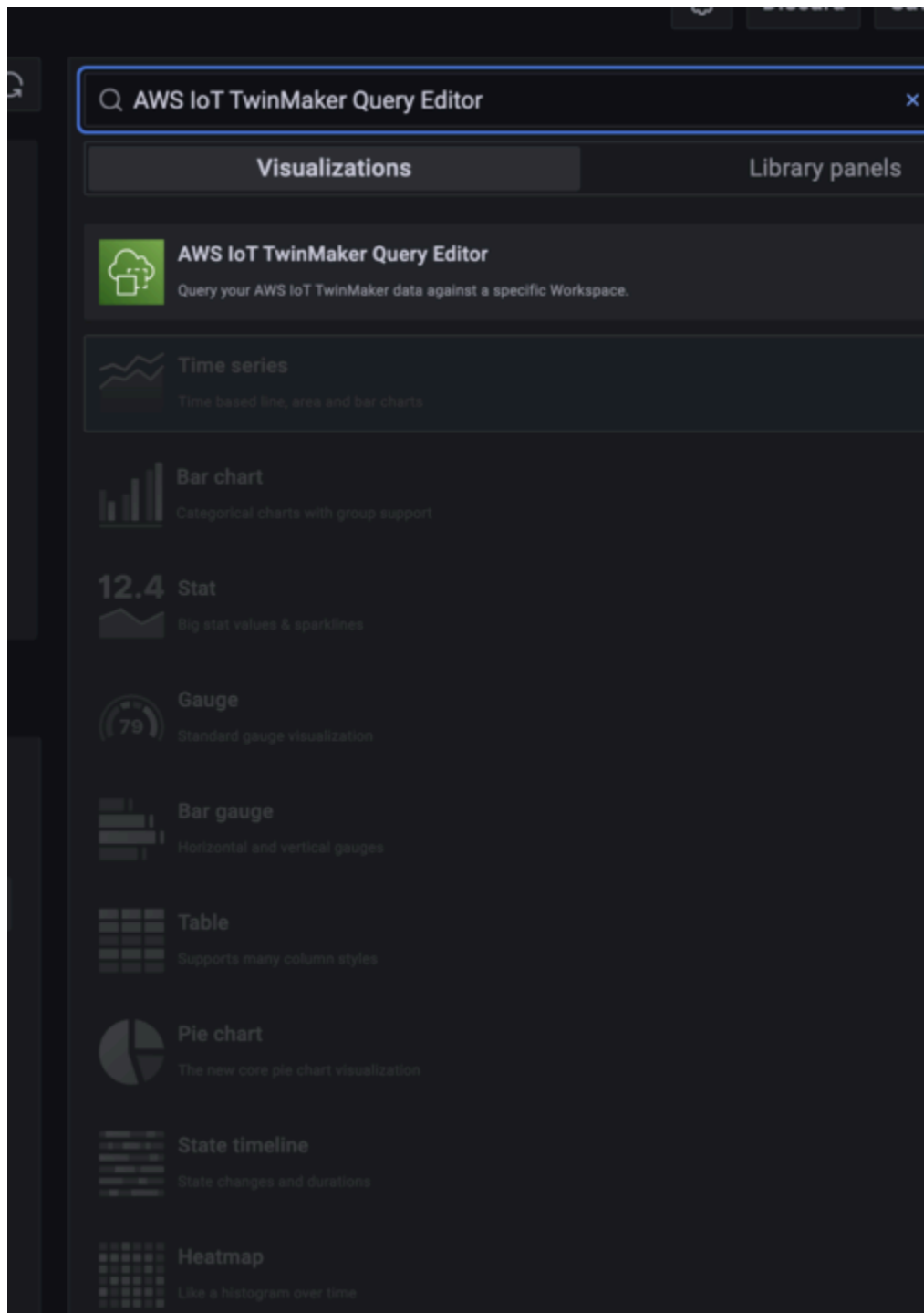
Lorsque vous configurez votre source de données AWS IoT TwinMaker Grafana, assurez-vous d'utiliser le rôle avec cette autorisation pour le champ ARN Assumer le rôle. Après l'avoir ajouté, vous pouvez sélectionner votre espace de travail dans le menu déroulant situé à côté de Workspace.

Pour de plus amples informations, veuillez consulter [Création d'un rôle IAM dans le tableau de bord](#).

## Configuration du panneau de l'éditeur de AWS IoT TwinMaker requêtes

Pour configurer un nouveau panneau de tableau de bord Grafana pour Knowledge Graph

1. Ouvrez votre tableau de AWS IoT TwinMaker bord Grafana.
2. Créez un nouveau panneau de tableau de bord. Pour connaître les étapes détaillées de création d'un panneau, voir [Création d'un tableau de bord](#) dans la documentation de Grafana.
3. Dans la liste des visualisations, sélectionnez Éditeur de AWS IoT TwinMaker requêtes.



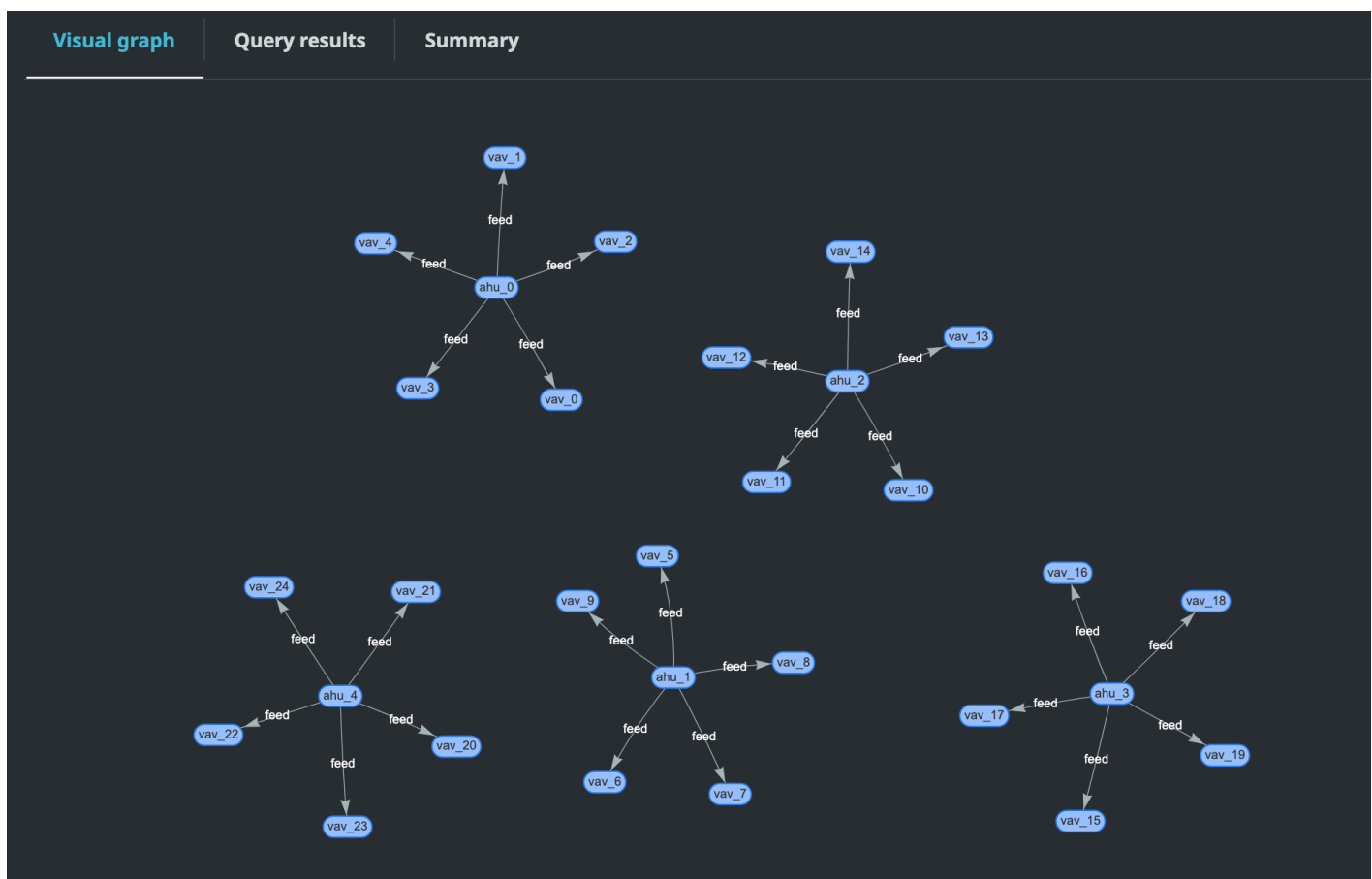
4. Sélectionnez la source de données sur laquelle exécuter les requêtes.
5. (Facultatif) Ajoutez un nom pour le nouveau panneau dans le champ prévu à cet effet.
6. Sélectionnez Appliquer pour enregistrer et confirmer votre nouveau panneau.

Le panneau Knowledge Graph fonctionne de la même manière que l'éditeur de requêtes fourni dans la AWS IoT TwinMaker console. Vous pouvez exécuter, écrire et effacer les requêtes que vous effectuez dans le panneau. Pour plus d'informations sur la façon d'écrire des requêtes, consultez [AWS IoT TwinMaker Knowledge Graph \(ressources supplémentaires\)](#).

### Comment utiliser l'éditeur AWS IoT TwinMaker de requêtes

Les résultats de vos requêtes sont affichés de trois manières, comme le montrent les images suivantes : visualisés sous forme de graphique, répertoriés dans un tableau ou présentés sous forme de résumé d'exécution.

- Visualisation graphique :



Le graphique visuel affiche uniquement les données des requêtes dont le résultat contient au moins une relation. Le graphique affiche les entités sous forme de nœuds et les relations sous forme d'arêtes orientées dans le graphe.

- Données tabulaires :

Visual graph
Query results
Summary

**Results returned (25)**

Export as ▼

< 1 >
⚙️

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

Le format de données tabulaire affiche les données de toutes les requêtes. Vous pouvez rechercher dans le tableau des résultats spécifiques ou des sous-ensembles de résultats. Les données peuvent être exportées au format JSON ou CSV.

- Résumé de l'exécution

| 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 |

Le résumé de l'exécution affiche la requête et les métadonnées relatives à l'état de la requête.

# AWS IoT TwinMaker Knowledge Graph (ressources supplémentaires)

Cette section fournit des exemples de base de la syntaxe partiQL utilisée pour écrire des requêtes dans le graphe de connaissances, ainsi que des liens vers la documentation partiQL qui fournit des informations sur le modèle de données du graphe de connaissances.

- [Documentation sur les modèles de données graphiques PartiQL](#)
- [Documentation sur les requêtes graphiques PartiQL](#)

Cet ensemble d'exemples montre les requêtes de base avec leurs réponses. Utilisez-le comme référence pour rédiger vos propres requêtes.

## Requêtes de base

- Obtenir toutes les entités à l'aide d'un filtre

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

Cette requête renvoie toutes les entités d'un espace de travail portant le nom `room_0`.

**FROM clause :** `EntityGraph` est la collection de graphes qui contient toutes les entités et leurs relations dans un espace de travail. Cette collection est automatiquement créée et gérée en AWS IoT TwinMaker fonction des entités de votre espace de travail.

**MATCH clause :** spécifie un modèle qui correspond à une partie du graphique. Dans ce cas, le modèle (`entity`) correspond à chaque nœud du graphe et est lié à la variable d'entité. La **FROM** clause doit être suivie de la **MATCH** clause.

**WHERE clause :** spécifie un filtre sur le `entityName` champ du nœud, où la valeur doit correspondre `room_0`.

**SELECT clause :** spécifie la `entity` variable afin que l'ensemble du nœud de l'entité soit renvoyé.

Réponse :

```
{
  "columnDescriptions": [
    {
      "name": "entity",
      "type": "NODE"
    }
  ],
  "rows": [
    {
      "rowData": [
        {
          "arn": "arn:aws:iottwinmaker:us-east-1: 577476956029: workspace /
SmartBuilding8292022 / entity / room_18f3ef90 - 7197 - 53 d1 - abab -
db9c9ad02781 ",
          "creationDate": 1661811123914,
          "entityId": "room_18f3ef90-7197-53d1-abab-db9c9ad02781",
          "entityName": "room_0",
          "lastUpdateDate": 1661811125072,
          "workspaceId": "SmartBuilding8292022",
          "description": "",
          "components": [
            {
              "componentName": "RoomComponent",
              "componentTypeId": "com.example.query.construction.room",
              "properties": [
                {
                  "propertyName": "roomFunction",
                  "propertyValue": "meeting"
                },
                {
                  "propertyName": "roomNumber",
                  "propertyValue": 0
                }
              ]
            }
          ]
        }
      ]
    }
  ]
}
```

`columnDescriptions` Renvoie les métadonnées relatives à la colonne, telles que le nom et le type. Le type renvoyé est `NODE`. Cela indique que le nœud entier a été renvoyé. Les autres valeurs du type peuvent être `EDGE` indiquer une relation ou `VALUE` une valeur scalaire telle qu'un entier ou une chaîne.

`rows` Renvoie une liste de lignes. Comme une seule entité a été mise en correspondance, `rowData` a une entité contenant tous les champs d'une entité est renvoyée.

#### Note

Contrairement au SQL où vous ne pouvez renvoyer que des valeurs scalaires, vous pouvez renvoyer un objet (au format JSON) en utilisant partiQL.

Chaque nœud contient tous les champs au niveau de l'entité tels que `entityId`, `arn` et les champs au niveau des composants tels que `components`, et, ainsi que les champs au niveau de la propriété tels que `componentTypeId` et `componentNameproperties`, le tout sous forme de JSON `propertyName` `propertyValue` imbriqué.

- Obtenez toutes les relations à l'aide d'un filtre :

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

Cette requête renvoie toutes les relations d'un espace de travail avec le nom de la relation `isLocationOf`.

La `MATCH` clause : spécifie un modèle qui correspond à deux nœuds (indiqués par `()`) connectés par une arête dirigée (indiquée par `-[]->`) et liés à une variable appelée `relationship`.

La `WHERE` clause : spécifie un filtre sur le `relationshipName` champ de l'arête, où se trouve la valeur `isLocationOf`.

La `SELECT` clause : spécifie la variable de relation afin que l'ensemble du nœud périphérique soit renvoyé.

Réponse

```

{
  "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
]
}

```

Le type de colonne dans `columnDescriptions` est un `EDGE`.

Chacun `rowData` représente une arête avec des champs tels que `relationshipName`. Il est identique au nom de propriété de relation défini sur l'entité. `sourceEntityId`, `sourceComponentName` et `sourceComponentTypeId` fournissent des informations sur l'entité et le composant sur lesquels la propriété de relation a été définie. `targetEntityId` spécifie l'entité vers laquelle cette relation pointe.

- Obtenez toutes les entités ayant une relation spécifique avec une entité spécifique

```

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

```

Cette requête renvoie tous les noms d'entité de toutes les entités ayant une `isLocationOf` relation avec l'`room_0` entité.

La `MATCH` clause : spécifie un modèle qui correspond à deux nœuds (`e1`, `e2`) ayant une arête dirigée (`r`).

La `WHERE` clause : spécifie un filtre sur le nom de la relation et le nom de l'entité source.

La SELECT clause : renvoie le `entityName` champ dans le e2 nœud.

## Réponse

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

Dans les `ColumnDescriptions`, le type de la colonne est `VALUE` puisque `c'entityName` est une chaîne.

Une entité est renvoyée. `floor_0`

## CORRESPONDRE

Les modèles suivants sont pris en charge dans une `MATCH` clause :

- Faites correspondre le nœud « b » pointant vers le nœud « a » :

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

- Faites correspondre le nœud « a » pointant vers le nœud « b » :

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

Aucune variable n'est liée à une relation en supposant qu'il n'est pas nécessaire de spécifier un filtre sur la relation.

- Faites correspondre le nœud « a » pointant vers le nœud « b » et le nœud « b » pointant vers le nœud « a » :

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

Cela renverra deux correspondances : une de « a » à « b » et une autre de « b » à « a ». Il est donc recommandé d'utiliser des arêtes orientées dans la mesure du possible.

- Le nom de la relation est également une étiquette du graphe EntityGraph de propriétés. Vous pouvez donc simplement spécifier le nom de la relation après deux points (:) au lieu de spécifier un filtre `rel.relationshipName` dans la WHERE clause.

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

- Enchaînement : les modèles peuvent être enchaînés pour correspondre à plusieurs relations.

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

- Les modèles de sauts variables peuvent également s'étendre sur plusieurs nœuds et arêtes :

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

Cette requête correspond à n'importe quel modèle avec des arêtes sortantes du nœud « a » entre 1 et 5 sauts. Les quantificateurs autorisés sont les suivants :

`{m, n}`- entre m et n répétitions

`{m, }`- 5 répétitions ou plus.

## À PARTIR DE :

Un nœud d'entité peut contenir des données imbriquées, telles que des composants qui contiennent eux-mêmes d'autres données imbriquées telles que des propriétés. Vous pouvez y accéder en démêlant le résultat du modèle MATCH.

```
SELECT e
FROM EntityGraph MATCH (e), e.components AS c, c.properties AS p
WHERE c.componentTypeId = 'com.example.query.construction.room',
AND p.propertyName = 'roomFunction'
AND p.propertyValue = 'meeting'
```

Accédez aux champs imbriqués en pointant des points `.` dans une variable. Une virgule (,) est utilisée pour désimbriquer (ou joindre) des entités contenant les composants, puis les

propriétés contenues dans ces composants. A est utilisé pour lier une variable aux variables non imbriquées afin qu'elles puissent être utilisées dans les clauses WHERE orSELECT. Cette requête renvoie toutes les entités qui contiennent une propriété nommée roomFunction avec une valeur meeting dans un composant avec un identifiant de type de composant com.example.query.construction.room

Pour accéder à plusieurs champs imbriqués d'un champ, tels que plusieurs composants d'une entité, utilisez la notation par virgule pour effectuer une jointure.

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

### SÉLECTIONNEZ :

- Renvoie un nœud :

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

- Renvoie une arête :

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

- Renvoie une valeur scalaire :

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

Formatez le nom du champ de sortie en lui attribuant un alias en utilisant AS. Ici, au lieu d'propertyValue un nom de colonne dans la réponse, roomfunction il est renvoyé.

- Alias de retour :

```
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%'
```

Il est fortement recommandé d'utiliser des alias pour être explicite, améliorer la lisibilité et éviter toute ambiguïté dans vos requêtes.

OÙ :

- Les opérateurs logiques pris en charge sont ANDNOT, etOR.
- Les opérateurs de comparaison pris en charge sont < <=>,=>,, et!=.
- Utilisez le IN mot clé si vous souhaitez spécifier plusieurs OR conditions dans le même champ.
- Filtrez sur un champ d'entité, de composant ou de propriété :

```
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'
```

- Filtrez sur la configuration propriété. unitVoici la clé de la carte de configuration et Celsius la valeur.

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

- Vérifiez si une propriété de carte contient une clé et une valeur données :

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

- Vérifiez si une propriété de carte contient une clé donnée :

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

- Vérifiez si une propriété de liste contient une valeur donnée :

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

- Utilisez cette lower() fonction pour les comparaisons sans distinction majuscules/majuscules. Par défaut, toutes les comparaisons font la distinction majuscules/minuscules.

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

## COMME :

Utile si vous ne connaissez pas la valeur exacte d'un champ et que vous pouvez effectuer une recherche en texte intégral sur le champ spécifié. %représente zéro ou plus.

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

- Recherche d'infixes : %room%
- Recherche par préfixe : room%
- Recherche de suffixes : %room
- Si vous avez « % » dans vos valeurs, insérez un caractère d'échappement dans LIKE et spécifiez le caractère d'échappement avecESCAPE.

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

## DISTINCT :

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

- Le DISTINCT mot clé élimine les doublons du résultat final.

DISTINCTn'est pas pris en charge sur les types de données complexes.

## COMPTE :

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

- Le COUNT mot clé calcule le nombre d'éléments dans le résultat d'une requête.
- COUNTn'est pas pris en charge sur les champs complexes imbriqués et les champs de modèle graphique.
- COUNTl'agrégation n'est pas prise en charge par DISTINCT les requêtes imbriquées.

Par exemple, COUNT(DISTINCT e.entityId) n'est pas pris en charge.

## CHEMIN

Les projections de modèles suivantes sont prises en charge lors des requêtes utilisant la projection de trajectoire :

- Requêtes à sauts variables

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

Cette requête fait correspondre et projette les métadonnées des nœuds de tous les modèles présentant des bords sortants depuis le nœud a en 1 à 3 sauts.

- Requêtes à saut fixe

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

Cette requête fait correspondre et projette les métadonnées des entités et des arêtes entrantes vers b.

- Requêtes non dirigées

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

Cette requête correspond et projette les métadonnées des nœuds dans des modèles à 1 saut reliant a et c via b.

```
{
  "columnDescriptions": [
    {
      "name": "path",
      "type": "PATH"
    }
  ],
  "rows": [
    {
      "rowData": [
        {
          "path": [
            {
              "entityId": "a",
              "entityName": "a"
            },
            {
              "relationshipName": "a-to-b-relation",
              "sourceEntityId": "a",
              "targetEntityId": "b"
            }
          ]
        }
      ]
    }
  ]
}
```

```

    {
      "entityId": "b",
      "entityName": "b"
    }
  ]
}
],
{
  "rowData": [
    {
      "path": [
        {
          "entityId": "b",
          "entityName": "b"
        },
        {
          "relationshipName": "b-to-c-relation",
          "sourceEntityId": "b",
          "targetEntityId": "c"
        },
        {
          "entityId": "c",
          "entityName": "c"
        }
      ]
    }
  ]
}
]
}

```

Cette réponse à la PATH requête comprend uniquement des métadonnées qui identifient tous les nœuds et les arêtes de chacun path/pattern entre a et c via b.

#### LIMITE et DÉCALAGE :

```

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

```

LIMIT indique le nombre de résultats à renvoyer dans la requête, ainsi OFFSET que le nombre de résultats à ignorer.

LIMIT et MaxResults :

L'exemple suivant montre une requête qui renvoie 500 résultats au total, mais qui n'en affiche que 50 à la fois par appel d'API. Ce modèle peut être utilisé lorsque vous devez limiter le nombre de résultats affichés, par exemple si vous ne pouvez afficher que 50 résultats dans une interface utilisateur.

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

- Le LIMIT mot clé affecte la requête et limite les lignes qui en résultent. Si vous devez contrôler le nombre de résultats renvoyés par appel d'API sans limiter le nombre total de résultats renvoyés, utilisez LIMIT.
- max-result est un paramètre facultatif pour l'[action d'ExecuteQuery API](#). max-result s'applique uniquement à l'API et à la façon dont les résultats sont lus dans les limites de la requête ci-dessus.

L'utilisation max-results dans une requête vous permet de réduire le nombre de résultats affichés sans limiter le nombre réel de résultats renvoyés.

La requête ci-dessous passe par la page de résultats suivante. Cette requête utilise l'appel d'ExecuteQueryAPI pour renvoyer les lignes 51 à 100, où la page de résultats suivante est spécifiée par le next-token — dans ce cas, le jeton est : "H7kyGmvK376L"

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

- La next-token chaîne indique la page de résultats suivante. Pour plus d'informations, consultez l'action de [ExecuteQuery](#) l'API.

AWS IoT TwinMaker Les limites d'une requête Knowledge Graph sont les suivantes :

| Nom de la limite   | Quota       | Ajustable |
|--|-------------|-----------|
| Délai d'exécution de la requête  | 10 secondes | Non       |
| Nombre maximum de sauts  | 10          | Oui       |
| Nombre maximum de JOIN self-s  | 20          | Oui       |
| Nombre maximum de champs projetés  | 20          | Oui       |
| Nombre maximum d'expressions conditionnelles (AND,OR,NOT)  | 10          | Oui       |
| Longueur maximale d'un modèle d'LIKEexpression (y compris les caractères génériques et les échappements) | 20          | Oui       |
| Nombre maximum d'éléments pouvant être spécifiés dans une IN clause                                      | 10          | Oui       |
| Valeur maximale pour OFFSET  | 3000        | Oui       |
| Valeur maximale pour LIMIT   | 3000        | Oui       |
| Valeur maximale pour les traversées (OFFSET+LIMIT)   | 3000        | Oui       |

# Synchronisation des actifs avec AWS IoT SiteWise

AWS IoT TwinMaker prend en charge la synchronisation des actifs (synchronisation des actifs) pour vos AWS IoT SiteWise actifs et modèles d'actifs. À l'aide du type de AWS IoT SiteWise composant, la synchronisation AWS IoT SiteWise des actifs prend les actifs et les modèles d'actifs existants et convertit ces ressources en AWS IoT TwinMaker entités, composants et types de composants. Les sections suivantes expliquent comment configurer la synchronisation des actifs et quels AWS IoT SiteWise actifs et modèles d'actifs peuvent être synchronisés avec votre AWS IoT TwinMaker espace de travail.

## Rubriques

- [Utilisation de la synchronisation des actifs avec AWS IoT SiteWise](#)
- [Différences entre les espaces de travail personnalisés et les espaces de travail par défaut](#)
- [Ressources synchronisées depuis AWS IoT SiteWise](#)
- [Analyser l'état et les erreurs de synchronisation](#)
- [Supprimer une tâche de synchronisation](#)
- [Limites de synchronisation des actifs](#)

## Utilisation de la synchronisation des actifs avec AWS IoT SiteWise

Cette rubrique explique comment activer et configurer la synchronisation AWS IoT SiteWise des actifs. Suivez les procédures appropriées en fonction du type d'espace de travail que vous utilisez.

### Important

Consultez [the section called “Différences entre les espaces de travail personnalisés et les espaces de travail par défaut”](#) pour plus d'informations sur les différences entre les espaces de travail personnalisés et par défaut.

## Rubriques

- [Utilisation d'un espace de travail personnalisé](#)
- [Utilisation de l'lo TSite WiseDefaultWorkspace](#)

## Utilisation d'un espace de travail personnalisé

Passez en revue ces conditions préalables avant d'activer la synchronisation des ressources.

### Conditions préalables

Avant toute utilisation AWS IoT SiteWise, vous devez effectuer les opérations suivantes :

- Vous disposez d'un AWS IoT TwinMaker espace de travail.
- Vous avez intégré des actifs et des modèles d'actifs AWS IoT SiteWise. Pour plus d'informations, consultez la section [Création de modèles d'actifs](#).
- Rôle IAM existant doté d'autorisations de lecture pour les AWS IoT SiteWise actions suivantes :
  - ListAssets
  - ListAssetModels
  - DescribeAsset
  - DescribeAssetModel
- Le rôle IAM doit disposer des autorisations d'écriture suivantes pour AWS IoT TwinMaker :
  - CreateEntity
  - UpdateEntity
  - DeleteEntity
  - CreateComponentType
  - UpdateComponentType
  - DeleteComponentType
  - ListEntities
  - GetEntity
  - ListComponentTypes

Utilisez le rôle IAM suivant comme modèle pour le rôle requis :

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

```

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

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

```

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

Utilisez la procédure suivante pour activer et configurer la synchronisation AWS IoT SiteWise des actifs.

1. Dans la [AWS IoT TwinMaker console](#), accédez à la page Paramètres.
2. Ouvrez l'onglet Sources du modèle.

## 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. Choisissez **Connect workspace** pour associer votre AWS IoT TwinMaker espace de travail à vos AWS IoT SiteWise actifs.

### Note

Vous ne pouvez utiliser la synchronisation des actifs qu'avec un seul AWS IoT TwinMaker espace de travail. Vous devez déconnecter la synchronisation d'un espace de travail et vous connecter à un autre espace de travail si vous souhaitez effectuer la synchronisation dans un autre espace de travail.

4. Accédez ensuite à l'espace de travail dans lequel vous souhaitez utiliser la synchronisation des actifs.
5. Choisissez **Add sources (Ajouter des sources)**. Cela ouvre la page source **Ajouter un modèle d'entité**.

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. Sur la page Ajouter une source de modèle d'entité, vérifiez que le champ source s'affiche AWS IoT SiteWise. Sélectionnez le rôle IAM que vous avez créé comme condition préalable au rôle IAM.
7. Vous avez maintenant activé la synchronisation AWS IoT SiteWise des actifs. Vous devriez voir une bannière de confirmation apparaître en haut de la page Workspace sélectionnée pour confirmer que la synchronisation des actifs est active. Vous devriez également voir une source de synchronisation répertoriée dans la section Sources du modèle d'entité.

**cookieFactory** Info View ▼ | Delete

---

**Workspace information** Edit

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

---

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

| Source           | Status                                      | Date last updated                |
|------------------|---|----------------------------------|
| AWS IoT SiteWise | <span style="color: green;">✔ Synced</span> | March 28, 2022, 14:32 (UTC+3:30) |

## Utilisation de l'lo TSite WiseDefaultWorkspace

Lorsque vous optez pour l'[AWS IoT SiteWiseAWS IoT TwinMaker intégration](#), un espace de travail par défaut nommé `IoTSiteWiseDefaultWorkspace` est créé et automatiquement synchronisé avec AWS IoT SiteWise.

Vous pouvez également utiliser l' `AWS IoT TwinMaker CreateWorkspaceAPI` pour créer un espace de travail nommé `IoTSiteWiseDefaultWorkspace`.

### Conditions préalables

Avant de créer `IoTSiteWiseDefaultWorkspace`, assurez-vous d'avoir effectué les opérations suivantes :

- Créez un rôle AWS IoT TwinMaker lié à un service. Pour plus d'informations, consultez [Utilisation de rôles liés à un service pour AWS IoT TwinMaker](#).
- Ouvrez la console IAM à l'adresse <https://console.aws.amazon.com/iam/>.

Passez en revue le rôle ou l'utilisateur et vérifiez qu'il est autorisé à `iotsitewise:EnableSiteWiseIntegration`.

Si nécessaire, ajoutez une autorisation au rôle ou à l'utilisateur :

JSON

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

## Différences entre les espaces de travail personnalisés et les espaces de travail par défaut

### Important

AWS IoT SiteWise Les nouvelles fonctionnalités, telles que [CompositionModel](#), ne sont disponibles que dans `IoTSiteWiseDefaultWorkspace`. Nous vous encourageons à utiliser un espace de travail par défaut plutôt qu'un espace de travail personnalisé.

Lorsque vous utilisez le `IoTSiteWiseDefaultWorkspace`, il existe quelques différences notables par rapport à l'utilisation d'un espace de travail personnalisé avec synchronisation des actifs.

- Lorsque vous créez un espace de travail par défaut, l'emplacement Amazon S3 et le rôle IAM sont facultatifs.

### Note

Vous pouvez l'utiliser `UpdateWorkspace` pour fournir l'emplacement Amazon S3 et le rôle IAM.

- Il `IoTSiteWiseDefaultWorkspace` n'y a pas de limite de nombre de AWS IoT SiteWise ressources avec laquelle synchroniser les ressources AWS IoT TwinMaker.
- Lorsque vous synchronisez des ressources depuis AWS IoT SiteWise, elles `SyncSource` le seront `SITWISE_MANAGED`. Cela inclut `Entities` et `ComponentTypes`.
- De nouvelles AWS IoT SiteWise fonctionnalités, telles que `CompositionModel` celles disponibles uniquement dans le `IoTSiteWiseDefaultWorkspace`.

Il existe quelques limitations spécifiques `IoTSiteWiseDefaultWorkspace`, à savoir :

- L'espace de travail par défaut ne peut pas être supprimé.
- Pour supprimer des ressources, vous devez d'abord supprimer les AWS IoT SiteWise ressources, puis les ressources correspondantes AWS IoT TwinMaker sont supprimées.

# Ressources synchronisées depuis AWS IoT SiteWise

Cette rubrique répertorie les actifs que vous pouvez synchroniser avec votre AWS IoT TwinMaker espace AWS IoT SiteWise de travail.

## Important

Consultez [Différences entre les espaces de travail personnalisés et les espaces de travail par défaut](#) pour plus d'informations sur les différences entre les espaces de travail personnalisés et par défaut.

## Espaces de travail personnalisés et par défaut

Les ressources suivantes sont synchronisées et disponibles dans les espaces de travail personnalisés et par défaut :

### Modèles d'actifs

AWS IoT TwinMaker crée un nouveau type de composant pour chaque modèle d'actif dans AWS IoT SiteWise.

- Le composant `TypeId` du modèle d'actifs utilisera l'un des modèles suivants :
  - Espace de travail personnalisé - `iotsitewise.assetmodel:assetModelId`
  - Espace de travail par défaut - `assetModelId`
- Chaque propriété du modèle d'actif est une nouvelle propriété du type de composant, avec l'un des modèles de dénomination suivants :
  - Espace de travail personnalisé - `Property_propertyId`
  - Espace de travail par défaut - `propertyId`

Le nom de propriété AWS IoT SiteWise est stocké tel que `displayName` dans la définition de la propriété.

- Chaque hiérarchie du modèle d'actif est une nouvelle propriété de type `LIST` et `nestedType` se trouve `RELATIONSHIP` dans le type de composant. La hiérarchie est mappée à la propriété avec un nom préfixé par l'un des éléments suivants :
  - Espace de travail personnalisé - `Hierarchy_hierarchyId`
  - Espace de travail par défaut - `hierarchyId`

## Actif

AWS IoT TwinMaker crée une nouvelle entité pour chaque actif dans AWS IoT SiteWise.

- `entityId` est la même chose que le `assetId` in AWS IoT SiteWise.
- Ces entités ont un seul composant appelé `siteWiseBase`, dont le type de composant correspond au modèle d'actif de cet actif.
- Toute dérogation au niveau de l'actif, telle que la définition d'un alias de propriété ou d'une unité de mesure, est reflétée dans l'entité dans AWS IoT TwinMaker.

## Espace de travail par défaut uniquement

Les ressources suivantes sont synchronisées et disponibles uniquement dans l'espace de travail par défaut, `IoTSiteWiseDefaultWorkspace`.

### AssetModelComponents

AWS IoT TwinMaker crée un nouveau type de composant pour chaque `AssetModelComponents` entrée AWS IoT SiteWise.

- Le composant `TypeId` du modèle d'actif utilise le modèle suivant : `assetModelId`.
- Chaque propriété du modèle d'actif est une nouvelle propriété du type de composant, dont le nom de propriété est `propertyId`. Le nom de propriété AWS IoT SiteWise est stocké tel que `displayName` dans la définition de la propriété.
- Chaque hiérarchie du modèle d'actif est une nouvelle propriété de type `LIST` et `nestedType` se trouve `RELATIONSHIP` dans le type de composant. La hiérarchie est mappée à la propriété avec un nom préfixé par `hierarchyId`.

### AssetModelCompositeModel

AWS IoT TwinMaker crée un nouveau type de composant pour chaque `AssetModelCompositeModel` entrée AWS IoT SiteWise.

- Le composant `TypeId` du modèle d'actif utilise le modèle suivant : `assetModelId_assetModelCompositeModelId`.
- Chaque propriété du modèle d'actif est une nouvelle propriété du type de composant, dont le nom de propriété est `propertyId`. Le nom de propriété AWS IoT SiteWise est stocké tel que `displayName` dans la définition de la propriété.

## AssetCompositeModels

AWS IoT TwinMaker crée un nouveau composant composite pour chaque `AssetCompositeModel` entrée AWS IoT SiteWise.

- `componentName` est la même chose que le `assetModelCompositeModelId` in AWS IoT SiteWise.

## Ressources non synchronisées

Les ressources suivantes ne sont pas synchronisées :

### Ressources et modèles d'actifs non synchronisés

- Les modèles d'alarme seront synchronisés en tant que modèles composites, mais les données correspondantes de l'actif relatives aux alarmes ne sont pas synchronisées.
- [AWS IoT SiteWise les flux de données](#) ne sont pas synchronisés. Seules les propriétés modélisées dans le modèle d'actifs sont synchronisées.
- Les valeurs des propriétés pour les attributs, les mesures, les transformations, les agrégats et le calcul des métadonnées, tels que les formules et les fenêtres, ne sont pas synchronisées. Seules les métadonnées relatives aux propriétés, telles que l'alias, l'unité de mesure et le type de données, sont synchronisées. Les valeurs peuvent être demandées à l'aide de l'API standard du connecteur de AWS IoT TwinMaker données, . [GetPropertyValueHistory](#)

## Utiliser des entités et des types de composants synchronisés dans AWS IoT TwinMaker

Une fois que les actifs sont synchronisés AWS IoT SiteWise, les types de composants synchronisés sont en lecture seule. AWS IoT TwinMaker Toute action de mise à jour ou de suppression doit être effectuée dans AWS IoT SiteWise, et ces modifications sont synchronisées AWS IoT TwinMaker si le `SyncJob` est toujours actif.

Les entités synchronisées et le composant AWS IoT SiteWise de base sont également en AWS IoT TwinMaker lecture seule. Vous pouvez ajouter des composants non synchronisés supplémentaires à l'entité synchronisée, à condition qu'aucun attribut au niveau de l'entité, tel que la description, ne soit mis à jour. `entityName`

Certaines restrictions s'appliquent à la manière dont vous pouvez interagir avec les entités synchronisées. Vous ne pouvez pas créer d'entités enfants sous une entité synchronisée dans la

hiérarchie de l'entité synchronisée. En outre, vous ne pouvez pas créer de types de composants non synchronisés qui s'étendent à partir d'un type de composant synchronisé.

#### Note

Les composants supplémentaires sont supprimés en même temps que l'entité si la ressource est supprimée dans AWS IoT SiteWise ou si vous supprimez la tâche de synchronisation.

Vous pouvez utiliser ces entités synchronisées dans les tableaux de bord Grafana et les ajouter sous forme de balises dans le compositeur de scène, comme les entités ordinaires. Vous pouvez également émettre des requêtes Knowledge Graph pour ces entités synchronisées.

#### Note

Les entités synchronisées sans modification ne sont pas facturées, mais ces entités vous sont facturées si des modifications ont été apportées. AWS IoT TwinMaker Par exemple, si vous ajoutez un composant non synchronisé à une entité synchronisée, cette entité est désormais facturée. AWS IoT TwinMaker Pour plus d'informations, consultez [Tarification d'AWS IoT TwinMaker](#).

## Analyser l'état et les erreurs de synchronisation

Cette rubrique fournit des conseils sur la manière d'analyser les erreurs et les statuts de synchronisation.

#### Important

Consultez [the section called “Différences entre les espaces de travail personnalisés et les espaces de travail par défaut”](#) pour plus d'informations sur les différences entre les espaces de travail personnalisés et par défaut.

## Synchroniser les statuts des tâches

Une tâche de synchronisation possède l'un des statuts suivants en fonction de son état.

- L'**CREATING** état de la tâche de synchronisation signifie que la tâche vérifie les autorisations et charge des données AWS IoT SiteWise pour préparer la synchronisation.
- L'**INITIALIZING** état de la tâche de synchronisation signifie que toutes les ressources existantes AWS IoT SiteWise sont synchronisées avec AWS IoT TwinMaker. Cette étape peut prendre plus de temps si l'utilisateur possède un grand nombre de ressources et de modèles d'actifs AWS IoT SiteWise. Vous pouvez contrôler le nombre de ressources synchronisées en vérifiant la tâche de synchronisation dans la [AWS IoT TwinMaker console](#) ou en appelant l'`ListSyncResourcesAPI`.
- L'**ACTIVE** état de la tâche de synchronisation signifie que l'étape d'initialisation est terminée. La tâche est maintenant prête à synchroniser toutes les nouvelles mises à jour depuis AWS IoT SiteWise.
- L'**ERROR** état de la tâche de synchronisation indique une erreur avec l'un des états précédents. Passez en revue le message d'erreur. Il y a peut-être un problème avec la configuration du rôle IAM. Si vous souhaitez utiliser un nouveau rôle IAM, supprimez la tâche de synchronisation à l'origine de l'erreur et créez-en une nouvelle avec le nouveau rôle.

Les erreurs de synchronisation apparaissent sur la page source du modèle, accessible depuis le tableau des sources du modèle d'entité de votre espace de travail. La page source du modèle affiche la liste des ressources dont la synchronisation a échoué. La plupart des erreurs sont automatiquement réessayées par la tâche de synchronisation, mais si la ressource nécessite une action, elle reste dans son **ERROR** état. Vous pouvez également obtenir une liste d'erreurs à l'aide de l'`ListSyncResourcesAPI`.

Pour voir toutes les erreurs répertoriées pour la source actuelle, procédez comme suit.

1. Accédez à votre espace de travail dans la [AWS IoT TwinMaker console](#).
2. Sélectionnez la AWS IoT SiteWise source répertoriée dans le modal Sources du modèle d'entité pour ouvrir la page de détails de la synchronisation des actifs.

The screenshot shows the AWS IoT SiteWise source configuration page. The breadcrumb navigation is: AWS IoT TwinMaker > Workspaces > SWSync > Source: AWS IoT SiteWise. The page title is 'AWS IoT SiteWise source' with a 'Disconnect' button in the top right.

**Overview**

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

**Errors (2)**

Find resources

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

3. Comme indiqué dans la capture d'écran précédente, toutes les ressources présentant des erreurs persistantes sont répertoriées dans le tableau des erreurs. Vous pouvez utiliser ce tableau pour rechercher et corriger les erreurs liées à des ressources spécifiques.

Les erreurs possibles sont les suivantes :

- Bien qu'il AWS IoT SiteWise prenne en charge les noms d'actifs dupliqués, il ne les prend en charge AWS IoT TwinMaker qu'au ROOT niveau, et non sous la même entité parent. Si vous avez deux actifs portant le même nom sous une entité parent dans AWS IoT SiteWise, l'un d'eux ne se synchronise pas. Pour corriger cette erreur, supprimez l'une des ressources ou déplacez-en une sous une autre ressource parent AWS IoT SiteWise avant de procéder à la synchronisation.
- Si vous avez déjà une entité avec le même identifiant que l'identifiant de AWS IoT SiteWise ressource, cette ressource ne se synchronise pas tant que vous ne supprimez pas l'entité existante.

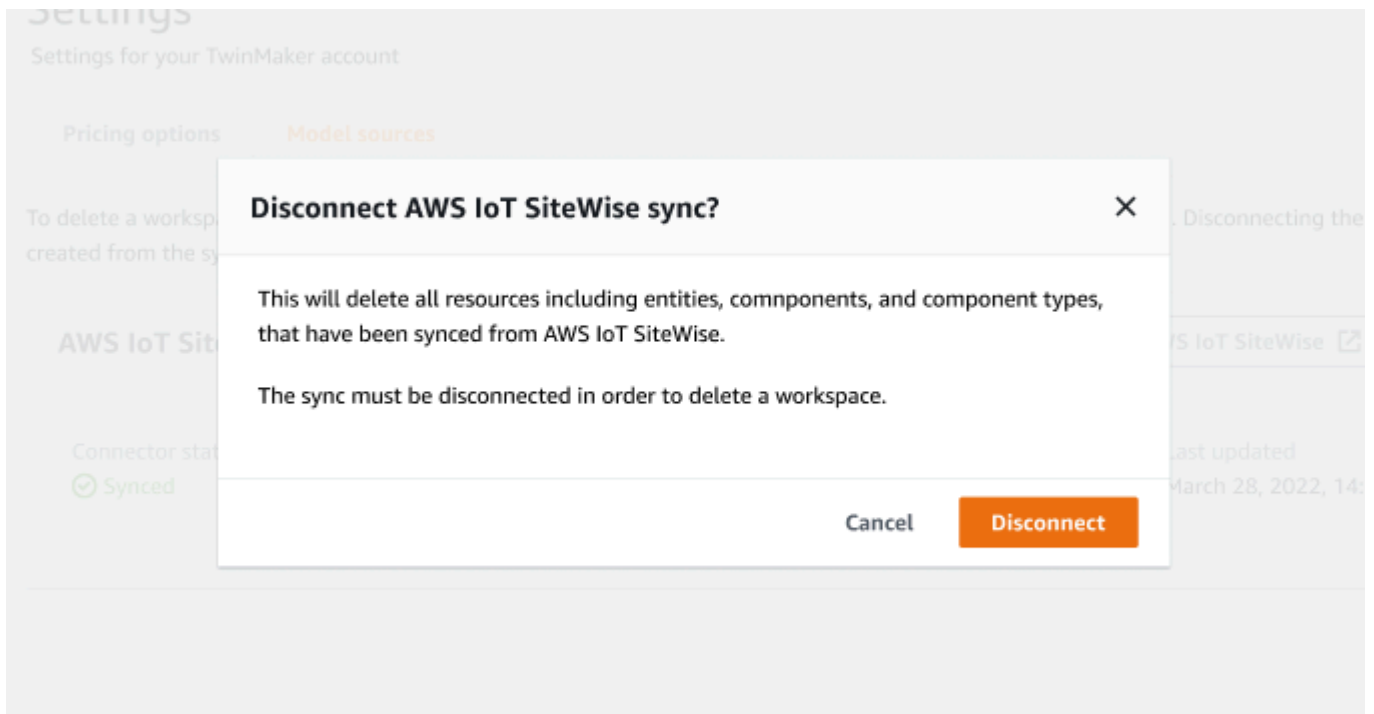
## Supprimer une tâche de synchronisation

Pour supprimer une tâche de synchronisation, procédez comme suit.

**⚠ Important**

Consultez [the section called “Différences entre les espaces de travail personnalisés et les espaces de travail par défaut”](#) pour plus d'informations sur les différences entre les espaces de travail personnalisés et par défaut.

1. Accédez à la [console AWS IoT TwinMaker](#).
2. Ouvrez l'espace de travail dans lequel vous souhaitez supprimer la tâche de synchronisation.
3. Sous Sources du modèle d'entité, sélectionnez la AWS IoT SiteWise source pour ouvrir la page de détails de la source.
4. Pour arrêter la tâche de synchronisation, choisissez Déconnecter. Confirmez votre choix de supprimer complètement la tâche de synchronisation.



Une fois qu'une tâche de synchronisation est supprimée, vous pouvez la créer à nouveau dans le même espace de travail ou dans un autre.

Vous ne pouvez pas supprimer un espace de travail s'il contient des tâches de synchronisation. Supprimez d'abord les tâches de synchronisation avant de supprimer un espace de travail.

En cas d'erreur lors de la suppression de la tâche de synchronisation, la tâche de synchronisation reste dans son DELETING état et est automatiquement réessayée. Vous pouvez désormais supprimer manuellement les entités ou les types de composants synchronisés en cas d'erreur liée à la suppression d'une ressource.

#### Note

Toutes les ressources synchronisées AWS IoT SiteWise sont d'abord supprimées, puis la tâche de synchronisation elle-même est supprimée.

## Limites de synchronisation des actifs

#### Important

Consultez [the section called “Différences entre les espaces de travail personnalisés et les espaces de travail par défaut”](#) pour plus d'informations sur les différences entre les espaces de travail personnalisés et par défaut.


Les [AWS IoT SiteWise quotas](#) étant supérieurs aux [AWS IoT TwinMaker quotas](#) par défaut, nous augmentons les limites suivantes pour les entités et les types de composants à partir AWS IoT SiteWise desquels la synchronisation est effectuée.

- 1 000 types de composants synchronisés dans un espace de travail, puisqu'il ne peut synchroniser que 1 000 modèles de ressources à partir de AWS IoT SiteWise celui-ci.
- 100 000 entités synchronisées dans un espace de travail, puisqu'il ne peut synchroniser que 100 000 actifs à partir de AWS IoT SiteWise celui-ci.
- 2 000 entités enfants maximum par entité parent. Il synchronise 2 000 actifs enfants par actif monoparental.

#### Note

L'[GetEntity](#) API renvoie uniquement les 50 premières entités enfants pour une propriété de hiérarchie, mais vous pouvez utiliser l'[GetPropertyValue](#) API pour paginer et récupérer la liste de toutes les entités enfants.

- 600 propriétés par composant synchronisé depuis AWS IoT SiteWise, ce qui permet de synchroniser des modèles d'actifs avec un total de 600 propriétés et hiérarchies.

 Note

Ces limites ne s'appliquent qu'aux entités synchronisées. Demandez une augmentation de quota si vous avez besoin d'augmenter ces limites pour les ressources non synchronisées.

# AWS IoT TwinMaker Intégration au tableau de bord Grafana

AWS IoT TwinMaker prend en charge l'intégration de Grafana via un plugin d'application. Utilisez la version 10.4.0 et les versions ultérieures de Grafana pour interagir avec votre application de jumeau numérique. Le AWS IoT TwinMaker plugin fournit des panneaux personnalisés, des modèles de tableau de bord et une source de données pour se connecter à vos données de jumeaux numériques.

Pour plus d'informations sur la façon d'intégrer Grafana et de configurer les autorisations pour votre tableau de bord, consultez les rubriques suivantes :

## Rubriques

- [Configuration CORS pour le visualiseur de scènes Grafana](#)
- [Configuration de votre environnement Grafana](#)
- [Création d'un rôle IAM dans le tableau de bord](#)
- [Création d'une politique en matière de lecteur AWS IoT TwinMaker vidéo](#)

### Note

Vous devez modifier la configuration CORS (partage de ressources entre origines) du compartiment Amazon S3 pour permettre à l'interface utilisateur de Grafana de charger les ressources depuis le compartiment. Pour obtenir des instructions, consultez [Configuration CORS pour le visualiseur de scènes Grafana](#).

Pour plus d'informations sur le plugin AWS IoT TwinMaker Grafana, consultez la documentation de [l'AWS IoT TwinMaker application](#).

Pour plus d'informations sur les composants clés du plugin Grafana, consultez ce qui suit :

- [AWS IoT TwinMaker datasource](#)
- [Modèles de tableau de bord](#)
- [Panneau Scene Viewer](#)
- [Panneau du lecteur vidéo](#)

# Configuration CORS pour le visualiseur de scènes Grafana

Le plugin AWS IoT TwinMaker Grafana nécessite une configuration CORS (partage de ressources entre origines), qui permet à l'interface utilisateur de Grafana de charger des ressources depuis le compartiment Amazon S3. Sans la configuration CORS, vous recevrez un message d'erreur tel que « Le chargement de la scène 3D a échoué en raison d'une défaillance du réseau » sur le Scene Viewer, car le domaine Grafana ne peut pas accéder aux ressources du compartiment Amazon S3.

Pour configurer votre compartiment Amazon S3 avec CORS, procédez comme suit :

1. Connectez-vous à la console IAM et ouvrez la [console Amazon S3](#).
2. Dans la liste des compartiments, choisissez le nom du compartiment que vous utilisez comme compartiment de ressources de votre AWS IoT TwinMaker espace de travail.
3. Choisissez Autorisations.
4. Dans la section Partage de ressources entre origines, sélectionnez Modifier pour ouvrir l'éditeur CORS.
5. Dans la zone de texte de l'éditeur de configuration CORS, tapez ou copiez-collez la configuration JSON CORS suivante en remplaçant le domaine de l'espace de travail Grafana par votre domaine *GRAFANA-WORKSPACE-DOMAIN*.

## Note

Vous devez conserver l'astérisque \* au début de l'élément "AllowedOrigins": JSON.

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

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

6. Sélectionnez Enregistrer les modifications pour terminer la configuration CORS.

Pour plus d'informations sur le CORS avec les compartiments Amazon S3, consultez [Utilisation du partage de ressources entre origines \(CORS\)](#).

## Configuration de votre environnement Grafana

Vous pouvez utiliser Amazon Managed Grafana pour un service entièrement géré ou configurer un environnement Grafana que vous gérez vous-même. Avec Amazon Managed Grafana, vous pouvez rapidement déployer, exploiter et adapter Grafana open source à vos besoins. Vous pouvez également configurer votre propre infrastructure pour gérer les serveurs Grafana.

Pour plus d'informations sur les deux options d'environnement Grafana, consultez les rubriques suivantes :

- [Amazon Managed Grafana](#)
- [Grafana autogéré](#)

## Amazon Managed Grafana

Amazon Managed Grafana fournit un AWS IoT TwinMaker plugin qui vous permet de vous intégrer rapidement à AWS IoT TwinMaker Grafana. Comme Amazon Managed Grafana gère les serveurs Grafana pour vous, vous pouvez visualiser vos données sans avoir à créer, empaqueter ou déployer du matériel ou toute autre infrastructure Grafana. Pour plus d'informations sur Amazon Managed Grafana, consultez [Qu'est-ce qu'Amazon Managed Grafana ?](#).

**Note**

Amazon Managed Grafana prend actuellement en charge la version 1.3.1 du plugin AWS IoT TwinMaker Grafana.

## Conditions préalables requises pour Amazon Managed Grafana

Pour l'utiliser AWS IoT TwinMaker dans un tableau de bord Amazon Managed Grafana, remplissez d'abord les conditions préalables suivantes :

- Créez un AWS IoT TwinMaker espace de travail. Pour plus d'informations sur la création d'espaces de travail, consultez [Getting started with AWS IoT TwinMaker](#).

**Note**

Lorsque vous créez un espace de travail Amazon Managed Grafana pour la première fois dans la console de AWS gestion, il AWS IoT TwinMaker n'est pas répertorié. Cependant, le plugin est déjà installé sur tous les espaces de travail. Vous pouvez trouver le AWS IoT TwinMaker plugin dans la liste des plugins open source Grafana. Vous pouvez trouver la AWS IoT TwinMaker source de données en choisissant Ajouter une source de données sur la page Sources de données.

Lorsque vous créez un espace de travail Amazon Managed Grafana, un rôle IAM est créé automatiquement pour gérer les autorisations pour l'instance Grafana. C'est ce que l'on appelle le rôle Workspace IAM. Il s'agit de l'option du fournisseur d'authentification que vous utiliserez pour configurer toutes les AWS IoT TwinMaker sources de données pour Grafana. Amazon Managed Grafana ne prend pas en charge l'ajout automatique d'autorisations pour AWS IoT TwinMaker. Vous devez donc configurer ces autorisations manuellement. Pour plus d'informations sur la configuration des autorisations manuelles, consultez [Création d'un rôle IAM dans le tableau de bord](#).

## Grafana autogéré

Vous pouvez choisir d'héberger votre propre infrastructure pour exécuter Grafana. Pour plus d'informations sur l'exécution locale de Grafana sur votre machine, consultez Installer [Grafana](#). Le AWS IoT TwinMaker plugin est disponible sur le catalogue public de Grafana. [Pour plus](#)

[d'informations sur l'installation de ce plugin dans votre environnement Grafana, consultez AWS IoT TwinMaker App.](#)

Lorsque vous exécutez Grafana localement, vous ne pouvez pas facilement partager des tableaux de bord ou donner accès à plusieurs utilisateurs. [Pour un guide de démarrage rapide écrit sur le partage de tableaux de bord à l'aide de Grafana local, consultez le référentiel d'exemples.AWS IoT TwinMaker](#) Cette ressource explique comment héberger un environnement Grafana sur Cloud9 et Amazon EC2 sur un point de terminaison public.

Vous devez déterminer le fournisseur d'authentification que vous utiliserez pour configurer les TwinMaker sources de données. Vous configurez les informations d'identification pour l'environnement en fonction de la chaîne d'informations d'identification par défaut (voir [Utilisation de la chaîne de fournisseurs d'informations d'identification par défaut](#)). Les informations d'identification par défaut peuvent être les informations d'identification permanentes de n'importe quel utilisateur ou rôle. Par exemple, si vous exécutez Grafana sur Amazon EC2, la chaîne d'informations d'identification par défaut a accès au rôle d'[exécution Amazon EC2](#), qui sera alors votre fournisseur d'authentification. Le nom de ressource IAM Amazon (ARN) du fournisseur d'authentification est requis dans les étapes de [Création d'un rôle IAM dans le tableau de bord](#).

## Création d'un rôle IAM dans le tableau de bord

Avec AWS IoT TwinMaker, vous pouvez contrôler l'accès aux données sur vos tableaux de bord Grafana. Les utilisateurs du tableau de bord Grafana doivent disposer de champs d'autorisation différents pour afficher les données et, dans certains cas, pour écrire des données. Par exemple, un opérateur d'alarme peut ne pas être autorisé à visionner des vidéos, tandis qu'un administrateur est autorisé à accéder à toutes les ressources. Grafana définit les autorisations via des sources de données, où des informations d'identification et un rôle IAM sont fournis. La AWS IoT TwinMaker source de données récupère les AWS informations d'identification avec les autorisations pour ce rôle. Si aucun rôle IAM n'est fourni, Grafana utilise l'étendue des informations d'identification, qui ne peut pas être réduite. AWS IoT TwinMaker

Pour utiliser vos AWS IoT TwinMaker tableaux de bord dans Grafana, vous devez créer un rôle IAM et y associer des politiques. Vous pouvez utiliser les modèles suivants pour vous aider à créer ces politiques.

### Créer une politique IAM

Créez une politique IAM appelée *YourWorkspaceId*DashboardPolicy dans la console IAM. Cette politique permet à vos espaces de travail d'accéder au compartiment et aux AWS IoT TwinMaker

ressources Amazon S3. Vous pouvez également décider d'utiliser [AWS IoT Greengrass Edge Connector pour Amazon Kinesis Video Streams](#), qui nécessite des autorisations pour les Kinesis Video Streams AWS IoT SiteWise et les ressources configurées pour le composant. En fonction de votre cas d'utilisation, choisissez l'un des modèles de politique suivants.

## 1. Aucune politique d'autorisation pour les vidéos

Si vous ne souhaitez pas utiliser le [panneau Grafana Video Player](#), créez la politique à l'aide du modèle suivant.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject"
      ],
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-bucket/*",
        "arn:aws:s3:::amzn-s3-demo-bucket"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "iottwinmaker:Get*",
        "iottwinmaker:List*"
      ],
      "Resource": [
        "arn:aws:iottwinmaker:us-east-1:111122223333:workspace/workspaceId",
        "arn:aws:iottwinmaker:us-east-1:111122223333:workspace/workspaceId/*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": "iottwinmaker:ListWorkspaces",
      "Resource": "*"
    }
  ]
}
```

```
]
}
```

Un compartiment Amazon S3 est créé pour chaque espace de travail. Il contient les modèles 3D et les scènes à visualiser sur un tableau de bord. Le [SceneViewer](#) panneau charge les éléments de ce compartiment.

## 2. Politique d'autorisation des vidéos limitée

Pour limiter l'accès au panneau du lecteur vidéo de Grafana, regroupez vos ressources AWS IoT Greengrass Edge Connector pour Amazon Kinesis Video Streams par balises. Pour plus d'informations sur la définition des autorisations associées à vos ressources vidéo, consultez [Création d'une politique en matière de lecteur AWS IoT TwinMaker vidéo](#).

## 3. Toutes les autorisations relatives aux vidéos

Si vous ne souhaitez pas regrouper vos vidéos, vous pouvez les rendre toutes accessibles depuis le lecteur vidéo Grafana. Toute personne ayant accès à un espace de travail Grafana peut lire des vidéos pour n'importe quel flux de votre compte et avoir accès en lecture seule à n'importe quelle AWS IoT SiteWise ressource. Cela inclut toutes les ressources créées dans le futur.

Créez la politique à l'aide du modèle suivant :

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject"
      ],
      "Resource": [
        "arn:aws:s3:::bucketName/*",
        "arn:aws:s3:::bucketName"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "iottwinmaker:Get*",
        "iottwinmaker:List*"
      ],
    }
  ]
}
```

```

    "Resource": [
      "arn:aws:iottwinmaker:us-
east-1:111122223333:workspace/workspaceId",
      "arn:aws:iottwinmaker:us-
east-1:111122223333: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*"
      }
    }
  }
]
}

```

Ce modèle de politique fournit les autorisations suivantes :

- Accès en lecture seule à un compartiment S3 pour charger une scène.
- Accès en lecture seule à toutes AWS IoT TwinMaker les entités et composants d'un espace de travail.
- Accès en lecture seule pour diffuser toutes les vidéos Kinesis Video Streams sur votre compte.
- Accès en lecture seule à l'historique des valeurs immobilières de tous les AWS IoT SiteWise actifs de votre compte.
- Ingestion de données dans n'importe quelle propriété d'un AWS IoT SiteWise actif étiqueté avec la clé `EdgeConnectorForKVS` et la valeur `workspaceId`.

## Marquer le contenu de votre caméra, AWS IoT SiteWise demander le téléchargement d'une vidéo depuis Edge

À l'aide du lecteur vidéo de Grafana, les utilisateurs peuvent demander manuellement que la vidéo soit téléchargée depuis le cache périphérique vers Kinesis Video Streams. Vous pouvez activer cette fonctionnalité pour toute AWS IoT SiteWise ressource associée à votre connecteur AWS IoT Greengrass Edge pour Amazon Kinesis Video Streams et associée à la `EdgeConnectorForKVS` clé.

La valeur de la balise peut être une liste d'identifiants de travail délimités par l'un des caractères suivants : `. . : + = @ _ / -` Par exemple, si vous souhaitez utiliser une AWS IoT SiteWise ressource associée à un connecteur AWS IoT Greengrass Edge pour Amazon Kinesis Video Streams AWS IoT TwinMaker dans les espaces de travail, vous pouvez utiliser une balise suivant ce modèle : `WorkspaceA/WorkspaceB/WorkspaceC` Le plugin Grafana impose que le `AWS IoT TwinMaker WorkspaceID` soit utilisé pour regrouper l'ingestion de données d'actifs. AWS IoT SiteWise

## Ajoutez des autorisations supplémentaires à la politique de votre tableau de bord

Le plugin AWS IoT TwinMaker Grafana utilise votre fournisseur d'authentification pour faire appel au rôle de tableau de `AssumeRole` bord que vous créez. En interne, le plugin restreint le plus grand nombre d'autorisations auxquelles vous avez accès en utilisant une politique de session lors de l' `AssumeRole` appel. Pour plus d'informations sur les politiques de session, consultez la section [Politiques de session](#).

Voici la politique permissive maximale que vous pouvez appliquer à votre rôle de tableau de bord pour un AWS IoT TwinMaker espace de travail :

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject"
      ],
      "Resource": [
        "arn:aws:s3:::bucketName/*",
        "arn:aws:s3:::bucketName"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "iottwinmaker:Get*",
        "iottwinmaker:List*"
      ],
      "Resource": [
        "arn:aws:iottwinmaker:us-east-1:111122223333:workspace/workspaceId",
        "arn:aws:iottwinmaker:us-east-1:111122223333: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"
        }
      }
    }
  ]
}

```

Si vous ajoutez des instructions qui Allow augmentent les autorisations, elles ne fonctionneront pas sur le AWS IoT TwinMaker plugin. Ceci est conçu pour garantir que les autorisations minimales nécessaires sont utilisées par le plugin.

Cependant, vous pouvez réduire davantage les autorisations. Pour plus d'informations, consultez [Création d'une politique en matière de lecteur AWS IoT TwinMaker vidéo](#).

## Création du rôle IAM du tableau de bord Grafana

Dans la console IAM, créez un rôle IAM appelé *YourWorkspaceId*DashboardRole. Attachez-le *YourWorkspaceId*DashboardPolicy au rôle.

Pour modifier la politique de confiance du rôle de tableau de bord, vous devez autoriser le fournisseur d'authentification Grafana à faire appel au rôle de tableau de bord. Mettez à jour la politique de confiance avec le modèle suivant :

## JSON

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

Pour plus d'informations sur la création d'un environnement Grafana et la recherche de votre fournisseur d'authentification, consultez. [Configuration de votre environnement Grafana](#)

## Création d'une politique en matière de lecteur AWS IoT TwinMaker vidéo

Voici un modèle de politique avec toutes les autorisations vidéo dont vous avez besoin pour le AWS IoT TwinMaker plugin de Grafana :

## JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject"
      ],
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-bucket/*",
        "arn:aws:s3:::amzn-s3-demo-bucket"
      ]
    }
  ],
}
```

```

    {
      "Effect": "Allow",
      "Action": [
        "iottwinmaker:Get*",
        "iottwinmaker:List*"
      ],
      "Resource": [
        "arn:aws:iottwinmaker:us-
east-1:111122223333:workspace/workspaceId",
        "arn:aws:iottwinmaker:us-
east-1:111122223333: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*"
        }
      }
    }
  ]
}

```

```
}
  }
]
}
```

Pour plus d'informations sur la politique complète, consultez le modèle de politique d'autorisation pour toutes les vidéos dans la [Créer une politique IAM](#) rubrique.

## Limitez l'accès à vos ressources

Le panneau Video Player de Grafana fait directement appel à Kinesis Video Streams et à l'IoT SiteWise pour offrir une expérience de lecture vidéo complète. Pour éviter tout accès non autorisé à des ressources qui ne sont pas associées à votre AWS IoT TwinMaker espace de travail, ajoutez des conditions à la politique IAM pour votre rôle de tableau de bord d'espace de travail.

## Limitez la portée des autorisations GET

Vous pouvez limiter l'accès à vos Amazon Kinesis Video Streams AWS IoT SiteWise et à vos ressources en balisant les ressources. Vous avez peut-être déjà tagué votre AWS IoT SiteWise appareil photo en fonction du AWS IoT TwinMaker WorkspaceID pour activer la fonctionnalité de demande de téléchargement de vidéos. Consultez la rubrique [Importer une vidéo depuis le périphérique](#). Vous pouvez utiliser la même paire clé-valeur pour limiter l'accès GET aux AWS IoT SiteWise ressources et pour baliser vos Kinesis Video Streams de la même manière.

Vous pouvez ensuite ajouter cette condition aux instructions kinesisvideo et iotsitewise dans :  
*YourWorkspaceIdDashboardPolicy*

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

## Cas d'utilisation réel : regroupement de caméras

Dans ce scénario, vous disposez d'un large éventail de caméras qui surveillent le processus de cuisson des biscuits dans une usine. Des lots de pâte à biscuits sont préparés dans la salle de préparation, la pâte est congelée dans le congélateur et les biscuits sont cuits dans la salle de cuisson. Il y a des caméras dans chacune de ces salles et différentes équipes d'opérateurs

surveillent séparément chaque processus. Vous souhaitez que chaque groupe d'opérateurs soit autorisé pour sa chambre respective. Lors de la création d'un jumeau numérique pour la fabrique de cookies, un seul espace de travail est utilisé, mais les autorisations de l'appareil photo doivent être définies par pièce.

Vous pouvez obtenir cette séparation des autorisations en étiquetant les groupes de caméras en fonction de leur GroupingID. Dans ce scénario, les identifiants de groupe sont BatterRoom, et FreezerRoom. BakingRoom La caméra de chaque pièce est connectée à Kinesis Video Streams et doit comporter une étiquette portant la mention : Key EdgeConnectorForKVS =, Value BatterRoom =. La valeur peut être une liste de groupes délimités par l'un des caractères suivants : . : + = @ \_ / -

Pour modifier le *YourWorkspaceId*DashboardPolicy, utilisez les déclarations de politique suivantes :

```

...
{
  "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*"
    }
  }
},

```

```
...
```

Ces instructions limitent la lecture de vidéos en streaming et l'accès à l'historique des AWS IoT SiteWise propriétés à des ressources spécifiques d'un groupe. Le *groupingId* est défini par votre cas d'utilisation. Dans notre scénario, il s'agirait du RoomID.

## Réduisez la portée de AWS IoT SiteWise BatchPutAssetPropertyValue l'autorisation

L'octroi de cette autorisation active la [fonctionnalité de demande de téléchargement de vidéos dans le lecteur vidéo](#). Lorsque vous téléchargez une vidéo, vous pouvez spécifier un intervalle de temps et soumettre la demande en choisissant Soumettre dans le panneau du tableau de bord de Grafana.

Pour accorder des BatchPutAssetPropertyValue autorisations à iotsitewise :, utilisez la politique par défaut :

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

En utilisant cette politique, les utilisateurs peuvent appeler BatchPutAssetPropertyValue pour n'importe quelle propriété de l' AWS IoT SiteWise appareil photo. Vous pouvez restreindre l'autorisation pour un PropertyID spécifique en le spécifiant dans la condition de l'instruction.

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

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

Le panneau Video Player de Grafana ingère des données dans la propriété de mesure, nommée VideoUploadRequest, pour lancer le téléchargement de vidéos depuis le cache périphérique vers Kinesis Video Streams. Trouvez le PropertyID de cette propriété dans la AWS IoT SiteWise console. Pour modifier le *YourWorkspaceId*DashboardPolicy, utilisez la déclaration de politique suivante :

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

Cette déclaration limite l'ingestion de données à une propriété spécifique de votre AWS IoT SiteWise appareil photo étiqueté. Pour plus d'informations, consultez la section [AWS IoT SiteWise Fonctionnement avec IAM](#).

# Connect les AWS IoT SiteWise alarmes aux tableaux de AWS IoT TwinMaker bord Grafana

## Note

Cette fonctionnalité est en version préliminaire publique et est sujette à modification.

AWS IoT TwinMaker est capable d'importer des alarmes AWS IoT SiteWise et des événements dans AWS IoT TwinMaker les composants. Cela vous permet de demander l'état des alarmes et de configurer les seuils d'alarme sans implémenter de connecteur de données personnalisé pour la migration AWS IoT SiteWise des données. Vous pouvez utiliser le plugin AWS IoT TwinMaker Grafana pour visualiser l'état des alarmes et configurer le seuil d'alarme dans Grafana, sans effectuer d'appels d'API AWS IoT TwinMaker ni interagir directement avec les alarmes. AWS IoT SiteWise

## Note

Avis de fin de support : le 20 mai 2026, AWS le support de AWS IoT Events. Après le 20 mai 2026, vous ne pourrez plus accéder à la AWS IoT Events console ni aux AWS IoT Events ressources. Pour plus d'informations, voir [AWS IoT Events fin du support](#).

## AWS IoT SiteWise prérequis de configuration des alarmes

Avant de créer des alarmes et de les intégrer dans votre tableau de bord Grafana, assurez-vous d'avoir passé en revue les prérequis suivants :

- Familiarisez-vous avec AWS IoT SiteWise le modèle et le système d'actifs. Pour plus d'informations, consultez les sections [Création de modèles d'actifs](#) et [Création d'actifs](#) dans le Guide de AWS IoT SiteWise l'utilisateur.
- Familiarisez-vous avec les modèles d'alarme IoT Events et apprenez à les associer à un AWS IoT SiteWise modèle. Pour plus d'informations, consultez la section [Définition des alarmes liées aux AWS IoT événements](#) dans le guide de AWS IoT SiteWise l'utilisateur.
- AWS IoT TwinMaker Intégrez Grafana afin de pouvoir accéder à vos AWS IoT TwinMaker ressources dans Grafana. Pour plus d'informations, voir, [AWS IoT TwinMaker Intégration au tableau de bord Grafana](#).

## Définir le rôle IAM du composant d' AWS IoT SiteWise alarme

AWS IoT TwinMaker utilise le rôle IAM de l'espace de travail pour interroger et configurer le seuil d'alarme dans Grafana. Les autorisations suivantes sont requises dans le rôle d' AWS IoT TwinMaker espace de travail afin d'interagir avec les AWS IoT SiteWise alarmes dans Grafana :

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

Dans la [AWS IoT TwinMaker console](#), créez une entité qui représente votre AWS IoT SiteWise actif. Assurez-vous d'ajouter un composant pour cette entité en l'utilisant `com.amazon.iotsitewise.alarm` comme type de composant et de sélectionner les modèles d'actifs et d'alarmes correspondants.

## 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.

La capture d'écran ci-dessus est un exemple de création de cette entité avec le type `com.amazon.iotsitewise.alarm`.

Lorsque vous créez ce composant, il importe AWS IoT TwinMaker automatiquement les propriétés d'alarme associées depuis AWS IoT SiteWise et AWS IoT Events. Vous pouvez ensuite répéter ce modèle de type de composant d'alarme pour créer des composants d'alarme pour tous les actifs nécessaires dans votre espace de travail.

## Interrogation et mise à jour via l' AWS IoT TwinMaker API

Après avoir créé les composants d'alarme, vous pouvez demander l'état de l'alarme, le seuil et mettre à jour les seuils d'alarme via l' AWS IoT TwinMaker API.

Vous trouverez ci-dessous un exemple de demande pour demander l'état d'une alarme :

```
aws iottwinmaker get-property-value-history --cli-input-json \  
{  
  "workspaceId": "{workspaceId}",  
  "entityId": "{entityId}",  
  "componentName": "{componentName}",
```

```

    "selectedProperties": ["alarm_status"],
    "startTime": "{startTimeIsoString}",
    "endTime": "{endTimeIsoString}"
  }'

```

Vous trouverez ci-dessous un exemple de demande pour demander le seuil d'alarme.

```

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

```

Vous trouverez ci-dessous un exemple de demande de mise à jour du seuil d'alarme :

```

aws iottwinmaker batch-put-property-values --cli-input-json \
'{
  "workspaceId": "{workspaceId}",
  "entries": [
    {
      "entityPropertyReference": {
        "entityId": "{entityId}",
        "componentName": "{componentName}",
        "propertyName": "alarm_threshold"
      },
      "propertyValues": [
        {
          "value": {
            "doubleValue": "{newThreshold}"
          },
          "time": "{effectiveTimeIsoString}"
        }
      ]
    }
  ]
}'

```

## Configurez votre tableau de bord Grafana pour les alarmes

Un deuxième rôle IAM de tableau de bord activé pour l'écriture doit être créé. Il s'agit d'un rôle normal, mais avec l'autorisation d'ajouter l'action `iottwinmaker:BatchPutPropertyValues` à l'ARN de l' TwinMaker espace de travail, comme dans l'exemple ci-dessous.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iottwinmaker:Get*",
        "iottwinmaker:List*",
        "iottwinmaker:BatchPutPropertyValues"
      ],
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-bucket",
        "arn:aws:s3:::amzn-s3-demo-bucket/*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": "iottwinmaker:ListWorkspaces",
      "Resource": "*"
    }
  ]
}
```

Vous pouvez également ajouter cette déclaration à la fin de votre rôle IAM à la place :

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

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

L'ARN d'écriture de la source de données doit être défini avec le rôle d'écriture du tableau de bord que vous avez créé.

Après avoir modifié votre rôle IAM, connectez-vous à votre tableau de bord Grafana pour prendre le rôle arn mis à jour. Cochez la case Définir les autorisations d'écriture pour le panneau de configuration d'alarme et copiez l'ARN du rôle d'écriture.

The screenshot displays the AWS IoT TwinMaker Settings interface. At the top, there are tabs for 'Settings' and 'Dashboards'. Below the tabs, the 'Name' field is set to 'AWS IoT TwinMaker-4' and the 'Default' toggle is turned off. 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 message is present under 'Assume Role ARN', stating: '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.' 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 (arn:aws:iam:\*) which is circled in red. At the bottom, there are buttons for 'Back', 'Explore', 'Delete', and 'Save & test'.

Utilisez le tableau de bord Grafana pour la visualisation des alarmes

Utilisez la procédure suivante pour ajouter un panneau de configuration d'alarme à votre tableau de bord et le configurer :

1. Sélectionnez l'espace de travail dans les options du panneau.
2. Définissez votre source de données dans la configuration de la requête.
3. Utilisez le type de requête suivant : `Get Property Value History by Entity`.
4. Sélectionnez une entité ou une variable d'entité à laquelle vous souhaitez ajouter une alarme.
5. Une fois que vous avez sélectionné l'entité, sélectionnez un composant ou une variable de composant auquel appliquer une propriété.
6. Pour la propriété, choisissez : `alarm_status` et `alarm_threshold`.

Lorsqu'il est connecté, vous devriez voir l'identifiant de l'alarme et son seuil actuel.

### Note

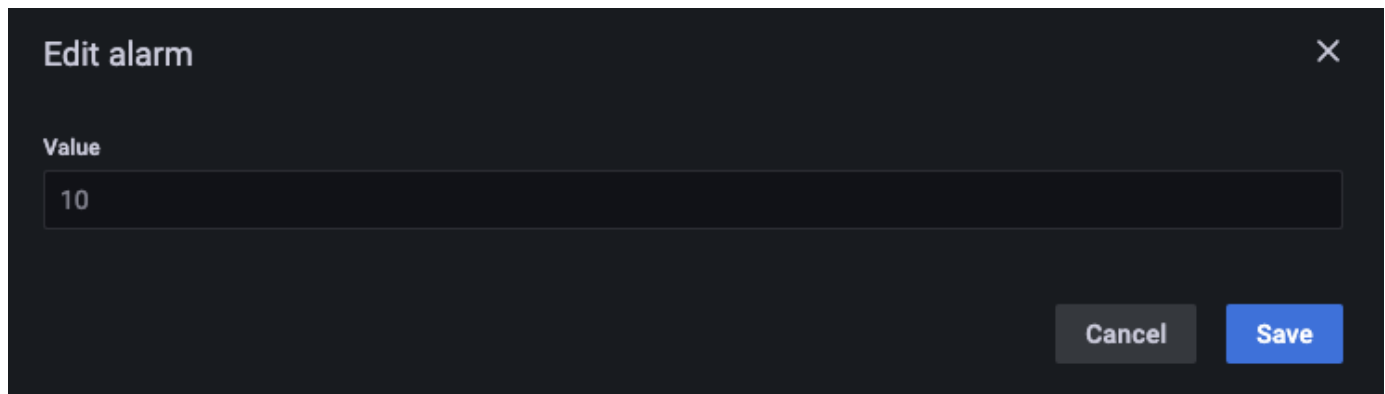
Pour l'aperçu public, aucune notification n'est affichée. Vous devez vérifier l'état et le seuil de votre alarme pour vous assurer que les propriétés ont été correctement appliquées.

7. L'ordre croissant des requêtes par défaut doit être utilisé pour que la dernière valeur s'affiche.
8. La section de filtre de la requête peut être laissée vide. Une configuration complète est illustrée ci-dessous :

The screenshot displays the Grafana Labs Alarm Config Test / Edit Panel. The main area shows the alarm configuration for 'OverheatAlarm' with a threshold of 10. The query configuration is set to 'Get Property Value History by Entity' for the entity 'ReactorCore' and component 'OverheatAlarm'. The selected properties are 'alarm\_status (STRING)' and 'alarm\_threshold (DOUBLE)'. The filter is set to 'name = value'. The right sidebar shows panel options and the workspace configuration.

9. En utilisant le bouton Modifier l'alarme, vous pouvez ouvrir une boîte de dialogue pour modifier le seuil d'alarme actuel.

## 10. Sélectionnez Enregistrer pour définir la nouvelle valeur de seuil.



The screenshot shows a dark-themed dialog box titled "Edit alarm" with a close button (X) in the top right corner. Below the title, there is a label "Value" and a text input field containing the number "10". At the bottom right of the dialog, there are two buttons: "Cancel" and "Save".

### Note

Ce panneau ne doit être utilisé qu'avec une plage de temps en temps réel incluant le présent. Son utilisation avec des plages de temps qui se terminent et commencent dans le passé peut afficher des valeurs inattendues lorsque vous modifiez toujours les seuils d'alarme en tant que seuil actuel.

# AWS IoT TwinMaker Intégration à Matterport

Matterport propose diverses options de capture pour scanner des environnements réels et créer des modèles 3D immersifs, également connus sous le nom de jumeaux numériques Matterport. Ces modèles sont appelés espaces Matterport. AWS IoT TwinMaker prend en charge l'intégration de Matterport, vous permettant d'importer vos jumeaux numériques Matterport dans vos scènes. AWS IoT TwinMaker En associant les jumeaux numériques Matterport à AWS IoT TwinMaker, vous pouvez visualiser et surveiller votre système de jumeaux numériques dans un environnement virtuel.



[Pour plus d'informations sur l'utilisation de Matterport, consultez la documentation de Matterport sur AWS IoT TwinMaker la page Matterport.](#)

## Sujets d'intégration

- [Présentation de l'intégration](#)
- [Conditions préalables à l'intégration de Matterport](#)
- [Générez et enregistrez vos informations d'identification Matterport](#)
- [Stockez vos informations d'identification Matterport dans AWS Secrets Manager](#)
- [Importer des espaces Matterport dans des scènes AWS IoT TwinMaker](#)
- [Utilisez les espaces Matterport dans votre tableau de bord Grafana AWS IoT TwinMaker](#)

- [Utilisez les espaces Matterport dans votre application Web AWS IoT TwinMaker](#)

## Présentation de l'intégration

Cette intégration vous permet d'effectuer les opérations suivantes :

- Utilisez vos tags et espaces Matterport dans le kit d' AWS IoT TwinMaker applications.
- Consultez vos données Matterport importées dans votre tableau de bord Grafana AWS IoT TwinMaker . Pour plus d'informations sur l'utilisation AWS IoT TwinMaker de Grafana, consultez la documentation d'intégration du tableau de bord [Grafana](#).
- Importez vos espaces Matterport dans vos AWS IoT TwinMaker scènes.
- Sélectionnez et importez les balises Matterport que vous souhaitez lier aux données de votre AWS IoT TwinMaker scène.
- Affichez automatiquement votre espace Matterport et balisez les modifications apportées à votre AWS IoT TwinMaker scène et approuvez les modifications à synchroniser.

Le processus d'intégration comprend 3 étapes critiques.

1. [Générez et enregistrez vos informations d'identification Matterport](#)
2. [Stockez vos informations d'identification Matterport dans AWS Secrets Manager](#)
3. [Importer des espaces Matterport dans des scènes AWS IoT TwinMaker](#)

Vous commencez votre intégration dans la [AWS IoT TwinMaker console](#). Sur la page Paramètres de la console, sous Ressources tierces, ouvrez l'intégration Matterport pour naviguer entre les différentes ressources requises pour l'intégration.

The screenshot shows the AWS IoT TwinMaker console interface. The top navigation bar includes the AWS logo, 'Services', 'Resource Groups', and user account information. The left sidebar contains navigation options like 'How it works', 'Workspaces', and 'Settings'. The main content area is titled 'Settings' and '3rd party resources'. It features 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. Below this is a 'Connected accounts' table which is currently empty, showing 'No connections' and a button to go to AWS Secret Manager.

## Conditions préalables à l'intégration de Matterport

Avant d'intégrer Matterport à, AWS IoT TwinMaker assurez-vous de remplir les conditions préalables suivantes :

- Vous avez acheté un compte Matterport d'entreprise et les produits [Matterport](#) nécessaires à l'intégration. AWS IoT TwinMaker
- Vous disposez d'un AWS IoT TwinMaker espace de travail. Pour plus d'informations, consultez la section [Mise en route avec AWS IoT TwinMaker](#).

- Vous avez mis à jour votre rôle dans AWS IoT TwinMaker l'espace de travail. Pour plus d'informations sur la création d'un rôle d'espace de travail, voir [Créer et gérer un rôle de service pour AWS IoT TwinMaker](#).

Ajoutez les éléments suivants à votre rôle dans l'espace de travail :

```
{
  "Effect": "Allow",
  "Action": "secretsmanager:GetSecretValue",
  "Resource": [
    "AWS Secrets Manager secret ARN"
  ]
}
```

- Vous devez contacter Matterport pour configurer les licences nécessaires à l'activation de l'intégration. Matterport activera également un Private Model Embed (PME) pour l'intégration.

Si vous avez déjà un gestionnaire de compte Matterport, contactez-le directement.

Utilisez la procédure suivante pour contacter Matterport et demander une intégration si vous n'avez pas de point de contact Matterport :

1. Ouvrez le [Matterport et AWS IoT TwinMaker](#) la page.
2. Cliquez sur le bouton Contactez-nous pour ouvrir le formulaire de contact.
3. Renseignez les informations requises sur le formulaire.
4. Lorsque vous êtes prêt, choisissez SAY HELLO pour envoyer votre demande à Matterport.

Une fois que vous avez demandé l'intégration, vous pouvez générer les informations d'identification nécessaires au SDK Matterport et au Private Model Embed (PME) nécessaires pour poursuivre le processus d'intégration.

#### Note

Cela peut impliquer que vous deviez payer des frais pour l'achat de nouveaux produits ou services.

# Générez et enregistrez vos informations d'identification Matterport

Pour intégrer Matterport à AWS IoT TwinMaker, vous devez fournir AWS Secrets Manager des informations d'identification Matterport. Utilisez la procédure suivante pour générer les informations d'identification du SDK Matterport.

1. Connectez-vous à votre compte [Matterport](#).
2. Accédez à la page des paramètres de votre compte.
3. Une fois sur la page des paramètres, sélectionnez l'option Outils de développement.
4. Sur la page des outils de développement, accédez à la section Gestion des clés du SDK.
5. Une fois dans la section Gestion des clés du SDK, sélectionnez l'option permettant d'ajouter une nouvelle clé du SDK.
6. Une fois que vous avez la clé du SDK Matterport, ajoutez des domaines à la clé pour et à AWS IoT TwinMaker votre serveur Grafana. Si vous utilisez le kit d' AWS IoT TwinMaker applications, assurez-vous d'ajouter également votre domaine personnalisé.
7. Ensuite, recherchez la section Gestion de l'intégration des applications, vous devriez voir votre application PME répertoriée. Enregistrez les informations suivantes :
  - L'identifiant du client
  - Le secret du client

## Note

Comme le secret client ne vous est présenté qu'une seule fois, nous vous recommandons vivement de l'enregistrer. Vous devez présenter votre secret client dans la AWS Secrets Manager console pour poursuivre l'intégration de Matterport.

Ces informations d'identification sont automatiquement créées lorsque vous avez acheté les composants nécessaires et que le PME de votre compte a été activé par Matterport. Si ces informations d'identification n'apparaissent pas, contactez Matterport. Pour demander un contact, consultez le [Matterport et](#) le formulaire de AWS IoT TwinMaker contact.

[Pour plus d'informations sur les informations d'identification du SDK Matterport, consultez la documentation officielle du SDK de Matterport. Vue d'ensemble des documents SDK Docs.](#)

# Stockez vos informations d'identification Matterport dans AWS Secrets Manager

Utilisez la procédure suivante pour enregistrer vos informations d'identification Matterport dans AWS Secrets Manager

## Note

Vous avez besoin du secret client créé à partir de la procédure décrite dans la [Générez et enregistrez vos informations d'identification Matterport](#) rubrique pour poursuivre l'intégration de Matterport.

1. Connectez-vous à la AWS Secrets Manager console.
2. Accédez à la page Secrets et sélectionnez Enregistrer un nouveau secret.
3. Pour le type de secret, sélectionnez Autre type de secret.
4. Dans la section Paires clé/valeur, ajoutez les paires clé-valeur suivantes, avec vos informations d'identification Matterport comme valeurs :
  - Créez une paire clé-valeur, avec Key : `application_key` et Value : `<your Matterport credentials>`
  - Créez une paire clé-valeur, avec Key : `client_id` et Value : `<your Matterport credentials>`
  - Créez une paire clé-valeur, avec Key : `client_secret` et Value : `<your Matterport credentials>`

Une fois terminé, vous devriez avoir une configuration similaire à l'exemple suivant :

### Key/value pairs [Info](#)

| Key/value                                    | Plaintext   |                                       |
|--|---|---------------------------------------|
| <input type="text" value="application_key"/> | <input type="text" value="matterport_application_key"/>         | <input type="button" value="Remove"/> |
| <input type="text" value="client_id"/>       | <input type="text" value="matterport_oauth_app_client_id"/>     | <input type="button" value="Remove"/> |
| <input type="text" value="client_secret"/>   | <input type="text" value="matterport_oauth_app_client_secret"/> | <input type="button" value="Remove"/> |
| <input type="button" value="+ Add row"/>     |   |                                       |

5. Pour la clé de chiffrement, vous pouvez laisser la clé de chiffrement par défaut `aws/secretsmanager` sélectionnée.
6. Choisissez `Next` pour accéder à la page de configuration secrète.
7. Remplissez le champ `Nom du secret` et `Description`.
8. Ajoutez un tag à ce secret dans la section `Tags`.

Lors de la création de la balise, attribuez la clé `AWSIoTtwinMaker_Matterport` comme indiqué dans la capture d'écran suivante :

AWS Secrets Manager > Secrets > Store a new secret

Step 1  
Choose secret type

Step 2  
**Configure secret**

Step 3  
Configure rotation - optional

Step 4  
Review

## Configure secret

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

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

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

**Description - optional**

Maximum 250 characters.

### Tags - optional

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

### Note

Vous devez ajouter un tag. Les balises sont obligatoires lors de l'ajout de secrets tiers AWS Secrets Manager, bien que les balises soient répertoriées comme facultatives.

Le champ Value est facultatif. Une fois que vous avez fourni une clé, vous pouvez sélectionner Ajouter pour passer à l'étape suivante.

9. Choisissez Next pour passer à la page Configurer la rotation. La configuration d'une rotation secrète est facultative. Si vous souhaitez terminer d'ajouter votre secret et que vous n'avez pas besoin d'une rotation, choisissez à nouveau Next. Pour plus d'informations sur la rotation secrète, voir [Rotation AWS Secrets Manager des secrets](#).
10. Confirmez votre configuration secrète sur la page de révision. Une fois que vous êtes prêt à ajouter votre secret, choisissez Store.

Pour plus d'informations sur l'utilisation AWS Secrets Manager, consultez la [AWS Secrets Manager documentation](#) suivante :

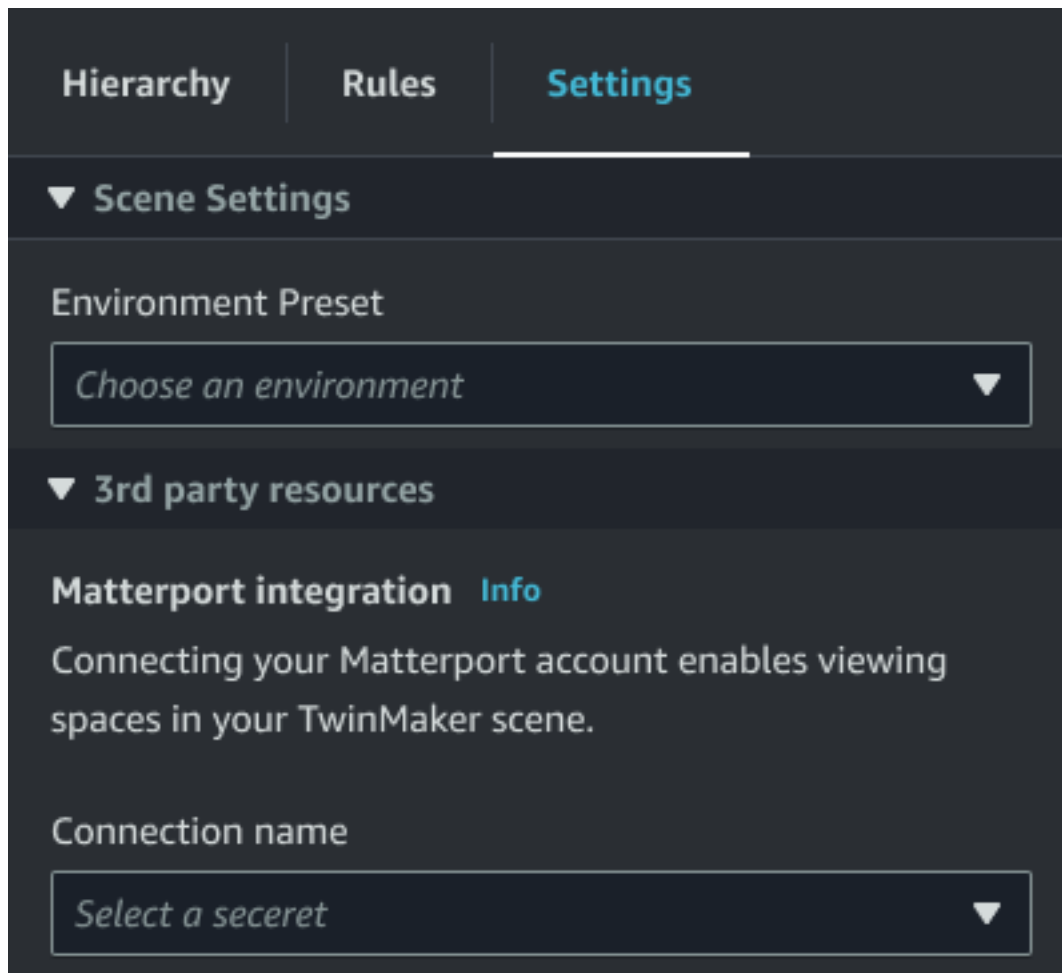
- [Créez et gérez des secrets avec AWS Secrets Manager](#)
- [Qu'est-ce que c'est AWS Secrets Manager ?](#)
- [Faire pivoter AWS Secrets Manager les secrets](#)

Vous êtes maintenant prêt à importer vos ressources Matterport dans AWS IoT TwinMaker des scènes. Reportez-vous à la procédure décrite dans la section suivante, [Importer des espaces Matterport dans des scènes AWS IoT TwinMaker](#)

## Importer des espaces Matterport dans des scènes AWS IoT TwinMaker

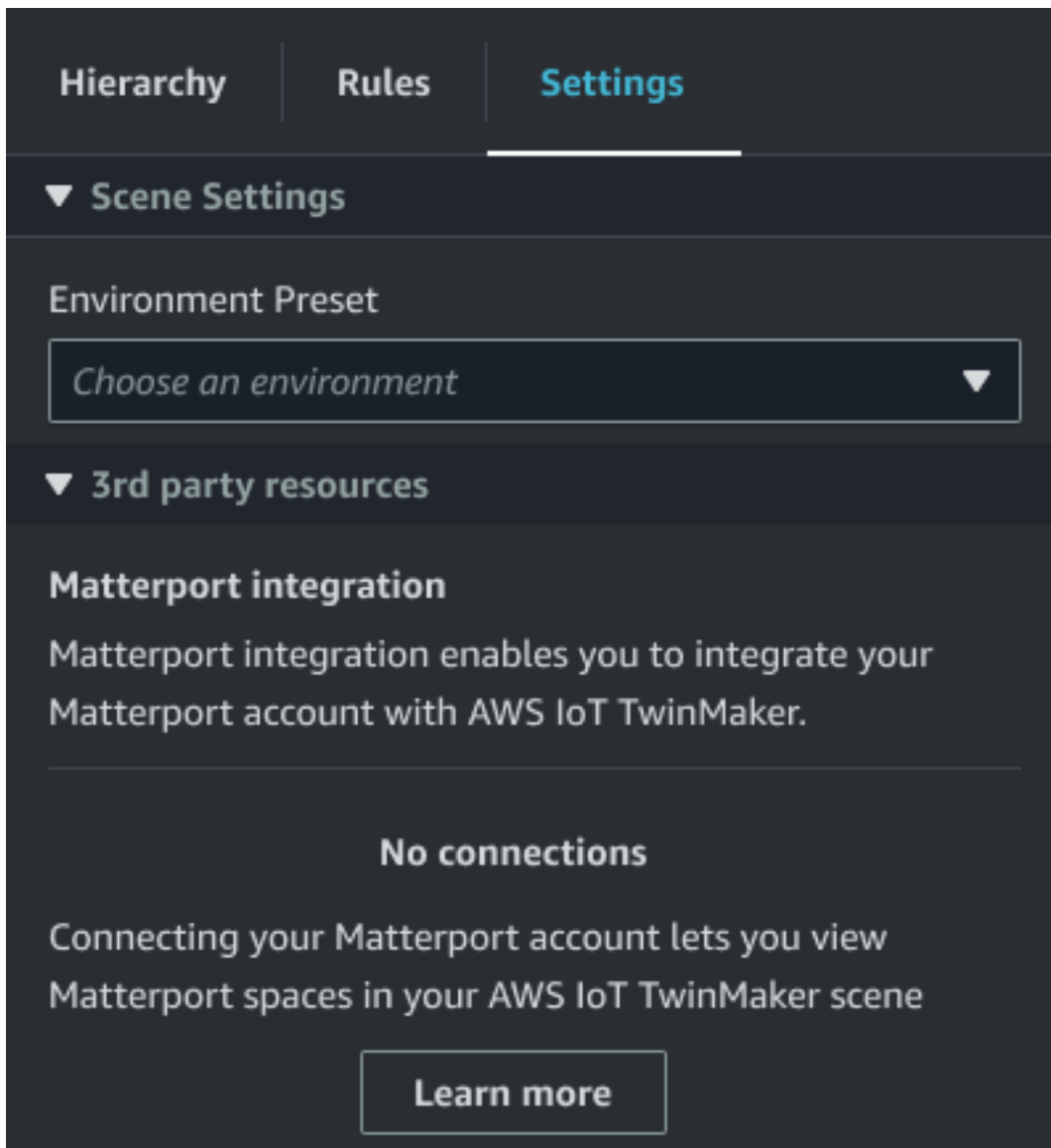
Ajoutez des scans Matterport à votre scène en sélectionnant le compte Matterport connecté sur la page des paramètres de scène. Suivez la procédure suivante pour importer vos scans et tags Matterport :

1. Connectez-vous à la [console AWS IoT TwinMaker](#).
2. Créez ou ouvrez une AWS IoT TwinMaker scène existante dans laquelle vous souhaitez utiliser un espace Matterport.
3. Une fois la scène ouverte, accédez à l'onglet Paramètres.
4. Dans Paramètres, sous Ressources tierces, recherchez le nom de la connexion et entrez le secret que vous avez créé dans la procédure à partir de [Stockez vos informations d'identification Matterport dans AWS Secrets Manager](#).

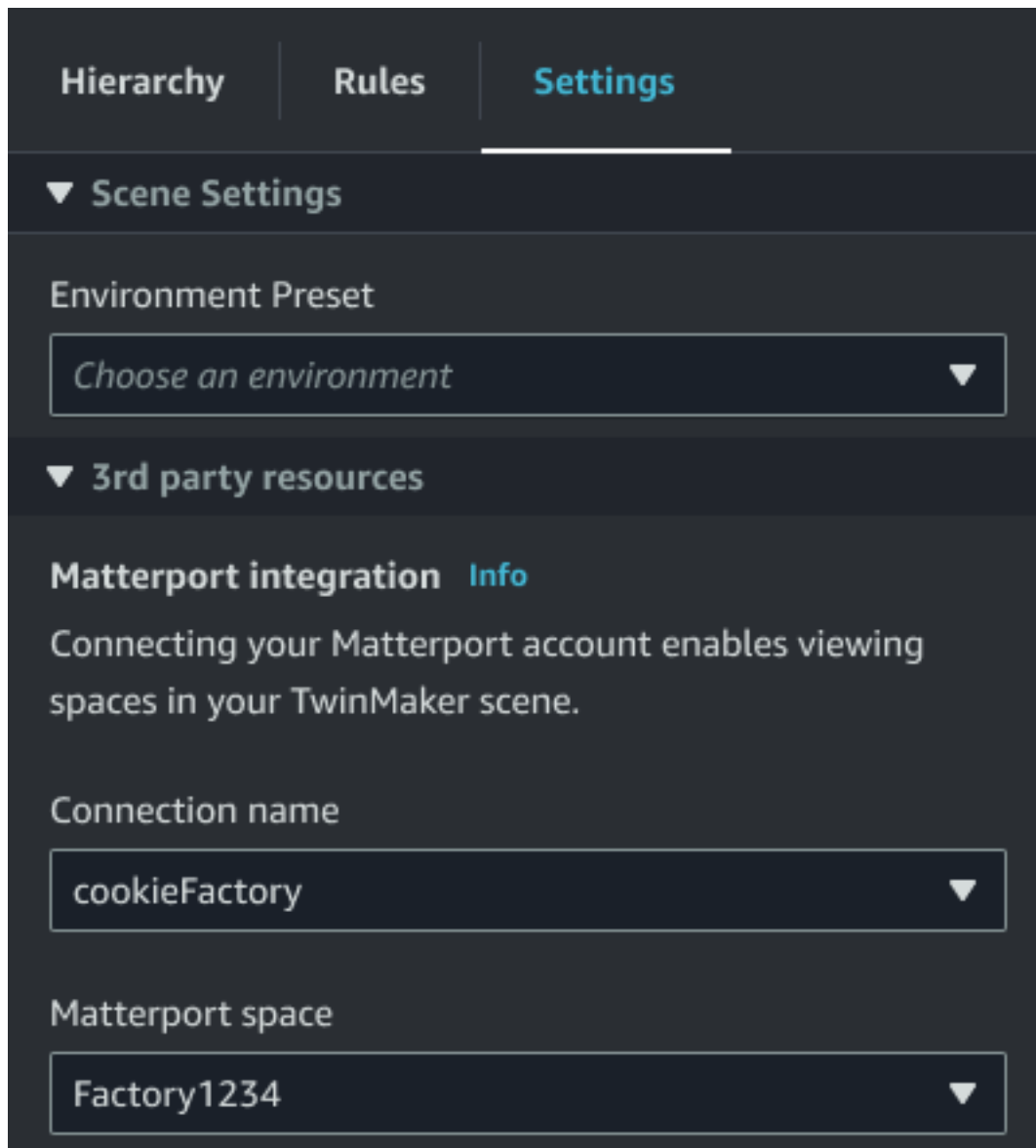


**Note**

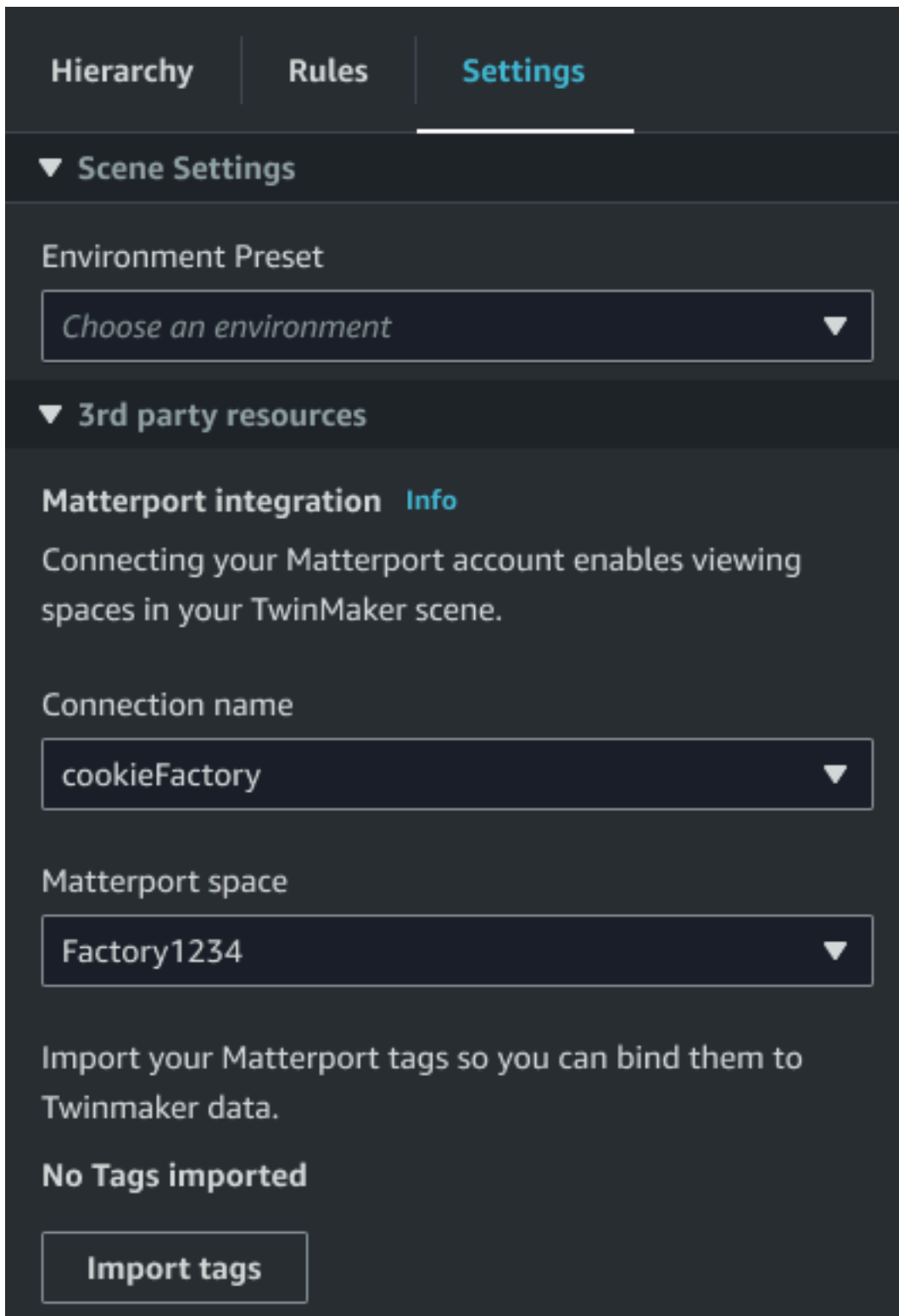
Si le message « Aucune connexion » s'affiche, accédez à la page des paramètres de la [AWS IoT TwinMaker console](#) pour commencer le processus d'intégration de Matterport.



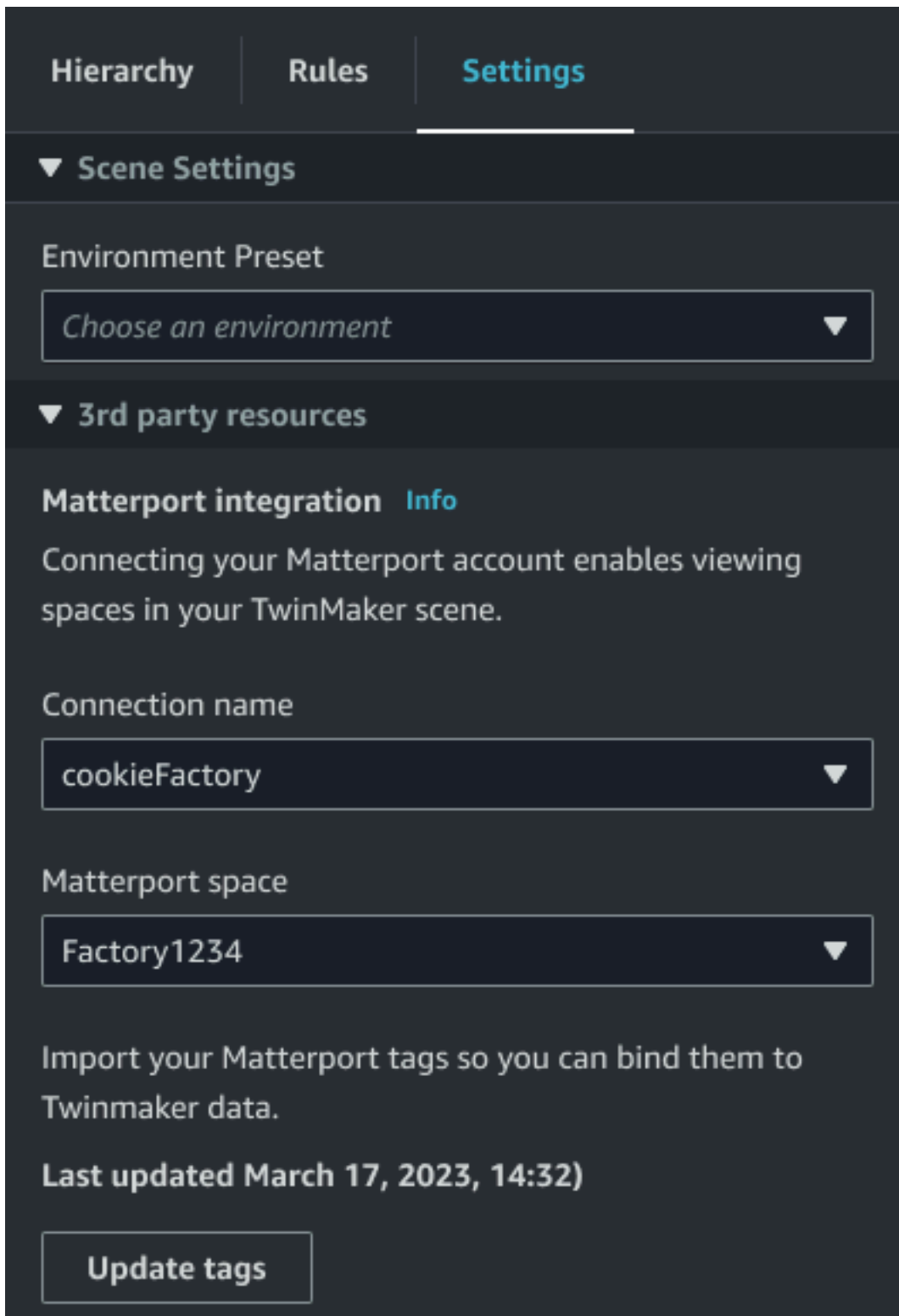
5. Ensuite, choisissez l'espace Matterport que vous souhaitez utiliser dans votre scène en le sélectionnant dans le menu déroulant de l'espace Matterport.



- Après avoir sélectionné un espace, vous pouvez importer vos tags Matterport et les convertir en tags de AWS IoT TwinMaker scène en appuyant sur le bouton Importer des tags.



Après avoir importé les tags Matterport, le bouton est remplacé par un bouton Mettre à jour les tags. Vous pouvez continuellement mettre à jour vos tags Matterport AWS IoT TwinMaker afin qu'ils reflètent toujours les modifications les plus récentes apportées à votre compte Matterport.



7. Vous avez réussi l'intégration AWS IoT TwinMaker à Matterport, et votre AWS IoT TwinMaker scène contient à la fois votre espace Matterport et vos tags importés. Vous pouvez travailler dans cette scène comme vous le feriez avec n'importe quelle autre AWS IoT TwinMaker scène.

Pour plus d'informations sur l'utilisation des AWS IoT TwinMaker scènes, voir [Création et modification de AWS IoT TwinMaker scènes](#).

## Utilisez les espaces Matterport dans votre tableau de bord Grafana AWS IoT TwinMaker

Une fois que vous avez importé votre espace Matterport dans une AWS IoT TwinMaker scène, vous pouvez visualiser cette scène avec l'espace Matterport dans votre tableau de bord Grafana. Si vous avez déjà configuré Grafana avec AWS IoT TwinMaker, vous pouvez simplement ouvrir le tableau de bord Grafana pour afficher votre scène avec l'espace Matterport importé.

Si vous n'avez pas encore configuré AWS IoT TwinMaker avec Grafana, terminez d'abord le processus d'intégration de Grafana. Vous avez deux choix lors de l'intégration AWS IoT TwinMaker à Grafana. Vous pouvez utiliser une instance Grafana autogérée ou vous pouvez utiliser Amazon Managed Grafana.

Consultez la documentation suivante pour en savoir plus sur les options et le processus d'intégration de Grafana :

- AWS IoT TwinMaker Intégration au tableau de [bord Grafana](#).
- [Amazon a géré Grafana](#).
- [Grafana autogéré](#).

## Utilisez les espaces Matterport dans votre application Web AWS IoT TwinMaker

Une fois que vous avez importé votre espace Matterport dans une AWS IoT TwinMaker scène, vous pouvez visualiser cette scène avec l'espace Matterport dans l' AWS IoT application Web de votre kit d'applications.

Consultez la documentation suivante pour en savoir plus sur l'utilisation du kit AWS IoT d'application :

- Pour en savoir plus sur AWS IoT TwinMaker l'utilisation du kit d' AWS IoT applications, consultez [Création d'une application Web personnalisée à l'aide des composants de l' AWS IoT TwinMaker interface utilisateur](#).
- Pour en savoir plus sur l'utilisation du kit AWS IoT d'application, rendez-vous sur la page [Github du kit d'AWS IoT application](#).

- Pour savoir comment démarrer une nouvelle application Web à l'aide du kit AWS IoT d'application, rendez-vous sur la page de documentation officielle de [l'IoT App Kit](#).

# AWS IoT TwinMaker intégration vidéo

Les caméras vidéo offrent une bonne opportunité pour la simulation numérique de jumeaux. Vous pouvez l'utiliser AWS IoT TwinMaker pour simuler l'emplacement et les conditions physiques de votre caméra. Créez des entités AWS IoT TwinMaker pour vos caméras sur site et utilisez des composants vidéo pour diffuser des vidéos en direct et des métadonnées de votre site vers votre AWS IoT TwinMaker scène ou vers un tableau de bord Grafana.

AWS IoT TwinMaker peut capturer des vidéos à partir d'appareils Edge de deux manières. Vous pouvez diffuser des vidéos depuis des appareils Edge grâce au connecteur Edge pour Kinesis Video Stream, ou vous pouvez enregistrer une vidéo sur l'appareil Edge et lancer le téléchargement de vidéos avec des messages MQTT. Utilisez ce composant pour diffuser des données vidéo à partir de vos appareils afin de les utiliser avec AWS IoT des services. Pour générer les ressources nécessaires et déployer le connecteur Edge pour Kinesis Video Streams, reportez-vous à la section [Getting started with the Edge Connector for Kinesis Video Streams](#) sur [GitHub](#) Pour plus d'informations sur le AWS IoT Greengrass composant, consultez la AWS IoT Greengrass documentation sur le [connecteur Edge pour Kinesis Video Streams](#).

Après avoir créé les AWS IoT SiteWise modèles requis et configuré le composant Kinesis Video Streams Greengrass, vous pouvez diffuser ou enregistrer des vidéos en mode périphérique sur votre application de jumeau numérique dans la console. AWS IoT TwinMaker Vous pouvez également consulter les diffusions en direct et les métadonnées de vos appareils dans un tableau de bord Grafana. Pour plus d'informations sur l'intégration de Grafana et AWS IoT TwinMaker, consultez [AWS IoT TwinMaker Intégration au tableau de bord Grafana](#)

## Utilisez le connecteur Edge pour Kinesis Video Stream afin de diffuser de la vidéo AWS IoT TwinMaker

Avec le connecteur Edge pour Kinesis Video Stream, vous pouvez diffuser des vidéos et des données vers une entité de votre AWS IoT TwinMaker scène. Pour ce faire, vous devez utiliser un composant vidéo. Pour créer le composant vidéo à utiliser dans vos scènes, procédez comme suit.

### Conditions préalables

Avant de créer le composant vidéo dans votre AWS IoT TwinMaker scène, assurez-vous d'avoir rempli les conditions préalables suivantes.

- Création des AWS IoT SiteWise modèles et ressources requis pour le connecteur Edge pour le flux vidéo Kinesis. Pour plus d'informations sur la création AWS IoT SiteWise des ressources pour le connecteur, voir [Mise en route avec le connecteur Edge pour Kinesis Video Stream](#).
- Déploiement du connecteur Kinesis Video Stream Edge sur votre AWS IoT Greengrass appareil. [Pour plus d'informations sur le déploiement du composant Kinesis Video Stream Edge Connector, consultez le fichier README de déploiement.](#)

## Création de composants vidéo pour les AWS IoT TwinMaker scènes

Procédez comme suit pour créer le connecteur Edge pour le composant de flux vidéo Kinesis pour votre scène.

1. Dans la AWS IoT TwinMaker console, ouvrez la scène à laquelle vous souhaitez ajouter le composant vidéo.
2. Une fois la scène ouverte, choisissez une entité existante ou créez l'entité à laquelle vous souhaitez ajouter le composant, puis choisissez Ajouter un composant.
3. Dans le volet Ajouter un composant, entrez le nom du composant, et dans le champ Type, choisissez `com.amazon.iotsitewise.connector.edgevideo`.
4. Choisissez un modèle d'actif en sélectionnant le nom du modèle de AWS IoT SiteWise caméra que vous avez créé. Ce nom doit avoir le format suivant `:EdgeConnectorForKVSCameraModel-0abc`, où la chaîne de lettres et de chiffres à la fin correspond à votre propre nom d'actif.
5. Pour Asset, choisissez les éléments de AWS IoT SiteWise caméra à partir desquels vous souhaitez diffuser de la vidéo. Une petite fenêtre apparaît et affiche un aperçu du flux vidéo en cours.

### Note

Pour tester votre streaming vidéo, choisissez Test. Ce test envoie un événement MQTT pour lancer la diffusion vidéo en direct. Patientez quelques instants pour voir la vidéo apparaître dans le lecteur.

6. Pour ajouter le composant vidéo à votre entité, choisissez Ajouter un composant.

# Ajouter des vidéos et des métadonnées issues du flux vidéo Kinesis à un tableau de bord Grafana

Après avoir créé un composant vidéo pour votre entité dans votre AWS IoT TwinMaker scène, vous pouvez configurer le panneau vidéo de Grafana pour voir les diffusions en direct. Assurez-vous d'avoir correctement intégré AWS IoT TwinMaker Grafana. Pour de plus amples informations, veuillez consulter [AWS IoT TwinMaker Intégration au tableau de bord Grafana](#).

## Important

Pour visionner une vidéo dans votre tableau de bord Grafana, vous devez vous assurer que les sources de données Grafana disposent des autorisations IAM appropriées. Pour créer le rôle et la politique requis, voir [Création d'un rôle IAM dans le tableau de bord](#).

Procédez comme suit pour voir Kinesis Video Streams et les métadonnées dans votre tableau de bord Grafana.

1. Ouvrez le AWS IoT TwinMaker tableau de bord.
2. Choisissez Ajouter un panneau, puis Ajouter un panneau vide.

## Note

Pour Grafana v10.4, le lecteur AWS IoT TwinMaker vidéo se trouve sous Widget. Sélectionnez Ajouter >> Widget.

3. Dans la liste des panneaux, choisissez le panneau du lecteur AWS IoT TwinMaker vidéo.
4. Dans le panneau du lecteur AWS IoT TwinMaker vidéo, entrez le nom du flux KinesisVideoStreamName, avec le nom du flux vidéo Kinesis à partir duquel vous souhaitez diffuser de la vidéo.

## Note

Pour diffuser des métadonnées vers le panneau vidéo de Grafana, vous devez d'abord avoir créé une entité avec un composant de streaming vidéo.

5. Facultatif : pour diffuser les métadonnées des AWS IoT SiteWise actifs vers le lecteur vidéo, dans Entity, choisissez l' AWS IoT TwinMaker entité que vous avez créée dans votre AWS IoT TwinMaker scène. Pour le nom du composant, choisissez le composant vidéo que vous avez créé pour l'entité de votre AWS IoT TwinMaker scène.

# Utilisation de la AWS IoT TwinMaker bibliothèque Flink

AWS IoT TwinMaker fournit une bibliothèque Flink que vous pouvez utiliser pour lire et écrire des données dans des magasins de données externes utilisés dans vos jumeaux numériques.

Vous utilisez la bibliothèque AWS IoT TwinMaker Flink en l'installant en tant que connecteur personnalisé dans Managed Service for Apache Flink et en exécutant des requêtes SQL Flink dans un bloc-notes Zeppelin dans Managed Service for Apache Flink. Le bloc-notes peut être transformé en une application de traitement de flux fonctionnant en continu. La bibliothèque utilise des AWS IoT TwinMaker composants pour récupérer les données de votre espace de travail.

La bibliothèque AWS IoT TwinMaker Flink nécessite les éléments suivants.

## Prérequis

1. Un espace de travail entièrement rempli avec des scènes et des composants. Utilisez les types de composants intégrés pour les données provenant des AWS services (AWS IoT SiteWise et de Kinesis Video Streams). Créez des types de composants personnalisés pour les données provenant de sources tierces. Pour plus d'informations, consultez [???](#).
2. Compréhension des blocs-notes Studio avec service géré pour Apache Flink pour Apache Flink. Ces ordinateurs portables sont alimentés par [Apache Zeppelin](#) et utilisent le framework [Apache Flink](#). Pour plus d'informations, voir [Utilisation d'un bloc-notes Studio avec le service géré pour Apache Flink pour Apache Flink](#).

Pour obtenir des instructions sur l'utilisation de la bibliothèque, consultez le [guide de l'utilisateur de la bibliothèque AWS IoT TwinMaker Flink](#).

Pour obtenir des instructions sur la configuration AWS IoT TwinMaker avec le démarrage rapide dans les [AWS IoT TwinMaker exemples](#), consultez le [fichier README de l'application Sample Insights](#).

# Connexion et surveillance AWS IoT TwinMaker

La surveillance joue un rôle important dans le maintien de la fiabilité, de la disponibilité AWS IoT TwinMaker et des performances de vos autres AWS solutions. AWS IoT TwinMaker prend en charge les outils de surveillance suivants pour surveiller le service, signaler un problème et prendre des mesures automatiques le cas échéant :

- Amazon CloudWatch surveille en temps réel vos AWS ressources et les applications que vous utilisez AWS. Vous pouvez collecter et suivre les métriques, créer des tableaux de bord personnalisés et définir des alarmes qui vous avertissent ou prennent des mesures lorsqu'une métrique spécifiée atteint un seuil que vous spécifiez. Par exemple, vous pouvez CloudWatch suivre l'utilisation du processeur ou d'autres indicateurs pour vos instances Amazon EC2 et lancer automatiquement de nouvelles instances en cas de besoin. Pour plus d'informations, consultez le [guide de CloudWatch l'utilisateur Amazon](#).
- Amazon CloudWatch Logs surveille, stocke et fournit un accès à vos fichiers journaux à partir de AWS IoT TwinMaker passerelles et CloudTrail d'autres sources. CloudWatch Les journaux peuvent surveiller les informations contenues dans les fichiers journaux et vous avertir lorsque certains seuils sont atteints. Vous pouvez également archiver vos données de journaux dans une solution de stockage hautement durable. Pour plus d'informations, consultez le [guide de l'utilisateur d'Amazon CloudWatch Logs](#).
- AWS CloudTrail capture les appels d'API et les événements associés effectués par ou pour le compte de votre AWS compte et envoie les fichiers journaux dans un compartiment Amazon S3 que vous spécifiez. Vous pouvez identifier les utilisateurs et les comptes appelés AWS, l'adresse IP source à partir de laquelle les appels ont été effectués et la date des appels. Pour plus d'informations, consultez le [Guide de l'utilisateur AWS CloudTrail](#).

## Rubriques

- [Surveillance AWS IoT TwinMaker à l'aide des CloudWatch métriques Amazon](#)
- [Journalisation des appels d' AWS IoT TwinMaker API avec AWS CloudTrail](#)

# Surveillance AWS IoT TwinMaker à l'aide des CloudWatch métriques Amazon

Vous pouvez effectuer un AWS IoT TwinMaker suivi en utilisant CloudWatch, qui collecte les données brutes et les transforme en indicateurs lisibles en temps quasi réel. Ces statistiques sont enregistrées pour une durée de 15 mois ; par conséquent, vous pouvez accéder aux informations historiques et acquérir un meilleur point de vue de la façon dont votre service ou application web s'exécute. Vous pouvez également définir des alarmes qui surveillent certains seuils et envoient des notifications ou prennent des mesures lorsque ces seuils sont atteints. Pour plus d'informations, consultez le [guide de CloudWatch l'utilisateur Amazon](#).

AWS IoT TwinMaker publie les métriques et les dimensions répertoriées dans les sections suivantes dans l'espace de `AWS/IoTTwinMaker` noms.

## Tip

AWS IoT TwinMaker publie des statistiques à une minute d'intervalle. Lorsque vous visualisez ces mesures sous forme de graphiques dans la CloudWatch console, nous vous recommandons de choisir une période d'une minute pour voir la résolution la plus élevée disponible de vos données métriques.

## Table des matières

- [Métriques](#)

## Métriques

AWS IoT TwinMaker publie les métriques suivantes.

### Métriques

| Métrique                                  | Description  |
|---|--|
| <code>ComponentTypeCreationFailure</code> | <p>Cette métrique indique si la création du type de composant est réussie.</p> <p>La métrique est publiée lorsqu'un type de composant est en <code>CREATING</code> état. Cela se</p> |

| Métrique                   | Description  |
|----------------------------|--|
|                            | <p>produit lorsqu'un type de composant est créé avec les propriétés requises dans l'initialiseur de schéma et que ces propriétés sont instanciées avec des valeurs par défaut.</p> <p>La valeur de la métrique peut indiquer 0 un succès ou 1 un échec.</p> <p>Dimensions : ComponentTypeId, WorkspaceId.</p> <p>Unités : nombre</p>   |
| ComponentTypeUpdateFailure | <p>Cette métrique indique si la mise à jour du type de composant est réussie.</p> <p>La métrique est publiée lorsqu'un type de composant est en UPDATING état. Cela se produit lorsqu'un type de composant est mis à jour avec les propriétés requises dans l'initialiseur de schéma et que ces propriétés sont instanciées avec des valeurs par défaut.</p> <p>La valeur de la métrique peut indiquer 0 un succès ou 1 un échec.</p> <p>Dimensions : ComponentTypeId, WorkspaceId.</p> <p>Unités : nombre</p> |

| Métrique              | Description   |
|-----------------------|---|
| EntityCreationFailure | <p>Cette métrique indique si la création de l'entité est réussie. La métrique est publiée lorsqu'un e entité est en CREATING état. Cela se produit lorsqu'une entité est créée avec un composant.</p> <p>La valeur de la métrique peut indiquer 0 un succès ou 1 un échec.</p> <p>Dimensions : EntityName, EntityId, Workspace Id.</p> <p>Unités : nombre</p> |
| EntityUpdateFailure   | <p>Cette métrique indique si la mise à jour de l'entité est réussie. La métrique est publiée lorsqu'une entité est en UPDATING état. Cela se produit lorsqu'une entité est mise à jour.</p> <p>La valeur de la métrique peut indiquer 0 un succès ou 1 un échec.</p> <p>Dimensions : EntityName, EntityId, Workspace Id.</p> <p>Unités : nombre</p>           |
| EntityDeletionFailure | <p>Cette métrique indique si la suppression de l'entité est réussie. La métrique est publiée lorsqu'une entité est en DELETING état. Cela se produit lorsqu'une entité est supprimée.</p> <p>La valeur de la métrique peut indiquer 0 un succès ou 1 un échec.</p> <p>Dimensions : EntityName, EntityId, Workspace Id.</p> <p>Unités : nombre</p>             |

 Tip

Toutes les métriques sont publiées dans l'espace de AWS/IoTTwinMaker noms.

## Journalisation des appels d' AWS IoT TwinMaker API avec AWS CloudTrail

AWS IoT TwinMaker est intégré à AWS CloudTrail un service qui fournit un enregistrement des actions entreprises par un utilisateur, un rôle ou un AWS service dans AWS IoT TwinMaker. CloudTrail capture les appels d'API AWS IoT TwinMaker sous forme d'événements. Les appels capturés incluent des appels provenant de la AWS IoT TwinMaker console et des appels de code vers les opérations de l' AWS IoT TwinMaker API. Si vous créez un suivi, vous pouvez activer la diffusion continue d' CloudTrail événements vers un compartiment Amazon S3, y compris les événements pour AWS IoT TwinMaker. Si vous ne configurez pas de suivi, vous pouvez toujours consulter les événements les plus récents dans la CloudTrail console dans Historique des événements. À l'aide des informations collectées par CloudTrail, vous pouvez déterminer la demande qui a été faite AWS IoT TwinMaker, l'adresse IP à partir de laquelle la demande a été faite, qui a fait la demande, quand elle a été faite et des détails supplémentaires.

Pour plus d'informations CloudTrail, consultez le [guide de AWS CloudTrail l'utilisateur](#).

### AWS IoT TwinMaker informations dans CloudTrail

Lorsque vous créez votre AWS compte, CloudTrail il est automatiquement activé. CloudTrail enregistre l'activité des événements de support qui se produisent dans AWS IoT TwinMaker, ainsi que d'autres événements AWS de service, dans l'historique des événements. Vous pouvez consulter, rechercher et télécharger les événements récents dans votre AWS compte. Pour plus d'informations, consultez la section [Affichage des événements avec l'historique des CloudTrail événements](#).

Pour un enregistrement continu des événements de votre AWS compte, y compris des événements pour AWS IoT TwinMaker, créez un parcours. Un suivi permet CloudTrail de fournir des fichiers journaux à un compartiment Amazon S3. Par défaut, lorsque vous créez un parcours dans la console, celui-ci s'applique à toutes les AWS régions. CloudTrail enregistre les événements de toutes les régions de la AWS partition et envoie les fichiers journaux dans le compartiment Amazon S3 que vous spécifiez. En outre, vous pouvez configurer d'autres AWS services pour analyser plus en détail les données d'événements collectées dans les CloudTrail journaux et agir en conséquence. Pour plus d'informations, consultez les ressources suivantes :

- [Présentation de la création d'un journal de suivi](#)
- [CloudTrail services et intégrations pris en charge](#)
- [Configuration des notifications Amazon SNS pour CloudTrail](#)
- [Réception de fichiers CloudTrail journaux de plusieurs régions](#) et [réception de fichiers CloudTrail journaux de plusieurs comptes](#)

La plupart AWS IoT TwinMaker des opérations sont enregistrées CloudTrail et documentées dans la [référence de l'AWS IoT TwinMaker API](#).

Les opérations du plan de données suivantes ne sont pas enregistrées par CloudTrail :

- [GetPropertyValue](#)
- [GetPropertyValueHistory](#)
- [BatchPutPropertyValues](#)

Chaque événement ou entrée de journal contient des informations sur la personne ayant initié la demande. Les informations relatives à l'identité permettent de déterminer les éléments suivants :

- Si la demande a été effectuée avec les informations d'identification utilisateur racine ou .
- Si la demande a été effectuée avec les informations d'identification de sécurité temporaires d'un rôle ou d'un utilisateur fédéré.
- Si la demande a été faite par un autre AWS service.

Pour de plus amples informations, veuillez consulter l'[élément userIdentity CloudTrail](#) .

# Sécurité dans AWS IoT TwinMaker

La sécurité du cloud AWS est la priorité absolue. En tant que AWS client, vous bénéficiez de centres de données et d'architectures réseau conçus pour répondre aux exigences des entreprises les plus sensibles en matière de sécurité.

La sécurité est une responsabilité partagée entre vous AWS et vous. Le [modèle de responsabilité partagée](#) décrit ceci comme la sécurité du cloud et la sécurité dans le cloud :

- Sécurité du cloud : AWS est chargée de protéger l'infrastructure qui exécute les AWS services dans le AWS Cloud. AWS vous fournit également des services que vous pouvez utiliser en toute sécurité. Des auditeurs tiers testent et vérifient régulièrement l'efficacité de notre sécurité dans le cadre des programmes de [AWS conformité Programmes](#) de de conformité. Pour en savoir plus sur les programmes de conformité qui s'appliquent à AWS IoT TwinMaker, voir [AWS Services concernés par programme de conformitéAWS](#) .
- Sécurité dans le cloud — Votre responsabilité est déterminée par le AWS service que vous utilisez. Vous êtes également responsable d'autres facteurs, y compris de la sensibilité de vos données, des exigences de votre entreprise, ainsi que de la législation et de la réglementation applicables.

Cette documentation vous aide à comprendre comment appliquer le modèle de responsabilité partagée lors de son utilisation AWS IoT TwinMaker. Les rubriques suivantes expliquent comment procéder à la configuration AWS IoT TwinMaker pour atteindre vos objectifs de sécurité et de conformité. Vous apprendrez également à utiliser d'autres AWS services qui vous aident à surveiller et à sécuriser vos AWS IoT TwinMaker ressources.

## Rubriques

- [Protection des données dans AWS IoT TwinMaker](#)
- [Identity and Access Management pour AWS IoT TwinMaker](#)
- [AWS IoT TwinMaker et points de terminaison VPC d'interface \( \)AWS PrivateLink](#)
- [Validation de conformité pour AWS IoT TwinMaker](#)
- [Résilience dans AWS IoT TwinMaker](#)
- [Sécurité de l'infrastructure dans AWS IoT TwinMaker](#)

# Protection des données dans AWS IoT TwinMaker

Le [modèle de responsabilité AWS partagée](#) de s'applique à la protection des données dans AWS IoT TwinMaker. Comme décrit dans ce modèle, AWS est chargé de protéger l'infrastructure mondiale qui gère tous les AWS Cloud. La gestion du contrôle de votre contenu hébergé sur cette infrastructure relève de votre responsabilité. Vous êtes également responsable des tâches de configuration et de gestion de la sécurité des Services AWS que vous utilisez. Pour plus d'informations sur la confidentialité des données, consultez [Questions fréquentes \(FAQ\) sur la confidentialité des données](#). Pour en savoir plus sur la protection des données en Europe, consultez le billet de blog [Modèle de responsabilité partagée d'AWS et RGPD \(Règlement général sur la protection des données\)](#) sur le Blog de sécuritéAWS .

À des fins de protection des données, nous vous recommandons de protéger les Compte AWS informations d'identification et de configurer les utilisateurs individuels avec AWS IAM Identity Center ou Gestion des identités et des accès AWS (IAM). Ainsi, chaque utilisateur se voit attribuer uniquement les autorisations nécessaires pour exécuter ses tâches. Nous vous recommandons également de sécuriser vos données comme indiqué ci-dessous :

- Utilisez l'authentification multifactorielle (MFA) avec chaque compte.
- SSL/TLS À utiliser pour communiquer avec AWS les ressources. Nous exigeons TLS 1.2 et recommandons TLS 1.3.
- Configurez l'API et la journalisation de l'activité des utilisateurs avec AWS CloudTrail. Pour plus d'informations sur l'utilisation des CloudTrail sentiers pour capturer AWS des activités, consultez la section [Utilisation des CloudTrail sentiers](#) dans le guide de AWS CloudTrail l'utilisateur.
- Utilisez des solutions de AWS chiffrement, ainsi que tous les contrôles de sécurité par défaut qu'ils contiennent Services AWS.
- Utilisez des services de sécurité gérés avancés tels qu'Amazon Macie, qui contribuent à la découverte et à la sécurisation des données sensibles stockées dans Amazon S3.
- Si vous avez besoin de modules cryptographiques validés par la norme FIPS 140-3 pour accéder AWS via une interface de ligne de commande ou une API, utilisez un point de terminaison FIPS. Pour plus d'informations sur les points de terminaison FIPS disponibles, consultez [Norme FIPS \(Federal Information Processing Standard\) 140-3](#).

Nous vous recommandons fortement de ne jamais placer d'informations confidentielles ou sensibles, telles que les adresses e-mail de vos clients, dans des balises ou des champs de texte libre tels que le champ Nom. Cela inclut lorsque vous travaillez avec AWS IoT TwinMaker ou d'autres Services

AWS utilisateurs de la console, de l'API ou AWS SDKs. AWS CLI Toutes les données que vous entrez dans des balises ou des champs de texte de forme libre utilisés pour les noms peuvent être utilisées à des fins de facturation ou dans les journaux de diagnostic. Si vous fournissez une adresse URL à un serveur externe, nous vous recommandons fortement de ne pas inclure d'informations d'identification dans l'adresse URL permettant de valider votre demande adressée à ce serveur.

## Chiffrement au repos

AWS IoT TwinMaker stocke les informations de votre espace de travail dans un compartiment Amazon S3 que le service crée pour vous, si vous le souhaitez. Le chiffrement côté serveur par défaut est activé dans le compartiment que le service crée pour vous. Si vous choisissez d'utiliser votre propre compartiment Amazon S3 lorsque vous créez un nouvel espace de travail, nous vous recommandons d'activer le chiffrement côté serveur par défaut. Pour plus d'informations sur le chiffrement par défaut dans Amazon S3, consultez [Configuration du comportement de chiffrement côté serveur par défaut pour les compartiments Amazon S3](#).

## Chiffrement en transit

Toutes les données envoyées AWS IoT TwinMaker sont envoyées via une connexion TLS à l'aide du protocole HTTPS. Elles sont donc sécurisées par défaut pendant le transit.

### Note

Nous vous recommandons d'utiliser le protocole HTTPS sur les adresses de compartiment Amazon S3 comme moyen de contrôle pour appliquer le chiffrement en transit lorsque vous AWS IoT TwinMaker interagissez avec un compartiment Amazon S3. Pour plus d'informations sur les compartiments Amazon S3, consultez [Création, configuration et utilisation des compartiments Amazon S3](#).

## Identity and Access Management pour AWS IoT TwinMaker

Gestion des identités et des accès AWS (IAM) est un outil Service AWS qui permet à un administrateur de contrôler en toute sécurité l'accès aux AWS ressources. Les administrateurs IAM contrôlent qui peut être authentifié (connecté) et autorisé (autorisé) à utiliser AWS IoT TwinMaker les ressources. IAM est un Service AWS outil que vous pouvez utiliser sans frais supplémentaires.

## Rubriques

- [Public ciblé](#)
- [Authentification par des identités](#)
- [Gestion de l'accès à l'aide de politiques](#)
- [Comment AWS IoT TwinMaker fonctionne avec IAM](#)
- [Exemples de politiques basées sur l'identité pour AWS IoT TwinMaker](#)
- [Résolution des problèmes AWS IoT TwinMaker d'identité et d'accès](#)
- [Utilisation de rôles liés à un service pour AWS IoT TwinMaker](#)
- [AWS politiques gérées pour AWS IoT TwinMaker](#)

## Public ciblé

La façon dont vous utilisez Gestion des identités et des accès AWS (IAM) varie en fonction de votre rôle :

- Utilisateur du service : demandez des autorisations à votre administrateur si vous ne pouvez pas accéder aux fonctionnalités (voir [Résolution des problèmes AWS IoT TwinMaker d'identité et d'accès](#))
- Administrateur du service : déterminez l'accès des utilisateurs et soumettez les demandes d'autorisation (voir [Comment AWS IoT TwinMaker fonctionne avec IAM](#))
- Administrateur IAM : rédigez des politiques pour gérer l'accès (voir [Exemples de politiques basées sur l'identité pour AWS IoT TwinMaker](#))

## Authentification par des identités

L'authentification est la façon dont vous vous connectez à AWS l'aide de vos informations d'identification. Vous devez être authentifié en tant qu'utilisateur IAM ou en assumant un rôle IAM.

Utilisateur racine d'un compte AWS

Vous pouvez vous connecter en tant qu'identité fédérée à l'aide d'informations d'identification provenant d'une source d'identité telle que AWS IAM Identity Center (IAM Identity Center), d'une authentification unique ou d'informations d'identification. Google/Facebook Pour plus d'informations sur la connexion, consultez [Connexion à votre Compte AWS](#) dans le Guide de l'utilisateur Connexion à AWS .

Pour l'accès par programmation, AWS fournit un SDK et une CLI pour signer les demandes de manière cryptographique. Pour plus d'informations, consultez [Signature AWS Version 4 pour les demandes d'API](#) dans le Guide de l'utilisateur IAM.

## Compte AWS utilisateur root

Lorsque vous créez un Compte AWS, vous commencez par une seule identité de connexion appelée utilisateur Compte AWS root qui dispose d'un accès complet à toutes Services AWS les ressources. Il est vivement déconseillé d'utiliser l'utilisateur racine pour vos tâches quotidiennes. Pour les tâches qui requièrent des informations d'identification de l'utilisateur racine, consultez [Tâches qui requièrent les informations d'identification de l'utilisateur racine](#) dans le Guide de l'utilisateur IAM.

## Identité fédérée

Il est recommandé d'obliger les utilisateurs humains à utiliser la fédération avec un fournisseur d'identité pour accéder à Services AWS l'aide d'informations d'identification temporaires.

Une identité fédérée est un utilisateur provenant de l'annuaire de votre entreprise, de votre fournisseur d'identité Web ou Directory Service qui y accède à Services AWS l'aide d'informations d'identification provenant d'une source d'identité. Les identités fédérées assument des rôles qui fournissent des informations d'identification temporaires.

Pour une gestion des accès centralisée, nous vous recommandons d'utiliser AWS IAM Identity Center. Pour plus d'informations, consultez [Qu'est-ce que IAM Identity Center ?](#) dans le Guide de l'utilisateur AWS IAM Identity Center .

## Utilisateurs et groupes IAM

Un [utilisateur IAM](#) est une identité qui dispose d'autorisations spécifiques pour une seule personne ou application. Nous vous recommandons d'utiliser ces informations d'identification temporaires au lieu des utilisateurs IAM avec des informations d'identification à long terme. Pour plus d'informations, voir [Exiger des utilisateurs humains qu'ils utilisent la fédération avec un fournisseur d'identité pour accéder à AWS l'aide d'informations d'identification temporaires](#) dans le guide de l'utilisateur IAM.

[Les groupes IAM](#) spécifient une collection d'utilisateurs IAM et permettent de gérer plus facilement les autorisations pour de grands ensembles d'utilisateurs. Pour plus d'informations, consultez [Cas d'utilisation pour les utilisateurs IAM](#) dans le Guide de l'utilisateur IAM.

## Rôles IAM

Un [rôle IAM](#) est une identité dotée d'autorisations spécifiques qui fournit des informations d'identification temporaires. Vous pouvez assumer un rôle en [passant d'un rôle d'utilisateur à un rôle IAM \(console\)](#) ou en appelant une opération d' AWS API AWS CLI ou d'API. Pour plus d'informations, consultez [Méthodes pour endosser un rôle](#) dans le Guide de l'utilisateur IAM.

Les rôles IAM sont utiles pour l'accès des utilisateurs fédérés, les autorisations temporaires des utilisateurs IAM, les accès intercompte, les accès entre services et les applications exécutées sur Amazon EC2. Pour plus d'informations, consultez [Accès intercompte aux ressources dans IAM](#) dans le Guide de l'utilisateur IAM.

## Gestion de l'accès à l'aide de politiques

Vous contrôlez l'accès en AWS créant des politiques et en les associant à AWS des identités ou à des ressources. Une politique définit les autorisations lorsqu'elles sont associées à une identité ou à une ressource. AWS évalue ces politiques lorsqu'un directeur fait une demande. La plupart des politiques sont stockées AWS sous forme de documents JSON. Pour plus d'informations les documents de politique JSON, consultez [Vue d'ensemble des politiques JSON](#) dans le Guide de l'utilisateur IAM.

À l'aide de politiques, les administrateurs précisent qui a accès à quoi en définissant quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

Par défaut, les utilisateurs et les rôles ne disposent d'aucune autorisation. Un administrateur IAM crée des politiques IAM et les ajoute aux rôles, que les utilisateurs peuvent ensuite assumer. Les politiques IAM définissent les autorisations quelle que soit la méthode que vous utilisez pour exécuter l'opération.

### Politiques basées sur l'identité

Les stratégies basées sur l'identité sont des documents de stratégie d'autorisations JSON que vous attachez à une identité (utilisateur, groupe ou rôle). Ces politiques contrôlent les actions que peuvent exécuter ces identités, sur quelles ressources et dans quelles conditions. Pour découvrir comment créer une politique basée sur l'identité, consultez [Définition d'autorisations IAM personnalisées avec des politiques gérées par le client](#) dans le Guide de l'utilisateur IAM.

Les politiques basées sur l'identité peuvent être des politiques intégrées (intégrées directement dans une seule identité) ou des politiques gérées (politiques autonomes associées à plusieurs identités).

Pour découvrir comment choisir entre des politiques gérées et en ligne, consultez [Choix entre les politiques gérées et les politiques en ligne](#) dans le Guide de l'utilisateur IAM.

## Politiques basées sur les ressources

Les politiques basées sur les ressources sont des documents de politique JSON que vous attachez à une ressource. Les exemples incluent les politiques de confiance de rôle IAM et les stratégies de compartiment Amazon S3. Dans les services qui sont compatibles avec les politiques basées sur les ressources, les administrateurs de service peuvent les utiliser pour contrôler l'accès à une ressource spécifique. Vous devez [spécifier un principal](#) dans une politique basée sur les ressources.

Les politiques basées sur les ressources sont des politiques en ligne situées dans ce service. Vous ne pouvez pas utiliser les politiques AWS gérées par IAM dans une stratégie basée sur les ressources.

## Autres types de politique

AWS prend en charge des types de politiques supplémentaires qui peuvent définir les autorisations maximales accordées par les types de politiques les plus courants :

- Limites d'autorisations : une limite des autorisations définit le nombre maximum d'autorisations qu'une politique basée sur l'identité peut accorder à une entité IAM. Pour plus d'informations, consultez [Limites d'autorisations pour des entités IAM](#) dans le Guide de l'utilisateur IAM.
- Politiques de contrôle des services (SCPs) — Spécifiez les autorisations maximales pour une organisation ou une unité organisationnelle dans AWS Organizations. Pour plus d'informations, consultez [Politiques de contrôle de service](#) dans le Guide de l'utilisateur AWS Organizations .
- Politiques de contrôle des ressources (RCPs) : définissez le maximum d'autorisations disponibles pour les ressources de vos comptes. Pour plus d'informations, voir [Politiques de contrôle des ressources \(RCPs\)](#) dans le guide de AWS Organizations l'utilisateur.
- Politiques de session : politiques avancées que vous passez en tant que paramètre lorsque vous créez par programmation une session temporaire pour un rôle ou un utilisateur fédéré. Pour plus d'informations, consultez [Politiques de session](#) dans le Guide de l'utilisateur IAM.

## Plusieurs types de politique

Lorsque plusieurs types de politiques s'appliquent à la requête, les autorisations en résultant sont plus compliquées à comprendre. Pour savoir comment AWS déterminer s'il faut autoriser

une demande lorsque plusieurs types de politiques sont impliqués, consultez la section [Logique d'évaluation des politiques](#) dans le guide de l'utilisateur IAM.

## Comment AWS IoT TwinMaker fonctionne avec IAM

Avant d'utiliser IAM pour gérer l'accès à AWS IoT TwinMaker, découvrez les fonctionnalités IAM disponibles. AWS IoT TwinMaker

Fonctionnalités IAM que vous pouvez utiliser avec AWS IoT TwinMaker

| Fonctionnalité IAM   | AWS IoT TwinMaker soutien |
|--|---------------------------|
| <a href="#">Politiques basées sur l'identité</a>           | Oui                       |
| <a href="#">Politiques basées sur les ressources</a>       | Non                       |
| <a href="#">Actions de politique</a>                       | Oui                       |
| <a href="#">Ressources de politique</a>                    | Oui                       |
| <a href="#">Clés de condition de politique</a>             | Oui                       |
| <a href="#">ACLs</a>                                       | Non                       |
| <a href="#">ABAC (identifications dans les politiques)</a> | Partielle                 |
| <a href="#">Informations d'identification temporaires</a>  | Oui                       |
| <a href="#">Autorisations de principal</a>                 | Oui                       |
| <a href="#">Rôles de service</a>                           | Oui                       |
| <a href="#">Rôles liés à un service</a>                    | Non                       |

Pour obtenir une vue d'ensemble de la façon dont AWS IoT TwinMaker les autres AWS services fonctionnent avec la plupart des fonctionnalités IAM, consultez la section [AWS Services compatibles avec IAM](#) dans le Guide de l'AWS IAM Identity Center utilisateur.

## Politiques basées sur l'identité pour AWS IoT TwinMaker

Prend en charge les politiques basées sur l'identité : oui

Les politiques basées sur l'identité sont des documents de politique d'autorisations JSON que vous pouvez attacher à une identité telle qu'un utilisateur, un groupe d'utilisateurs ou un rôle IAM. Ces politiques contrôlent quel type d'actions des utilisateurs et des rôles peuvent exécuter, sur quelles ressources et dans quelles conditions. Pour découvrir comment créer une politique basée sur l'identité, consultez [Définition d'autorisations IAM personnalisées avec des politiques gérées par le client](#) dans le Guide de l'utilisateur IAM.

Avec les politiques IAM basées sur l'identité, vous pouvez spécifier des actions et ressources autorisées ou refusées, ainsi que les conditions dans lesquelles les actions sont autorisées ou refusées. Pour découvrir tous les éléments que vous utilisez dans une politique JSON, consultez [Références des éléments de politique JSON IAM](#) dans le Guide de l'utilisateur IAM.

Exemples de politiques basées sur l'identité pour AWS IoT TwinMaker

Pour consulter des exemples de politiques AWS IoT TwinMaker basées sur l'identité, consultez [Exemples de politiques basées sur l'identité pour AWS IoT TwinMaker](#)

## Politiques basées sur les ressources au sein de AWS IoT TwinMaker

Prend en charge les politiques basées sur les ressources : non

Les politiques basées sur les ressources sont des documents de politique JSON que vous attachez à une ressource. Par exemple, les politiques de confiance de rôle IAM et les politiques de compartiment Amazon S3 sont des politiques basées sur les ressources. Dans les services qui sont compatibles avec les politiques basées sur les ressources, les administrateurs de service peuvent les utiliser pour contrôler l'accès à une ressource spécifique. Pour la ressource dans laquelle se trouve la politique, cette dernière définit quel type d'actions un principal spécifié peut effectuer sur cette ressource et dans quelles conditions. Vous devez [spécifier un principal](#) dans une politique basée sur les ressources. Les principaux peuvent inclure des comptes, des utilisateurs, des rôles, des utilisateurs fédérés ou. Services AWS

Pour permettre un accès intercompte, vous pouvez spécifier un compte entier ou des entités IAM dans un autre compte en tant que principal dans une politique basée sur les ressources. Pour plus d'informations, consultez [Accès intercompte aux ressources dans IAM](#) dans le Guide de l'utilisateur IAM.

## Actions politiques pour AWS IoT TwinMaker

Prend en charge les actions de politique : oui

Les administrateurs peuvent utiliser les politiques AWS JSON pour spécifier qui a accès à quoi. C'est-à-dire, quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

L'élément `Action` d'une politique JSON décrit les actions que vous pouvez utiliser pour autoriser ou refuser l'accès à une politique. Intégration d'actions dans une politique afin d'accorder l'autorisation d'exécuter les opérations associées.

Pour consulter la liste des AWS IoT TwinMaker actions, reportez-vous à la section [Actions définies par AWS IoT TwinMaker](#) dans la référence d'autorisation de service.

Les actions de politique en AWS IoT TwinMaker cours utilisent le préfixe suivant avant l'action :

```
iottwinmaker
```

Pour indiquer plusieurs actions dans une seule déclaration, séparez-les par des virgules.

```
"Action": [  
  "iottwinmaker:action1",  
  "iottwinmaker:action2"  
]
```

Pour consulter des exemples de politiques AWS IoT TwinMaker basées sur l'identité, consultez. [Exemples de politiques basées sur l'identité pour AWS IoT TwinMaker](#)

## Ressources politiques pour AWS IoT TwinMaker

Prend en charge les ressources de politique : oui

Les administrateurs peuvent utiliser les politiques AWS JSON pour spécifier qui a accès à quoi. C'est-à-dire, quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

L'élément de politique JSON `Resource` indique le ou les objets auxquels l'action s'applique. Il est recommandé de définir une ressource à l'aide de son [Amazon Resource Name \(ARN\)](#). Pour les

actions qui ne sont pas compatibles avec les autorisations de niveau ressource, utilisez un caractère générique (\*) afin d'indiquer que l'instruction s'applique à toutes les ressources.

```
"Resource": "*"
```

Pour consulter la liste des types de AWS IoT TwinMaker ressources et leurs caractéristiques ARNs, voir [Ressources définies par AWS IoT TwinMaker](#) dans la référence d'autorisation de service. Pour savoir grâce à quelles actions vous pouvez spécifier l'ARN de chaque ressource, consultez [Actions définies par AWS IoT TwinMaker](#).

Pour consulter des exemples de politiques AWS IoT TwinMaker basées sur l'identité, consultez. [Exemples de politiques basées sur l'identité pour AWS IoT TwinMaker](#)

## Clés de conditions de politique pour AWS IoT TwinMaker

Prend en charge les clés de condition de politique spécifiques au service : oui

Les administrateurs peuvent utiliser les politiques AWS JSON pour spécifier qui a accès à quoi. C'est-à-dire, quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

L'élément `Condition` indique à quel moment les instructions s'exécutent en fonction de critères définis. Vous pouvez créer des expressions conditionnelles qui utilisent des [opérateurs de condition](#), tels que les signes égal ou inférieur à, pour faire correspondre la condition de la politique aux valeurs de la demande. Pour voir toutes les clés de condition AWS globales, voir les clés de [contexte de condition AWS globales](#) dans le guide de l'utilisateur IAM.

Pour consulter la liste des clés de AWS IoT TwinMaker condition, voir [Clés de condition pour AWS IoT TwinMaker](#) la référence d'autorisation de service. Pour savoir avec quelles actions et ressources vous pouvez utiliser une clé de condition, consultez la section [Actions définies par AWS IoT TwinMaker](#).

Pour consulter des exemples de politiques AWS IoT TwinMaker basées sur l'identité, consultez. [Exemples de politiques basées sur l'identité pour AWS IoT TwinMaker](#)

## Listes de contrôle d'accès (ACLs) dans AWS IoT TwinMaker

Supports ACLs : Non

Les listes de contrôle d'accès (ACLs) contrôlent les principaux (membres du compte, utilisateurs ou rôles) autorisés à accéder à une ressource. ACLs sont similaires aux politiques basées sur les ressources, bien qu'elles n'utilisent pas le format de document de politique JSON.

## Contrôle d'accès basé sur les attributs (ABAC) avec AWS IoT TwinMaker

Prend en charge ABAC (identifications dans les politiques) : partiellement

Le contrôle d'accès par attributs (ABAC) est une stratégie d'autorisation qui définit les autorisations en fonction des attributs nommés balise. Vous pouvez associer des balises aux entités et aux AWS ressources IAM, puis concevoir des politiques ABAC pour autoriser les opérations lorsque la balise du principal correspond à la balise de la ressource.

Pour contrôler l'accès basé sur des étiquettes, vous devez fournir les informations d'étiquette dans l'[élément de condition](#) d'une politique utilisant les clés de condition `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` ou `aws:TagKeys`.

Si un service prend en charge les trois clés de condition pour tous les types de ressources, alors la valeur pour ce service est Oui. Si un service prend en charge les trois clés de condition pour certains types de ressources uniquement, la valeur est Partielle.

Pour plus d'informations sur ABAC, consultez [Définition d'autorisations avec l'autorisation ABAC](#) dans le Guide de l'utilisateur IAM. Pour accéder à un didacticiel décrivant les étapes de configuration de l'ABAC, consultez [Utilisation du contrôle d'accès par attributs \(ABAC\)](#) dans le Guide de l'utilisateur IAM.

## Utilisation d'informations d'identification temporaires avec AWS IoT TwinMaker

Prend en charge les informations d'identification temporaires : oui

Les informations d'identification temporaires fournissent un accès à court terme aux AWS ressources et sont automatiquement créées lorsque vous utilisez la fédération ou que vous changez de rôle. AWS recommande de générer dynamiquement des informations d'identification temporaires au lieu d'utiliser des clés d'accès à long terme. Pour plus d'informations, consultez [Informations d'identification de sécurité temporaires dans IAM](#) et [Services AWS compatibles avec IAM](#) dans le Guide de l'utilisateur IAM.

## Autorisations principales interservices pour AWS IoT TwinMaker

Prend en charge les sessions d'accès direct (FAS) : oui

Les sessions d'accès direct (FAS) utilisent les autorisations du principal appelant et Service AWS, combinées Service AWS à la demande d'envoi de demandes aux services en aval. Pour plus de détails sur la politique relative à la transmission de demandes FAS, consultez la section [Sessions de transmission d'accès](#).

## Rôles de service pour AWS IoT TwinMaker

Prend en charge les rôles de service : oui

Un rôle de service est un [rôle IAM](#) qu'un service endosse pour accomplir des actions en votre nom. Un administrateur IAM peut créer, modifier et supprimer un rôle de service à partir d'IAM. Pour plus d'informations, consultez [Création d'un rôle pour la délégation d'autorisations à un Service AWS](#) dans le Guide de l'utilisateur IAM.

### Warning

La modification des autorisations associées à un rôle de service peut perturber AWS IoT TwinMaker les fonctionnalités. Modifiez les rôles de service uniquement lorsque AWS IoT TwinMaker vous recevez des instructions à cet effet.

## Rôles liés à un service pour AWS IoT TwinMaker

Prend en charge les rôles liés à un service : non

Un rôle lié à un service est un type de rôle de service lié à un. Service AWS Le service peut endosser le rôle afin d'effectuer une action en votre nom. Les rôles liés à un service apparaissent dans votre Compte AWS répertoire et appartiennent au service. Un administrateur IAM peut consulter, mais ne peut pas modifier, les autorisations concernant les rôles liés à un service.

Pour plus d'informations sur la création ou la gestion des rôles liés à un service, consultez [Services AWS qui fonctionnent avec IAM](#). Recherchez un service dans le tableau qui inclut un Yes dans la colonne Rôle lié à un service. Choisissez le lien Oui pour consulter la documentation du rôle lié à ce service.

## Exemples de politiques basées sur l'identité pour AWS IoT TwinMaker

Par défaut, les utilisateurs et les rôles ne sont pas autorisés à créer ou modifier les ressources AWS IoT TwinMaker . Pour octroyer aux utilisateurs des autorisations d'effectuer des actions sur les ressources dont ils ont besoin, un administrateur IAM peut créer des politiques IAM.

Pour apprendre à créer une politique basée sur l'identité IAM à l'aide de ces exemples de documents de politique JSON, consultez [Création de politiques IAM \(console\)](#) dans le Guide de l'utilisateur IAM.

Pour plus de détails sur les actions et les types de ressources définis par AWS IoT TwinMaker, y compris le format de ARNs pour chacun des types de ressources, voir [Actions, ressources et clés de condition AWS IoT TwinMaker](#) dans la référence d'autorisation de service.

## Rubriques

- [Bonnes pratiques en matière de politiques](#)
- [Utilisation de la AWS IoT TwinMaker console](#)
- [Autorisation accordée aux utilisateurs pour afficher leurs propres autorisations](#)

## Bonnes pratiques en matière de politiques

Les politiques basées sur l'identité déterminent si quelqu'un peut créer, accéder ou supprimer AWS IoT TwinMaker des ressources dans votre compte. Ces actions peuvent entraîner des frais pour votre Compte AWS. Lorsque vous créez ou modifiez des politiques basées sur l'identité, suivez ces instructions et recommandations :

- Commencez AWS par les politiques gérées et passez aux autorisations du moindre privilège : pour commencer à accorder des autorisations à vos utilisateurs et à vos charges de travail, utilisez les politiques AWS gérées qui accordent des autorisations pour de nombreux cas d'utilisation courants. Ils sont disponibles dans votre Compte AWS. Nous vous recommandons de réduire davantage les autorisations en définissant des politiques gérées par les AWS clients spécifiques à vos cas d'utilisation. Pour plus d'informations, consultez [politiques gérées par AWS](#) ou [politiques gérées par AWS pour les activités professionnelles](#) dans le Guide de l'utilisateur IAM.
- Accordez les autorisations de moindre privilège : lorsque vous définissez des autorisations avec des politiques IAM, accordez uniquement les autorisations nécessaires à l'exécution d'une seule tâche. Pour ce faire, vous définissez les actions qui peuvent être entreprises sur des ressources spécifiques dans des conditions spécifiques, également appelées autorisations de moindre privilège. Pour plus d'informations sur l'utilisation d'IAM pour appliquer des autorisations, consultez [politiques et autorisations dans IAM](#) dans le Guide de l'utilisateur IAM.
- Utilisez des conditions dans les politiques IAM pour restreindre davantage l'accès : vous pouvez ajouter une condition à vos politiques afin de limiter l'accès aux actions et aux ressources. Par exemple, vous pouvez écrire une condition de politique pour spécifier que toutes les demandes doivent être envoyées via SSL. Vous pouvez également utiliser des conditions pour accorder

l'accès aux actions de service si elles sont utilisées par le biais d'un service spécifique Service AWS, tel que CloudFormation. Pour plus d'informations, consultez [Conditions pour éléments de politique JSON IAM](#) dans le Guide de l'utilisateur IAM.

- Utilisez l'Analyseur d'accès IAM pour valider vos politiques IAM afin de garantir des autorisations sécurisées et fonctionnelles : l'Analyseur d'accès IAM valide les politiques nouvelles et existantes de manière à ce que les politiques IAM respectent le langage de politique IAM (JSON) et les bonnes pratiques IAM. IAM Access Analyzer fournit plus de 100 vérifications de politiques et des recommandations exploitables pour vous aider à créer des politiques sécurisées et fonctionnelles. Pour plus d'informations, consultez [Validation de politiques avec IAM Access Analyzer](#) dans le Guide de l'utilisateur IAM.
- Exiger l'authentification multifactorielle (MFA) : si vous avez un scénario qui nécessite des utilisateurs IAM ou un utilisateur root, activez l'authentification MFA pour une sécurité accrue. Compte AWS Pour exiger la MFA lorsque des opérations d'API sont appelées, ajoutez des conditions MFA à vos politiques. Pour plus d'informations, consultez [Sécurisation de l'accès aux API avec MFA](#) dans le Guide de l'utilisateur IAM.

Pour plus d'informations sur les bonnes pratiques dans IAM, consultez [Bonnes pratiques de sécurité dans IAM](#) dans le Guide de l'utilisateur IAM.

## Utilisation de la AWS IoT TwinMaker console

Pour accéder à la AWS IoT TwinMaker console, vous devez disposer d'un ensemble minimal d'autorisations. Ces autorisations doivent vous permettre de répertorier et d'afficher les détails AWS IoT TwinMaker des ressources de votre Compte AWS. Si vous créez une politique basée sur l'identité qui est plus restrictive que l'ensemble minimum d'autorisations requis, la console ne fonctionnera pas comme prévu pour les entités (utilisateurs ou rôles) tributaires de cette politique.

Il n'est pas nécessaire d'accorder des autorisations de console minimales aux utilisateurs qui appellent uniquement l'API AWS CLI ou l' AWS API. Autorisez plutôt l'accès à uniquement aux actions qui correspondent à l'opération d'API qu'ils tentent d'effectuer.

Pour garantir que les utilisateurs et les rôles peuvent toujours utiliser la AWS IoT TwinMaker console, associez également la politique AWS IoT TwinMaker ConsoleAccess ou la politique ReadOnly AWS gérée aux entités. Pour plus d'informations, consultez la section [Ajouter des autorisations à un utilisateur](#) dans le Guide de AWS IAM Identity Center l'utilisateur.

## Autorisation accordée aux utilisateurs pour afficher leurs propres autorisations

Cet exemple montre comment créer une politique qui permet aux utilisateurs IAM d'afficher les politiques en ligne et gérées attachées à leur identité d'utilisateur. Cette politique inclut les autorisations permettant d'effectuer cette action sur la console ou par programmation à l'aide de l'API AWS CLI or 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": "*"
    }
  ]
}
```

## Résolution des problèmes AWS IoT TwinMaker d'identité et d'accès

Utilisez les informations suivantes pour vous aider à diagnostiquer et à résoudre les problèmes courants que vous pouvez rencontrer lorsque vous travaillez avec AWS IoT TwinMaker IAM.

### Rubriques

- [Je ne suis pas autorisé à effectuer une action dans AWS IoT TwinMaker](#)
- [Je ne suis pas autorisé à effectuer iam : PassRole](#)
- [Je souhaite permettre à des personnes extérieures Compte AWS à moi d'accéder à mes AWS IoT TwinMaker ressources](#)

### Je ne suis pas autorisé à effectuer une action dans AWS IoT TwinMaker

Si vous recevez une erreur qui indique que vous n'êtes pas autorisé à effectuer une action, vos politiques doivent être mises à jour afin de vous permettre d'effectuer l'action.

L'exemple d'erreur suivant se produit quand l'utilisateur IAM mateojackson tente d'utiliser la console pour afficher des informations détaillées sur une ressource *my-example-widget* fictive, mais ne dispose pas des autorisations `iot twinmaker:GetWidget` fictives.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
iot twinmaker:GetWidget on resource: my-example-widget
```

Dans ce cas, la politique qui s'applique à l'utilisateur mateojackson doit être mise à jour pour autoriser l'accès à la ressource *my-example-widget* à l'aide de l'action `iot twinmaker:GetWidget`.

Si vous avez besoin d'aide, contactez votre AWS administrateur. Votre administrateur vous a fourni vos informations d'identification de connexion.

### Je ne suis pas autorisé à effectuer iam : PassRole

Si vous recevez une erreur selon laquelle vous n'êtes pas autorisé à exécuter `iam:PassRole` l'action, vos stratégies doivent être mises à jour afin de vous permettre de transmettre un rôle à AWS IoT TwinMaker.

Certains services AWS permettent de transmettre un rôle existant à ce service au lieu de créer un nouveau rôle de service ou un rôle lié à un service. Pour ce faire, vous devez disposer des autorisations nécessaires pour transmettre le rôle au service.

L'exemple d'erreur suivant se produit lorsqu'un utilisateur IAM nommé `marymajor` essaie d'utiliser la console pour exécuter une action dans AWS IoT TwinMaker. Toutefois, l'action nécessite que le service ait des autorisations accordées par une fonction de service. Mary n'est pas autorisée à transmettre le rôle au service.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
iam:PassRole
```

Dans ce cas, les politiques de Mary doivent être mises à jour pour lui permettre d'exécuter l'action `iam:PassRole`.

Si vous avez besoin d'aide, contactez votre AWS administrateur. Votre administrateur vous a fourni vos informations d'identification de connexion.

## Je souhaite permettre à des personnes extérieures Compte AWS à moi d'accéder à mes AWS IoT TwinMaker ressources

Vous pouvez créer un rôle que les utilisateurs provenant d'autres comptes ou les personnes extérieures à votre organisation pourront utiliser pour accéder à vos ressources. Vous pouvez spécifier qui est autorisé à assumer le rôle. Pour les services qui prennent en charge les politiques basées sur les ressources ou les listes de contrôle d'accès (ACLs), vous pouvez utiliser ces politiques pour autoriser les utilisateurs à accéder à vos ressources.

Pour plus d'informations, consultez les éléments suivants :

- Pour savoir si ces fonctionnalités sont prises AWS IoT TwinMaker en charge, consultez [Comment AWS IoT TwinMaker fonctionne avec IAM](#).
- Pour savoir comment fournir l'accès à vos ressources sur celles Comptes AWS que vous possédez, consultez la section [Fournir l'accès à un utilisateur IAM dans un autre utilisateur Compte AWS que vous possédez](#) dans le Guide de l'utilisateur IAM.
- Pour savoir comment fournir l'accès à vos ressources à des tiers Comptes AWS, consultez la section [Fournir un accès à des ressources Comptes AWS détenues par des tiers](#) dans le guide de l'utilisateur IAM.
- Pour savoir comment fournir un accès par le biais de la fédération d'identité, consultez [Fournir un accès à des utilisateurs authentifiés en externe \(fédération d'identité\)](#) dans le Guide de l'utilisateur IAM.

- Pour en savoir plus sur la différence entre l'utilisation des rôles et des politiques basées sur les ressources pour l'accès intercompte, consultez [Accès intercompte aux ressources dans IAM](#) dans le Guide de l'utilisateur IAM.

## Utilisation de rôles liés à un service pour AWS IoT TwinMaker

AWS IoT TwinMaker utilise des Gestion des identités et des accès AWS rôles liés à un [service](#) (IAM). Un rôle lié à un service est un type unique de rôle IAM directement lié à. AWS IoT TwinMaker Les rôles liés au service sont prédéfinis par AWS IoT TwinMaker et incluent toutes les autorisations dont le service a besoin pour appeler d'autres AWS services en votre nom.

Un rôle lié à un service facilite la configuration AWS IoT TwinMaker car vous n'avez pas à ajouter manuellement les autorisations nécessaires. AWS IoT TwinMaker définit les autorisations associées à ses rôles liés aux services et, sauf indication contraire, seul AWS IoT TwinMaker peut assumer ses rôles. Les autorisations définies comprennent la politique de confiance et la politique d'autorisation. De plus, cette politique d'autorisation ne peut pas être attachée à une autre entité IAM.

Vous pouvez supprimer un rôle lié à un service uniquement après la suppression préalable de ses ressources connexes. Cela protège vos AWS IoT TwinMaker ressources car vous ne pouvez pas supprimer par inadvertance l'autorisation d'accès aux ressources.

Pour plus d'informations sur les autres services qui prennent en charge les rôles liés à un service, consultez la section [AWS Services qui fonctionnent avec IAM](#) et recherchez les services dont la valeur est Oui dans la colonne Rôles liés à un service. Sélectionnez un Oui ayant un lien pour consulter la documentation du rôle lié à un service, pour ce service.

## Autorisations de rôle liées à un service pour AWS IoT TwinMaker

AWS IoT TwinMaker utilise le rôle lié au service nommé `AWSServiceRoleForIoT` : permet d' AWS IoT TwinMaker appeler d'autres AWS services et de synchroniser leurs ressources en votre nom.

Le rôle lié au service `AWSService RoleForlo TTwin Maker` fait confiance aux services suivants pour assumer le rôle :

- `iottwinmaker.amazonaws.com`

La politique d'autorisations de rôle nommée `AWSIoT Twin MakerServiceRolePolicy` AWS IoT TwinMaker permet d'effectuer les actions suivantes sur les ressources spécifiées :

- Action : `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` sur all your `iotsitewise` asset and `asset-model` resources

Vous devez configurer les autorisations de manière à permettre à vos utilisateurs, groupes ou rôles de créer, modifier ou supprimer un rôle lié à un service. Pour plus d'informations, consultez [Autorisations de rôles liés à un service](#) dans le Guide de l'utilisateur IAM.

## Création d'un rôle lié à un service pour AWS IoT TwinMaker

Vous n'avez pas besoin de créer manuellement un rôle lié à un service. Lorsque vous synchronisez vos AWS IoT SiteWise actifs et vos modèles d'actifs (synchronisation des actifs) dans l' AWS Management Console AWS API AWS CLI, vous AWS IoT TwinMaker créez le rôle lié au service pour vous.

Si vous supprimez ce rôle lié à un service et que vous avez ensuite besoin de le recréer, vous pouvez utiliser la même procédure pour recréer le rôle dans votre compte. Lorsque vous synchronisez vos AWS IoT SiteWise actifs et vos modèles d'actifs (synchronisation des actifs), AWS IoT TwinMaker crée à nouveau le rôle lié au service pour vous.

Vous pouvez également utiliser la console IAM pour créer un rôle lié à un service avec le cas d'utilisation « IoT TwinMaker - Managed Role ». Dans l'API AWS CLI ou dans l' AWS API, créez un rôle lié à un service avec le nom du `iottwinmaker.amazonaws.com` service. Pour plus d'informations, consultez [Création d'un rôle lié à un service](#) dans le Guide de l'utilisateur IAM. Si vous supprimez ce rôle lié à un service, vous pouvez utiliser ce même processus pour créer le rôle à nouveau.

## Modification d'un rôle lié à un service pour AWS IoT TwinMaker

AWS IoT TwinMaker ne vous permet pas de modifier le rôle lié au service `AWSServiceRoleForIoT Twin Maker`. Après avoir créé un rôle lié à un service, vous ne pouvez pas changer le nom du rôle, car plusieurs entités peuvent faire référence à ce rôle. Néanmoins, vous pouvez modifier la description du rôle à l'aide d'IAM. Pour plus d'informations, consultez [Modification d'un rôle lié à un service](#) dans le Guide de l'utilisateur IAM.

## Supprimer un rôle lié à un service pour AWS IoT TwinMaker

Si vous n'avez plus besoin d'utiliser une fonctionnalité ou un service qui nécessite un rôle lié à un service, nous vous recommandons de supprimer ce rôle. De cette façon, vous n'avez aucune entité inutilisée qui n'est pas surveillée ou gérée activement. Toutefois, vous devez nettoyer tous les espaces de travail ServiceLinked-Workspaces qui utilisent encore votre rôle lié à un service avant de pouvoir le supprimer manuellement.

### Note

Si le AWS IoT TwinMaker service utilise le rôle lorsque vous essayez de supprimer les ressources, la suppression risque d'échouer. Si cela se produit, patientez quelques minutes et réessayez.

Pour supprimer manuellement le rôle lié au service à l'aide d'IAM

Utilisez la console IAM, le AWS CLI, ou l' AWS API pour supprimer le rôle lié au service AWSService RoleForIoT Twin Maker. Pour plus d'informations, consultez la section [Suppression d'un rôle lié à un service](#) dans le Guide de l'utilisateur IAM.

## Régions prises en charge pour les rôles AWS IoT TwinMaker liés à un service

AWS IoT TwinMaker prend en charge l'utilisation de rôles liés au service dans toutes les régions où le service est disponible. Pour plus d'informations, consultez [AWS Régions et points de terminaison](#).

## AWS politiques gérées pour AWS IoT TwinMaker

Pour ajouter des autorisations aux utilisateurs, aux groupes et aux rôles, il est plus facile d'utiliser des politiques AWS gérées que de les rédiger vous-même. Il faut du temps et de l'expertise pour [créer des politiques gérées par le client IAM](#) qui ne fournissent à votre équipe que les autorisations dont elle a besoin. Pour démarrer rapidement, vous pouvez utiliser nos politiques AWS gérées. Ces politiques couvrent des cas d'utilisation courants et sont disponibles dans votre Compte AWS. Pour plus d'informations sur les politiques AWS gérées, voir les [politiques AWS gérées](#) dans le guide de l'utilisateur IAM.

AWS les services maintiennent et mettent à jour les politiques AWS gérées. Vous ne pouvez pas modifier les autorisations dans les politiques AWS gérées. Les services ajoutent occasionnellement

des autorisations à une politique gérée par AWS pour prendre en charge de nouvelles fonctionnalités. Ce type de mise à jour affecte toutes les identités (utilisateurs, groupes et rôles) auxquelles la politique est attachée. Les services sont très susceptibles de mettre à jour une politique gérée par AWS quand une nouvelle fonctionnalité est lancée ou quand de nouvelles opérations sont disponibles. Les services ne suppriment pas les autorisations d'une politique AWS gérée. Les mises à jour des politiques n'endommageront donc pas vos autorisations existantes.

En outre, AWS prend en charge les politiques gérées pour les fonctions professionnelles qui couvrent plusieurs services. Par exemple, la politique `ReadOnlyAccess` AWS gérée fournit un accès en lecture seule à tous les AWS services et ressources. Lorsqu'un service lance une nouvelle fonctionnalité, il AWS ajoute des autorisations en lecture seule pour les nouvelles opérations et ressources. Pour obtenir la liste des politiques de fonctions professionnelles et leurs descriptions, consultez la page [politiques gérées par AWS pour les fonctions de tâche](#) dans le Guide de l'utilisateur IAM.

## AWS politique gérée : `AWSIoT Twin MakerServiceRolePolicy`

Vous ne pouvez pas vous associer `AWSIoT Twin MakerServiceRolePolicy` à vos entités IAM. Cette politique est attachée à un rôle lié au service qui permet à d'effectuer des actions en votre nom. Pour de plus amples informations, veuillez consulter [Autorisations de rôle liées à un service pour AWS IoT TwinMaker](#).

La politique d'autorisations de rôle nommée `AWSIoT Twin MakerServiceRolePolicy` AWS IoT TwinMaker permet d'effectuer les actions suivantes sur les ressources spécifiées :

- Action : `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` sur all your `iotsitewise` asset and asset-model resources

## JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "SiteWiseAssetReadAccess",
      "Effect": "Allow",
      "Action": [
        "iotsitewise:DescribeAsset"
      ],
      "Resource": [
        "arn:aws:iotsitewise:*:*:asset/*"
      ]
    },
    {
      "Sid": "SiteWiseAssetModelReadAccess",
      "Effect": "Allow",
      "Action": [
        "iotsitewise:DescribeAssetModel"
      ],
      "Resource": [
        "arn:aws:iotsitewise:*:*:asset-model/*"
      ]
    },
    {
      "Sid": "SiteWiseAssetModelAndAssetListAccess",
      "Effect": "Allow",
      "Action": [
        "iotsitewise:ListAssets",
        "iotsitewise:ListAssetModels"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Sid": "TwinMakerAccess",
      "Effect": "Allow",
      "Action": [
        "iottwinmaker:GetEntity",
        "iottwinmaker:CreateEntity",

```



| Modifier | Description   | Date |
|----------|---|------|
|          | <p>suivantes sur les ressources spécifiées :</p> <ul style="list-style-type: none"> <li>Action : <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> sur all your iotsitewise asset and asset-model resources</li> </ul> <p>Pour de plus amples informations, veuillez consulter <a href="#">Autorisations de rôle liées</a></p> |      |

| Modifier                              | Description  | Date        |
|---------------------------------------|--|-------------|
|                                       | <a href="#">à un service pour AWS IoT TwinMaker.</a>                         |             |
| a commencé à suivre les modifications | a commencé à suivre les modifications apportées AWS à ses politiques gérées. | 11 mai 2022 |

## AWS IoT TwinMaker et points de terminaison VPC d'interface ()AWS PrivateLink

Vous pouvez établir une connexion privée entre votre cloud privé virtuel (VPC) et en AWS IoT TwinMaker créant un point de terminaison VPC d'interface. Les points de terminaison de l'interface sont alimentés par [AWS PrivateLink](#) lesquels vous pouvez accéder en privé AWS IoT TwinMaker APIs sans passerelle Internet, périphérique de traduction d'adresses réseau (NAT), connexion VPN ou connexion AWS Direct Connect. AWS IoT TwinMaker prend en charge les deux IPv4 et IPv6 (double pile) via ses points de terminaison d'interface. Les instances de votre VPC n'ont pas besoin d'adresses IP publiques pour communiquer avec elles. AWS IoT TwinMaker APIs Le trafic entre votre VPC et celui qui AWS IoT TwinMaker ne quitte pas le réseau Amazon.

Chaque point de terminaison d'interface est représenté par une ou plusieurs [interfaces réseau Elastic](#) dans vos sous-réseaux.

Pour de plus amples informations, consultez [Points de terminaison VPC \(AWS PrivateLink\)](#) dans le Guide de l'utilisateur Amazon VPC.

### Considérations relatives aux points de AWS IoT TwinMaker terminaison VPC

Avant de configurer un point de terminaison VPC d'interface pour AWS IoT TwinMaker, consultez les [propriétés et les limites du point de terminaison d'interface](#) dans le guide de l'utilisateur Amazon VPC.

AWS IoT TwinMaker permet d'appeler toutes ses actions d'API depuis votre VPC.

- Pour les opérations d'API du plan de données, utilisez le point de terminaison suivant :

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

Les opérations de l'API du plan de données sont les suivantes :

- [GetPropertyValue](#)
- [GetPropertyValueHistory](#)
- [BatchPutPropertyValues](#)
- Pour les opérations de l'API du plan de contrôle, utilisez le point de terminaison suivant :

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

Les opérations d'API du plan de contrôle prises en charge sont les suivantes :

- [CreateComponentType](#)
- [CreateEntity](#)
- [CreateScene](#)
- [CreateWorkspace](#)
- [DeleteComponentType](#)
- [DeleteEntity](#)
- [DeleteScene](#)
- [DeleteWorkspace](#)
- [GetComponentType](#)
- [GetEntity](#)
- [GetScene](#)
- [GetWorkspace](#)
- [ListComponentTypes](#)
- [ListComponentTypes](#)
- [ListEntities](#)
- [ListScenes](#)
- [ListTagsForResource](#)
- [ListWorkspaces](#)
- [TagResource](#)
- [UntagResource](#)
- [UpdateComponentType](#)
- [UpdateEntity](#)

- [UpdateScene](#)
- [UpdateWorkspace](#)

## Création d'un point de terminaison VPC d'interface pour AWS IoT TwinMaker

Vous pouvez créer un point de terminaison VPC pour le AWS IoT TwinMaker service à l'aide de la console Amazon VPC ou du (). AWS Command Line Interface AWS CLI Pour plus d'informations, consultez [Création d'un point de terminaison d'interface](#) dans le Guide de l'utilisateur Amazon VPC.

Créez un point de terminaison VPC AWS IoT TwinMaker qui utilise le nom de service suivant.

- Pour les opérations d'API du plan de données, utilisez le nom de service suivant :

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

- Pour les opérations d'API du plan de contrôle, utilisez le nom de service suivant :

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

Si vous activez le DNS privé pour le point de terminaison, vous pouvez envoyer des demandes d' AWS IoT TwinMaker API en utilisant son nom DNS par défaut pour la région, par exemple, `iottwinmaker.us-east-1.amazonaws.com`.

Pour plus d'informations, consultez [Accès à un service via un point de terminaison d'interface](#) dans le Guide de l'utilisateur Amazon VPC.

AWS IoT TwinMaker PrivateLink est pris en charge dans les régions suivantes :

- us-east-1

Le ControlPlane service est pris en charge dans les zones de disponibilité suivantes : use1-az1, use1-az2, et use1-az6.

Le DataPlane service est pris en charge dans les zones de disponibilité suivantes : use1-az1, use1-az2, et use1-az4.

- us-west-2

Les DataPlane services ControlPlane et sont pris en charge dans les zones de disponibilité suivantes : usw2-az1 usw2-az2, et usw2-az3.

- eu-west-1
- eu-central-1
- ap-southeast-1
- ap-southeast-2

Pour plus d'informations sur les zones de disponibilité, consultez [la section Zone de disponibilité IDs pour vos AWS ressources - AWS Resource Access Manager](#).

## Accès AWS IoT TwinMaker via un point de terminaison VPC d'interface

Lorsque vous créez un point de terminaison d'interface, il AWS IoT TwinMaker génère des noms d'hôte DNS spécifiques au point de terminaison avec lesquels vous pouvez communiquer. AWS IoT TwinMaker L'option DNS privé est activée par défaut. Pour plus d'informations, consultez la section [Utilisation de zones hébergées privées](#) dans le guide de l'utilisateur Amazon VPC.

Si vous activez le DNS privé pour le point de terminaison, vous pouvez envoyer des demandes d'API AWS IoT TwinMaker via l'un des points de terminaison VPC suivants.

- Pour les opérations de l'API du plan de données, utilisez le point de terminaison suivant. Remplacez *region* par votre AWS région.

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

- Pour les opérations de l'API du plan de contrôle, utilisez le point de terminaison suivant. Remplacez *region* par votre AWS région.

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

Si vous désactivez le DNS privé pour le point de terminaison, vous devez effectuer les opérations suivantes pour y accéder AWS IoT TwinMaker via le point de terminaison :

- Spécifiez l'URL du point de terminaison VPC dans les demandes d'API.

- Pour les opérations de l'API du plan de données, utilisez l'URL du point de terminaison suivant. Remplacez *vpc-endpoint-id* et *region* par l'ID et la région de votre point de terminaison VPC.

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

- Pour les opérations de l'API du plan de contrôle, utilisez l'URL du point de terminaison suivant. Remplacez *vpc-endpoint-id* et *region* par l'ID et la région de votre point de terminaison VPC.

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

- Désactivez l'injection de préfixe d'hôte. Le AWS CLI et AWS SDKs ajoutent différents préfixes d'hôte au point de terminaison du service lorsque vous appelez chaque opération d'API. Cela entraîne la production AWS CLI AWS SDKs d'une valeur non valide URLs AWS IoT TwinMaker lorsque vous spécifiez un point de terminaison VPC.

#### Important

Vous ne pouvez pas désactiver l'injection de préfixe d'hôte dans AWS CLI ou Outils AWS pour PowerShell. Cela signifie que si vous avez désactivé le DNS privé, vous ne pourrez pas utiliser le point de AWS CLI terminaison VPC ni Outils AWS pour PowerShell y accéder AWS IoT TwinMaker via celui-ci. Si vous souhaitez utiliser ces outils pour accéder AWS IoT TwinMaker via le point de terminaison, activez le DNS privé.

Pour plus d'informations sur la façon de désactiver l'injection de préfixe d'hôte dans le AWS SDKs, consultez les sections de documentation suivantes pour chaque SDK :

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

- [AWS SDK pour Python \(Boto3\)](#)
- [AWS SDK pour Ruby](#)

Pour plus d'informations, consultez [Accès à un service via un point de terminaison d'interface](#) dans le Guide de l'utilisateur Amazon VPC.

## Création d'une politique de point de terminaison VPC pour AWS IoT TwinMaker

Vous pouvez attacher une stratégie de point de terminaison à votre point de terminaison d'un VPC qui contrôle l'accès à AWS IoT TwinMaker. La politique spécifie les informations suivantes :

- Le principal qui peut exécuter des actions.
- Les actions qui peuvent être effectuées.
- Les ressources sur lesquelles les actions peuvent être exécutées.

Pour plus d'informations, consultez [Contrôle de l'accès aux services avec points de terminaison d'un VPC](#) dans le Guide de l'utilisateur Amazon VPC.

Exemple : politique de point de terminaison VPC pour les actions AWS IoT TwinMaker

Voici un exemple de politique de point de terminaison pour AWS IoT TwinMaker. Lorsqu'elle est attachée à un point de terminaison, cette politique accorde l'accès aux AWS IoT TwinMaker actions répertoriées pour l'utilisateur IAM `iottwinmakeradmin` dans le AWS compte `123456789012` sur toutes les ressources.

```
{
  "Statement": [
    {
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:user/role"
      },
      "Resource": "*",
      "Effect": "Allow",
      "Action": [
        "iottwinmaker:CreateEntity",
        "iottwinmaker:GetScene",
        "iottwinmaker:ListEntities"
      ]
    }
  ]
}
```

```
    ]
  }
]
}
```

## Validation de conformité pour AWS IoT TwinMaker

Pour savoir si un [programme Services AWS de conformité Service AWS s'inscrit dans le champ d'application de programmes de conformité](#) spécifiques, consultez Services AWS la section de conformité et sélectionnez le programme de conformité qui vous intéresse. Pour des informations générales, voir Programmes de [AWS conformité Programmes AWS](#) de .

Vous pouvez télécharger des rapports d'audit tiers à l'aide de AWS Artifact. Pour plus d'informations, voir [Téléchargement de rapports dans AWS Artifact](#) .

Votre responsabilité en matière de conformité lors de l'utilisation Services AWS est déterminée par la sensibilité de vos données, les objectifs de conformité de votre entreprise et les lois et réglementations applicables. Pour plus d'informations sur votre responsabilité en matière de conformité lors de l'utilisation Services AWS, consultez [AWS la documentation de sécurité](#).

## Résilience dans AWS IoT TwinMaker

L'infrastructure AWS mondiale est construite autour Régions AWS de zones de disponibilité. Régions AWS fournissent plusieurs zones de disponibilité physiquement séparées et isolées, connectées par un réseau à faible latence, à haut débit et hautement redondant. Avec les zones de disponibilité, vous pouvez concevoir et exploiter des applications et des bases de données qui basculent automatiquement d'une zone à l'autre sans interruption. Les zones de disponibilité sont davantage disponibles, tolérantes aux pannes et ont une plus grande capacité de mise à l'échelle que les infrastructures traditionnelles à un ou plusieurs centres de données.

Pour plus d'informations sur les zones de disponibilité Régions AWS et les zones de disponibilité, consultez la section [Infrastructure AWS globale](#).

Outre l'infrastructure AWS mondiale, AWS IoT TwinMaker propose plusieurs fonctionnalités pour répondre à vos besoins en matière de résilience et de sauvegarde des données.

## Sécurité de l'infrastructure dans AWS IoT TwinMaker

En tant que service géré, AWS IoT TwinMaker il est protégé par les procédures de sécurité du réseau AWS mondial décrites dans le livre blanc [Amazon Web Services : présentation des processus de sécurité](#).

Vous utilisez des appels d'API AWS publiés pour accéder AWS IoT TwinMaker via le réseau. Les clients doivent prendre en charge le protocole TLS (Transport Layer Security) 1.2 ou version ultérieure. Nous recommandons TLS 1.3 ou version ultérieure. Les clients doivent aussi prendre en charge les suites de chiffrement PFS (Perfect Forward Secrecy) comme Ephemeral Diffie-Hellman (DHE) ou Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). La plupart des systèmes modernes tels que Java 7 et les versions ultérieures prennent en charge ces modes.

En outre, les demandes doivent être signées à l'aide d'un ID de clé d'accès et d'une clé d'accès secrète associée à un principal IAM. Vous pouvez également utiliser [AWS Security Token Service](#) (AWS STS) pour générer des informations d'identification de sécurité temporaires et signer les demandes.

# Points de terminaison et quotas

## AWS IoT TwinMaker points de terminaison et quotas

Vous trouverez des informations sur les AWS IoT TwinMaker points de terminaison et les quotas dans la [référence AWS générale](#).

- Pour plus d'informations sur les points de terminaison de service, consultez la section [Points de terminaison AWS IoT TwinMaker de service](#).
- Pour plus d'informations sur les quotas, consultez la section [Quotas AWS IoT TwinMaker de service](#).
- Pour plus d'informations sur les limites de limitation des API, consultez la section [Limites de limitation des AWS IoT TwinMaker API](#).

## Informations supplémentaires sur les points de terminaison AWS IoT TwinMaker

Pour vous connecter par programmation à AWS IoT TwinMaker, utilisez un point de terminaison. Si vous utilisez un client HTTP, vous devez préfixer le plan de contrôle et le plan de données APIs comme suit. Cependant, il n'est pas nécessaire d'ajouter un préfixe au AWS SDK et aux AWS Command Line Interface commandes, car ils ajoutent automatiquement le préfixe nécessaire.

- Utilisez le api préfixe du plan APIs de contrôle. Par exemple, `api.iottwinmaker.us-west-1.amazonaws.com`.
- Utilisez le data préfixe du plan APIs de données. Par exemple, `data.iottwinmaker.us-west-1.amazonaws.com`.

# Historique du Guide de l'utilisateur AWS IoT TwinMaker

Le tableau suivant décrit les versions de documentation pour AWS IoT TwinMaker.

| Modification  | Description   | Date             |
|---|---|------------------|
| <a href="#">Nouveau rôle lié à un service et nouvelle politique IAM</a> | AWS IoT TwinMaker a ajouté un nouveau rôle lié à un service, appelé. <a href="#">AWSServiceRoleForIoT TwinMaker</a> AWS IoT TwinMaker a ajouté ce nouveau rôle lié au service pour permettre d'AWS IoT TwinMaker appeler d'autres AWS services et de synchroniser leurs ressources en votre nom. La nouvelle politique <code>AWSIoT TwinMakerServiceRolePolicy</code> IAM est attachée à ce rôle, et elle autorise d'autres services AWS IoT TwinMaker à appeler d'autres AWS services et à synchroniser leurs ressources en votre nom. | 17 novembre 2023 |
| <a href="#">Première version</a>  | Publication initiale du guide de AWS IoT TwinMaker l'utilisateur  | 30 novembre 2021 |

Les traductions sont fournies par des outils de traduction automatique. En cas de conflit entre le contenu d'une traduction et celui de la version originale en anglais, la version anglaise prévaudra.